

Foreman - Bug #13801

foreman passenger memory leak

02/18/2016 06:08 PM - Michael Eklund

Status:	New	
Priority:	Normal	
Assignee:		
Category:	Web Interface	
Target version:		
Difficulty:		Fixed in Releases:
Triaged:		Found in Releases: 1.12.4
Bugzilla link:		Red Hat JIRA:
Pull request:		
Description		
I don't have any real information other than I was seeing OOMs and checked passenger:		
<pre>----- Passenger processes ----- PID VMSize Private Name ----- 2105 218.0 MB 0.1 MB PassengerWatchdog 2108 1464.9 MB 1.3 MB PassengerHelperAgent 2115 234.2 MB 0.1 MB PassengerLoggingAgent 2618 18017.7 MB 17460.1 MB Passenger RackApp: /usr/share/foreman 2628 732.8 MB 8.6 MB Passenger RackApp: /usr/share/foreman 2663 861.9 MB 214.9 MB Passenger RackApp: /usr/share/foreman 2673 732.8 MB 207.4 MB Passenger RackApp: /usr/share/foreman 2716 733.1 MB 11.5 MB Passenger RackApp: /usr/share/foreman 2725 733.2 MB 222.7 MB Passenger RackApp: /usr/share/foreman 2738 733.3 MB 8.4 MB Passenger RackApp: /usr/share/foreman 2755 733.4 MB 4.3 MB Passenger RackApp: /usr/share/foreman 2762 733.5 MB 220.6 MB Passenger RackApp: /usr/share/foreman 2769 733.6 MB 249.5 MB Passenger RackApp: /usr/share/foreman 2780 733.7 MB 4.1 MB Passenger RackApp: /usr/share/foreman 12655 733.4 MB 316.8 MB Passenger RackApp: /usr/share/foreman ### Processes: 15 ### Total private dirty RSS: 18930.45 MB</pre>		

History

#1 - 02/19/2016 03:39 AM - Dominic Cleal

- Category set to Web Interface

Do you have any plugins installed (Administer > About, or foreman-rake plugin:list)? Try to reproduce it on a regular Foreman installation if you do.

I can't really suggest anything else without some information about how to reproduce it.

It's possible that sending a SIGABRT to the large process will give you some backtrace information about what it's doing, perhaps it's stuck processing a request. This ought to be logged to Apache's error log.

You can also try tuning the problem away by limiting the lifetime of Passenger workers, e.g. <https://www.phusionpassenger.com/library/config/apache/reference/#passengermaxrequests>

#2 - 02/19/2016 09:58 AM - Michael Eklund

```
root@cfg01.atl:/usr/lib/collectd# foreman-rake plugin:list  
[deprecated] I18n.enforce_available_locales will default to true in the future. If you really want to skip validation of your locale you can set I18n.enforce_available_locales = false to avoid this message.  
Collecting plugin information  
Foreman plugin: foreman_digitalocean, 0.2.1, Tommy McNeely, Daniel Lobato, Provision and manage DigitalOcean droplets from Foreman.  
Foreman plugin: foreman_discovery, 4.1.2, Amos Benari, Bryan Kearney, ChairmanTubeAmp, Daniel Lobato Garcia, Domin
```

ic Cleal, Eric D. Helms, Frank Wall, Greg Sutcliffe, Imri Zvik, Joseph Mitchell Magen, Lukas Zapletal, Lukáš Zapletal, Marek Hulan, Martin Bačovský, Matt Jarvis, Michael Moll, Nick, Ohad Levy, Ori Rabin, Petr Chalupa, Phirince Philip, Sc ubafloyd, Shlomi Zadok, Stephen Benjamin, Yann Cézard, MaaS Discovery Plugin engine for Foreman
Foreman plugin: foreman_graphite, 0.0.3, Ohad Levy, adds graphite support to foreman
Foreman plugin: foreman_hooks, 0.3.9, Dominic Cleal, Plugin engine for Foreman that enables running custom hook scripts on Foreman events
Foreman plugin: foreman_setup, 3.0.2, Dominic Cleal, Plugin for Foreman that helps set up provisioning.
Foreman plugin: puppetdb_foreman, 0.2.0, Daniel Lobato Garcia, Disable hosts on PuppetDB after they are deleted or built in Foreman, and proxy the PuppetDB dashboard to Foreman. Follow https://github.com/theforeman/puppetdb_foreman and raise an issue/submit a pull request if you need extra functionality. You can also find some help in #theforeman IRC channel on Freenode.

They are normal plugins, I think.

I have set passenger_max_requests to 50. It looks like I can tune that number up some, though.

#3 - 02/19/2016 09:59 AM - Michael Eklund

Though foreman_graphite is pretty recent. I may disable, as the info is not that useful.

#4 - 02/19/2016 06:52 PM - Michael Eklund

I let it run without recycling threads today and was able to get a backtrace with the SIGABRT, though it does not look that useful. I did disable foreman_graphite, so it is not that one.

```
App 20601 stderr: [ 2016-02-19 18:40:41.1492 20858/0x0000000099c0f0 (Main thread) request_handler.rb:394 ]: ===== Process 20858: backtrace dump =====
App 20601 stderr: -----
App 20601 stderr: # Thread: #<Thread:0x0000000099c0f0 run>(Main thread), [main thread], [current thread], alive = true
App 20601 stderr: -----
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/utils.rb:146:in `block in global_backtrace_report'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/utils.rb:145:in `each'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/utils.rb:145:in `global_backtrace_report'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:394:in `print_status_report'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:380:in `block in install_useful_signal_handlers'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:517:in `call'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:517:in `select'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:517:in `wait_until_termination_requested'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:206:in `main_loop'
App 20601 stderr: /usr/share/passenger/helper-scripts/rack-preloader.rb:161:in `'
App 20601 stderr: /usr/share/passenger/helper-scripts/rack-preloader.rb:29:in `'
App 20601 stderr: /usr/share/passenger/helper-scripts/rack-preloader.rb:28:in `'
App 20601 stderr: -----
App 20601 stderr: # Thread: #<Thread:0x00000000981d40 sleep>(Worker 1), alive = true
App 20601 stderr: -----
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler/thread_handler.rb:127:in `accept_and_process_next_request'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler/thread_handler.rb:110:in `main_loop'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:448:in `block (3 levels) in start_threads'
App 20601 stderr: /usr/share/foreman/vendor/ruby/1.9.1/gems/logging-2.0.0/lib/logging/diagnostic_context.rb:448:in `call'
App 20601 stderr: /usr/share/foreman/vendor/ruby/1.9.1/gems/logging-2.0.0/lib/logging/diagnostic_context.rb:448:in `block in create_with_logging_context'
App 20601 stderr: -----
App 20601 stderr: # Thread: #<Thread:0x00000000981980 sleep>(HTTP helper worker), alive = true
App 20601 stderr: -----
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler/thread_handler.rb:127:in `accept_and_process_next_request'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler/thread_handler.rb:110:in `main_loop'
App 20601 stderr: /usr/lib/ruby/vendor_ruby/phusion_passenger/request_handler.rb:464:in `block (2 levels) in start_threads'
App 20601 stderr: /usr/share/foreman/vendor/ruby/1.9.1/gems/logging-2.0.0/lib/logging/diagnostic_context.rb:448:in `call'
App 20601 stderr: /usr/share/foreman/vendor/ruby/1.9.1/gems/logging-2.0.0/lib/logging/diagnostic_context.rb:448:in `block in create_with_logging_context'
```

```
App 20601 stderr:
App 20601 stderr:
App 20601 stderr: [ 2016-02-19 18:40:41.1493 20858/0x0000000099c0f0(Main thread) request_handler.rb:395 ]: Thr
eads: [#<Thread:0x00000000981d40 sleep>, #<Thread:0x00000000981980 sleep>]
```

#5 - 02/19/2016 06:57 PM - Michael Eklund

Passenger status:

```
root@cfg01.atl:/usr/lib/collectd# passenger-status
Version : 4.0.37
Date    : 2016-02-19 18:33:25 -0500
Instance: 20517
----- General information -----
Max pool size : 12
Processes      : 12
Requests in top-level queue : 0
```

```
----- Application groups -----
/usr/share/foreman#default:
App root: /usr/share/foreman
Requests in queue: 0
* PID: 20840  Sessions: 0    Processed: 1545  Uptime: 6h 59m 58s
  CPU: 1%    Memory : 235M    Last used: 2h 36m 4
* PID: 20849  Sessions: 0    Processed: 2     Uptime: 6h 59m 58s
  CPU: 0%    Memory : 211M    Last used: 6h 59m 5
* PID: 20858  Sessions: 0    Processed: 644  Uptime: 6h 59m 58s
  CPU: 12%   Memory : 22419M  Last used: 28s ago
* PID: 20867  Sessions: 0    Processed: 834  Uptime: 6h 59m 57s
  CPU: 0%    Memory : 246M    Last used: 57m 44s
* PID: 20876  Sessions: 0    Processed: 2     Uptime: 6h 59m 57s
  CPU: 0%    Memory : 82M    Last used: 3h 54m 3
* PID: 20894  Sessions: 0    Processed: 1     Uptime: 6h 59m 57s
  CPU: 0%    Memory : 214M    Last used: 6h 59m 5
* PID: 20903  Sessions: 0    Processed: 1     Uptime: 6h 59m 57s
  CPU: 0%    Memory : 214M    Last used: 6h 59m 5
* PID: 20912  Sessions: 0    Processed: 61    Uptime: 6h 59m 57s
  CPU: 0%    Memory : 226M    Last used: 3h 54m 3
* PID: 20922  Sessions: 0    Processed: 1424  Uptime: 6h 59m 56s
  CPU: 1%    Memory : 231M    Last used: 2h 36m 4
* PID: 20929  Sessions: 0    Processed: 1683  Uptime: 6h 59m 56s
  CPU: 1%    Memory : 232M    Last used: 3h 54m 3
* PID: 327    Sessions: 0    Processed: 0     Uptime: 1h 47m 13s
  CPU: 0%    Memory : 52M    Last used: 1h 47m 1
* PID: 3101  Sessions: 0    Processed: 0     Uptime: 1h 26m 57s
  CPU: 0%    Memory : 50M    Last used: 1h 26m 5
```

```
root@cfg01.atl:/usr/lib/collectd# passenger-memory-stats
Version: 4.0.37
Date    : 2016-02-19 18:34:02 -0500
```

```
----- Apache processes -----
PID    PPID    VMSize    Private  Name
-----
20517  1       106.8 MB  0.2 MB   /usr/sbin/apache2 -k start
20520  20517  105.2 MB  0.4 MB   /usr/sbin/apache2 -k start
20539  20517  1988.0 MB 5.7 MB   /usr/sbin/apache2 -k start
20540  20517  1988.2 MB 7.6 MB   /usr/sbin/apache2 -k start
### Processes: 4
### Total private dirty RSS: 13.91 MB
```

```
----- Nginx processes -----
### Processes: 0
### Total private dirty RSS: 0.00 MB
```

```
----- Passenger processes -----
PID    VMSize    Private    Name
-----
327    465.2 MB  52.6 MB    Passenger RackApp: /usr/share/foreman
3101   464.1 MB  50.4 MB    Passenger RackApp: /usr/share/foreman
20521  218.0 MB  0.3 MB     PassengerWatchdog
20524  1464.9 MB 1.6 MB     PassengerHelperAgent
20531  298.3 MB  1.1 MB     PassengerLoggingAgent
20840  662.0 MB  218.3 MB   Passenger RackApp: /usr/share/foreman
```

```
20849 660.8 MB 54.9 MB Passenger RackApp: /usr/share/foreman
20858 22815.5 MB 22413.0 MB Passenger RackApp: /usr/share/foreman
20867 660.9 MB 238.5 MB Passenger RackApp: /usr/share/foreman
20876 531.0 MB 5.1 MB Passenger RackApp: /usr/share/foreman
20894 667.3 MB 127.9 MB Passenger RackApp: /usr/share/foreman
20903 667.4 MB 126.3 MB Passenger RackApp: /usr/share/foreman
20912 661.4 MB 152.2 MB Passenger RackApp: /usr/share/foreman
20922 661.5 MB 212.9 MB Passenger RackApp: /usr/share/foreman
20929 661.8 MB 160.2 MB Passenger RackApp: /usr/share/foreman
### Processes: 15
### Total private dirty RSS: 23815.28 MB
```

#6 - 02/23/2016 07:39 AM - Lukas Zapletal

What platform are you on? Can you do foreman-debug -u ?

#7 - 03/16/2016 03:08 PM - Michael Eklund

Ubuntu 14.04. I have hidden the problem with PassengerMaxRequests 100

I could schedule a time to make it happen and run debug when it grows if it will record useful info for you.

#8 - 05/13/2016 10:18 AM - Mike Fröhner

Hello,

I am having the same problem. After clicking the 'classes' button at <https://foreman/environments> one single ruby process consumes more than 50% of my 16GB memory. After restarting apache2 and closing the browsers tab the process was gone after some seconds.

I executed foreman-debug -u (but not while problem was happening):

```
HOSTNAME: foreman.
OS: debian
RELEASE: 8.4
FOREMAN: 1.11.1
RUBY: ruby 2.1.5p273 (2014-11-13) [x86_64-linux-gnu]
PUPPET: 3.8.6
```

A debug file has been created: /tmp/foreman-debug-mAEhX.tar.xz (221864 bytes)

Uploading...

The tarball has been uploaded, please contact us on our mailing list or IRC referencing the following URL:

<http://debugs.theforeman.org/foreman-debug-mAEhX.tar.xz>

#9 - 05/13/2016 10:22 AM - Mike Fröhner

I executed passenger-status while the problem was ongoing:

```
root@foreman:~ # passenger-status
Version : 4.0.53
Date : 2016-05-13 15:30:38 +0200
Instance: 9012
----- General information -----
Max pool size : 6
Processes : 6
Requests in top-level queue : 0

----- Application groups -----
/usr/share/foreman#default:
App root: /usr/share/foreman
Requests in queue: 0 * PID: 9165 Sessions: 0 Processed: 19 Uptime: 4m 6s
CPU: 26% Memory : 929M Last used: 1s ago * PID: 9345 Sessions: 1 Processed: 8 Uptime: 2m 20s
CPU: 64% Memory : 8682M Last used: 1m 49s ago * PID: 9455 Sessions: 0 Processed: 13 Uptime: 1m 4s
CPU: 4% Memory : 321M Last used: 38s ago

/etc/puppet/rack#default:
App root: /etc/puppet/rack
Requests in queue: 0 * PID: 9283 Sessions: 0 Processed: 199 Uptime: 3m 8s
CPU: 5% Memory : 109M Last used: 1s ago * PID: 9486 Sessions: 0 Processed: 79 Uptime: 57s
CPU: 15% Memory : 109M Last used: 26s ago * PID: 9510 Sessions: 0 Processed: 185 Uptime: 57s
CPU: 10% Memory : 45M Last used: 27s ago
```

#10 - 03/27/2017 12:09 PM - Will Foster

We're seeing the same issue on our Foreman (128G memory, 10 physical cores, Dell R630). We do frequent hammer params searches as part of our tooling that utilizes Foreman on the backend. Bouncing httpd seems to restore the memory and stop the machine from swapping but it slowly creeps up again.

The offender is Passenger RackApp: /usr/share/foreman

foreman-debug -u output

```
HOSTNAME: foreman.rdu.openstack.engineering.example.com
OS: redhat
RELEASE: Red Hat Enterprise Linux Server release 7.3 (Maipo)
FOREMAN: 1.12.4
RUBY: ruby 2.0.0p648 (2015-12-16) [x86_64-linux]
PUPPET: 3.8.7
DENIALS: 94
```

<http://debugs.theforeman.org/foreman-debug-XnyVX.tar.xz>

#11 - 03/27/2017 04:50 PM - Kambiz Aghaiepour

Here is what we see when we run `passenger-memory-stats` :

PID	VMSize	Private	Name
51624	212.2 MB	0.3 MB	PassengerWatchdog
51627	2233.4 MB	4.0 MB	PassengerHelperAgent
51634	215.0 MB	0.9 MB	PassengerLoggingAgent
51681	1551.1 MB	88.3 MB	Passenger AppPreloader: /usr/share/foreman
51688	151.9 MB	25.3 MB	Passenger AppPreloader: /etc/puppet/rack
51742	287.0 MB	32.3 MB	Passenger RackApp: /etc/puppet/rack
51749	287.1 MB	36.4 MB	Passenger RackApp: /etc/puppet/rack
51756	283.8 MB	33.2 MB	Passenger RackApp: /etc/puppet/rack
51785	8951.8 MB	8191.5 MB	Passenger RackApp: /usr/share/foreman
51793	8504.8 MB	7601.2 MB	Passenger RackApp: /usr/share/foreman
51803	8057.5 MB	7112.0 MB	Passenger RackApp: /usr/share/foreman
51811	9723.0 MB	8685.9 MB	Passenger RackApp: /usr/share/foreman
51819	9211.7 MB	8131.4 MB	Passenger RackApp: /usr/share/foreman
51826	9085.5 MB	7888.6 MB	Passenger RackApp: /usr/share/foreman
52545	8063.2 MB	6760.5 MB	Passenger RackApp: /usr/share/foreman
52555	8192.5 MB	6828.9 MB	Passenger RackApp: /usr/share/foreman
52614	7936.9 MB	6500.3 MB	Passenger RackApp: /usr/share/foreman
52664	5248.3 MB	3766.2 MB	Passenger RackApp: /usr/share/foreman
53284	5121.0 MB	3651.0 MB	Passenger RackApp: /usr/share/foreman
53325	4609.1 MB	3118.9 MB	Passenger RackApp: /usr/share/foreman
53359	5890.3 MB	4296.9 MB	Passenger RackApp: /usr/share/foreman
53682	287.4 MB	36.8 MB	Passenger RackApp: /etc/puppet/rack
53732	3459.0 MB	1870.4 MB	Passenger RackApp: /usr/share/foreman
53755	3075.7 MB	1458.1 MB	Passenger RackApp: /usr/share/foreman
54026	287.4 MB	36.3 MB	Passenger RackApp: /etc/puppet/rack
54199	284.0 MB	33.5 MB	Passenger RackApp: /etc/puppet/rack

1. Processes: 26
2. Total private dirty RSS: 86189.14 MB

and subsequent runs show the memory utilization under VMSize and Private continue to go up until memory on the system is exhausted. Currently we are working around the issue by restarting apache but this is not a viable workaround since it means the application is at times unavailable. What appears to cause memory utilization to continue to go up is a number of API calls (via other scripts run out of cron) that query foreman using e.g.:

```
hammer host list --search params.&lt;some host parameter&gt;=&lt;some value&gt;;
```

Kambiz

#12 - 03/27/2017 04:53 PM - Kambiz Aghaiepour

```
hammer host list --search params.<some host parameter>=<some value>
```

#13 - 03/28/2017 06:12 PM - Will Foster

Kambiz Aghaiepour wrote:

[...]

Adding our oom killer stack trace here

```

Mar 26 03:28:20 foreman kernel: kthreadd invoked oom-killer: gfp_mask=0x3000d0, order=2, oom_score_adj=0
Mar 26 03:28:21 foreman kernel: kthreadd cpuset=/ mems_allowed=0-1
Mar 26 03:28:21 foreman kernel: CPU: 9 PID: 2 Comm: kthreadd Not tainted 3.10.0-327.36.1.el7.x86_64 #1
Mar 26 03:28:21 foreman kernel: Hardware name: Dell Inc. PowerEdge R630/0CNCJW, BIOS 1.5.4 10/002/2015
Mar 26 03:28:21 foreman kernel: ffff8810285a8b80 000000003fae2746 ffff88102860baa8 ffffffff81636301
Mar 26 03:28:21 foreman kernel: ffff88102860bb38 ffffffff8163129c ffff8810284df210 ffff8810284df228
Mar 26 03:28:21 foreman kernel: 31a414ff00000206 fbfeffff00000000 0000000000000001 ffffffff81128c03
Mar 26 03:28:21 foreman kernel: Call Trace:
Mar 26 03:28:21 foreman kernel: [<ffffffff81636301>] dump_stack+0x19/0x1b
Mar 26 03:28:21 foreman kernel: [<ffffffff8163129c>] dump_header+0x8e/0x214
Mar 26 03:28:21 foreman kernel: [<ffffffff81128c03>] ? proc_do_uts_string+0xf3/0x130
Mar 26 03:28:21 foreman kernel: [<ffffffff8116d21e>] oom_kill_process+0x24e/0x3b0
Mar 26 03:28:21 foreman kernel: [<ffffffff81088e4e>] ? has_capability_noaudit+0x1e/0x30
Mar 26 03:28:21 foreman kernel: [<ffffffff8116da46>] out_of_memory+0x4b6/0x4f0
Mar 26 03:28:21 foreman kernel: [<ffffffff81173c36>] __alloc_pages_nodemask+0xaa6/0xba0
Mar 26 03:28:21 foreman kernel: [<ffffffff81078dd3>] copy_process.part.25+0x163/0x1610
Mar 26 03:28:21 foreman kernel: [<ffffffff810c22de>] ? dequeue_task_fair+0x42e/0x640
Mar 26 03:28:22 foreman kernel: [<ffffffff810a5ac0>] ? kthread_create_on_node+0x140/0x140
Mar 26 03:28:22 foreman kernel: [<ffffffff8107a461>] do_fork+0xe1/0x320
Mar 26 03:28:22 foreman kernel: [<ffffffff8107a6c6>] kernel_thread+0x26/0x30
Mar 26 03:28:22 foreman kernel: [<ffffffff810a6692>] kthreadd+0x2b2/0x2f0
Mar 26 03:28:22 foreman kernel: [<ffffffff810a63e0>] ? kthread_create_on_cpu+0x60/0x60
Mar 26 03:28:22 foreman kernel: [<ffffffff81646958>] ret_from_fork+0x58/0x90
Mar 26 03:28:22 foreman kernel: [<ffffffff810a63e0>] ? kthread_create_on_cpu+0x60/0x60

Mar 26 04:39:10 foreman kernel: diagnostic_con* invoked oom-killer: gfp_mask=0x280da, order=0, oom_score_adj=0
Mar 26 04:39:11 foreman kernel: diagnostic_con* cpuset=/ mems_allowed=0-1
Mar 26 04:39:11 foreman kernel: CPU: 4 PID: 61308 Comm: diagnostic_con* Not tainted 3.10.0-327.36.1.el7.x86_64
#1
Mar 26 04:39:11 foreman kernel: Hardware name: Dell Inc. PowerEdge R630/0CNCJW, BIOS 1.5.4 10/002/2015
Mar 26 04:39:11 foreman kernel: ffff880eb00f5c00 00000000a26edb70 ffff88101496faf8 ffffffff81636301
Mar 26 04:39:11 foreman kernel: ffff88101496fb88 ffffffff8163129c ffff88102114f440 ffff88102114f458
Mar 26 04:39:11 foreman kernel: 05a404e300000206 fbfeffff00000000 0000000000000001 ffffffff81128c03
Mar 26 04:39:11 foreman kernel: Call Trace:
Mar 26 04:39:11 foreman kernel: [<ffffffff81636301>] dump_stack+0x19/0x1b
Mar 26 04:39:11 foreman kernel: [<ffffffff8163129c>] dump_header+0x8e/0x214
Mar 26 04:39:11 foreman kernel: [<ffffffff81128c03>] ? proc_do_uts_string+0xf3/0x130
Mar 26 04:39:11 foreman kernel: [<ffffffff8116d21e>] oom_kill_process+0x24e/0x3b0
Mar 26 04:39:11 foreman kernel: [<ffffffff81088e4e>] ? has_capability_noaudit+0x1e/0x30
Mar 26 04:39:12 foreman kernel: [<ffffffff8116da46>] out_of_memory+0x4b6/0x4f0
Mar 26 04:39:12 foreman kernel: [<ffffffff81173c36>] __alloc_pages_nodemask+0xaa6/0xba0
Mar 26 04:39:12 foreman kernel: [<ffffffff811b7f9a>] alloc_pages_vma+0x9a/0x150
Mar 26 04:39:12 foreman kernel: [<ffffffff81197b45>] handle_mm_fault+0xba5/0xf80
Mar 26 04:39:12 foreman kernel: [<ffffffff81641f00>] __do_page_fault+0x150/0x450
Mar 26 04:39:12 foreman kernel: [<ffffffff81642223>] do_page_fault+0x23/0x80
Mar 26 04:39:12 foreman kernel: [<ffffffff8163e508>] page_fault+0x28/0x30

```

#14 - 03/29/2017 12:56 PM - Will Foster

Final update here, looks like our issue was not related to memory leaks or passenger but instead a tremendous amount of new interfaces being collected via Puppet facts and then pegging out Foreman to update host entries. We had upwards of 30,000 interfaces picked up across 48 x hosts.

We took the following steps to cull this and we're running a lengthy hammer process to remove them all.

Under Settings -> Provisioning:

```

Ignore interfaces with matching identifier [ lo, usb*, vnet*, macvtap*, _vdsmdummy_, docker*, veth*, ens*, vir
br*, br* ]
Ignore Puppet facts for provisioning = true

```