Foreman - Bug #14854

Libvirt connection leaks

04/27/2016 12:08 PM - Thomas McKay

Status:	Closed		
Priority:	Normal		
Assignee:	Lukas Zapletal		
Category:	Compute resources - libvirt		
Target version:			
Difficulty:		**Fixed in Releases:**	3.1.0
Triaged:	Yes	**Found in Releases:**	
Bugzilla link:	1980166	**Red Hat JIRA:**	
Pull request:	https://github.com/theforeman/foreman/pull/8652		

Description

After a morning of provisioning VMs on libvirt where the VMs failed to install packages (could not fetch glibc-common rpm so anaconda failed to finish), the libvirt compute resource became unreachable with the message in UI of

```
Call to virConnectOpen failed: End of file while reading data: Ncat: Connection reset by peer.: Input/output error
```

/var/log/messages on the libvirt host indicated

```
libvirtd[26937]: Too many active clients (20), dropping connection from 127.0.0.1;0
```

Restarting the rails server freed up the connections and the compute resource was usable again.

It appears that libvirt connections are being held onto by the server.

Related issues:

Related to Smart Proxy - Bug #14880: Libvirt connection leaks	**Rejected**
Related to Foreman - Bug #6405: Failure to reconnect on libvirtd restart	**New**

Associated revisions

Revision 264e4a70 - 10/12/2021 12:32 PM - Lukas Zapletal

Fixes #14854 - connection libvirt leak fixed

History

#1 - 04/27/2016 12:10 PM - Thomas McKay

I used this command to watch the connection count during debug

```
sudo netstat -anp | grep libvirt | wc -l
```

#2 - 04/27/2016 12:48 PM - Thomas McKay

A note on setup. This is katello running in a VM on laptop's libvirt. The compute resource is: qemu+ssh://root@thomasmckay.usersys.redhat.com/system

#3 - 04/29/2016 03:28 AM - Lukas Zapletal

Thanks Tom, good observation. It looks like the behavior is same for local sockets. And the same problem is for foreman proxy libvirt provider. I will add new ticket and fix that first, then I will take look on this one.

#4 - 04/29/2016 07:09 AM - Lukas Zapletal

- Subject changed from libvirt connections not closed - Call to virConnectOpen failed: End of file while reading data: Ncat: Connection reset by peer.: Input/output error to Libvirt connection leaks

I think the best way to handle this is to create SimpleConnectionManager that will provide a block opening and closing connection automatically. Then we need to rewrite all our code to blocks.

This opens up doors for implementing PooledConnectionManager (e.g. via https://github.com/mperham/connection_pool gem) later on, so we can re-use connections. The gem assumes self-healing connections which is not the case for libvirt, so the manager need to implement "ping" check before every call and heal the broken connections. For libvirt this can be implemented via fog with the "get_node_info" call which raises

```
Libvirt::RetrieveError: Call to virNodeGetInfo failed: internal error: client socket is closed
```

on broken connections.

#5 - 04/29/2016 07:12 AM - Lukas Zapletal

- Related to Bug #14880: Libvirt connection leaks added

#6 - 10/25/2018 11:08 AM - Lukas Zapletal

- Related to Bug #6405: Failure to reconnect on libvirtd restart added

#7 - 07/08/2021 12:14 PM - Lukas Zapletal

- Triaged changed from No to Yes

Please increase the following values

#max_anonymous_clients = 20
#max_workers = 20

in /etc/libvirt/libvirtd.conf and restart libvirtd daemon. The libvirt daemon in RHEL is not configured for heavy concurrent client use, we generally do recomment oVirt or Red Hat Enterprise virtualization for enterprise workloads.

#8 - 07/08/2021 02:03 PM - The Foreman Bot

- Status changed from New to Ready For Testing

- Assignee set to Lukas Zapletal

- Pull request https://github.com/theforeman/foreman/pull/8652 added

#9 - 07/08/2021 02:06 PM - Lukas Zapletal

- Bugzilla link set to 1980166

#10 - 10/12/2021 12:32 PM - The Foreman Bot

- Fixed in Releases 3.1.0 added

#11 - 10/12/2021 01:01 PM - Lukas Zapletal

- Status changed from Ready For Testing to Closed

Applied in changeset foreman|264e4a700df85417818ece54b0cc9196952970e0.