Pulp - Issue #1771

requests or urllib3 can't read a file which causes Nectar to fail mysteriously

03/14/2016 07:13 PM - jcline@redhat.com

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	rbarlow	**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:			
Severity:	3. High	**Groomed:**	No
Version:	2.8.0	**Sprint Candidate:**	No
Platform Release:	2.8.3	**Tags:**	Pulp 2
OS:		**Sprint:**	Sprint 1
Triaged:	Yes	**Quarter:**	

Description

Katello encountered this traceback while lazy-syncing while syncing a whole bunch of repos.

```
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: requests.packages.urllib3.connectionpool:INFO: Resetting dropped connection: cdn.redhat.com
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.threaded:WARNING: Skipping requests to cdn.redhat.com due to repeated connection failures: [Errno 2] No such file or directory
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048) u'response_code'
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048) Traceback (most recent call last):
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048)   File "/usr/lib/python2.7/site-packages/nectar/downloaders/base.py", line 145, in _fire_event_to_listener
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048)     event_listener_callback(*args, **kwargs)
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048)   File "/usr/lib/python2.7/site-packages/pulp/streamer/server.py", line 101, in download_failed
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048)     '%%(response_msg)s' %% error_report))
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: nectar.downloaders.base:ERROR: (29271-90048) KeyError: u'response_code'
Mar 14 08:22:26 sat-r220-04.lab.eng.rdu2.redhat.com pulp_streamer[29271]: [-] 127.0.0.1 - - [14/Mar/2016:12:22:25 +0000] "GET /var/lib/pulp/content/units/distribution/49/da05e1e526ca2cefec9985aea4c85ba2d1a04e167ecf1acbe50b0063fc30c9/images/pxeboot/vmlinuz HTTP/1.1" 200 - "-" "Wget/1.17.1 (linux-gnu)"
```

Note the ConnectionError was caused by "[Errno 2] No such file or directory". After some research (see notes below) this is due to https://github.com/kennethreitz/requests/issues/2863.

The plan is to stop Nectar from writing temporary certificates to disk and then deleting them, as this will not work even if requests #2863 got fixed. Instead, they should be stored permanently on disk (/var/lib/pulp/pki or similar). In addition to this, a second issue has been filed to add a work-around to the streamer (https://pulp.plan.io/issues/1788). Note that the problem of stale CAs, client certificates, and client keys will still be present on all our downloading components, but it is unlikely they will experience a problem.

Associated revisions

Revision 0e2560e0 - 03/30/2016 10:58 PM - rbarlow

Streamer download requests use permanently stored pem files.

This commit begins a shift towards storing repository PEM data (CA certificates, client certificates, and client keys) on the filesystem, and passing paths to those certs to Nectar rather than allowing nectar to write them in temporary files.

This commit provides the infrastructure for this change, but only converts the streamer to using it. All other downloads in Pulp still take the old, now deprecated path.

https://pulp.plan.io/issues/1771

fixes #1771

Revision 0e2560e0 - 03/30/2016 10:58 PM - rbarlow

Streamer download requests use permanently stored pem files.

This commit begins a shift towards storing repository PEM data (CA certificates, client certificates, and client keys) on the filesystem, and passing paths to those certs to Nectar rather than allowing nectar to write them in temporary files.

This commit provides the infrastructure for this change, but only converts the streamer to using it. All other downloads in Pulp still take the old, now deprecated path.

https://pulp.plan.io/issues/1771

fixes #1771

Revision b9307f58 - 04/04/2016 09:17 PM - bmbouter

Adds apache_manage_sys_content_rw to pulp-streamer SELinux policy

re #1771 https://pulp.plan.io/issues/1771

Revision b9307f58 - 04/04/2016 09:17 PM - bmbouter

Adds apache_manage_sys_content_rw to pulp-streamer SELinux policy

re #1771 https://pulp.plan.io/issues/1771

Revision f634c21b - 04/11/2016 09:19 PM - rbarlow

Add a migration that writes Importer TLS files to local storage. (#2505)

https://pulp.plan.io/issues/1771

fixes #1771

Revision f634c21b - 04/11/2016 09:19 PM - rbarlow

Add a migration that writes Importer TLS files to local storage. (#2505)

https://pulp.plan.io/issues/1771

fixes #1771

History

#1 - 03/16/2016 09:50 PM - jcline@redhat.com

- Status changed from NEW to ASSIGNED

- Assignee set to jcline@redhat.com

I've been digging into this for most of the day today, so I might as well assign it to myself.

Here's what I've found so far (mostly this is for my own benefit tomorrow morning).

To reproduce this, I believe you need to do the following:

1. Start or restart the pulp_streamer
2. Make a request directly to the streamer for a file that requires a client certificate: something like `` http://127.0.0.1:8751/var/lib/pulp/content/units/rpm/hex\[:2\]/hex\[2:\]/filename.rpm\`\`
3. Wait. The TCP socket needs to die. Give it 5-10 minutes.
4. Make a second request to the streamer (it can be for the exact same file). This will fail with "No such file or directory".

My current theory: urllib3 uses connection pooling (which is great!). However, while the connection sits in the pool, it may get terminated by the server when it hits a certain age. When this happens, urllib3 notes this when it pulls its connection object out of the pool and to fix this problem it restarts the connection. However, it **doesn't** seem to reconfigure client certificates or CA certificates. This is problematic because when it restarts the connection, it needs to re-check the server's certificate.

Nectar writes the CA and client certificates to /tmp for each NectarConfig. When the life of the NectarConfig is over, finalize is called to remove these temporary files. If that doesn't happen, we leak 3 files per request to a protected repository. The problem, of course, is that after a request is over, we destroy the CA and client certificate. This is fine initially because they're only used when the connection is first established, but when it dies they're needed again. They aren't there and urllib3 explodes. Womp womp.

I haven't confirmed this theory yet, but that's my first task tomorrow morning.

#2 - 03/16/2016 10:13 PM - mhrivnak

Assuming this theory is true, I wonder if it's possible to have urrlib3's connection pool behave such that it does not attempt to automatically re-establish a dropped connection, but rather waits for another request that wants to use that connection.

#3 - 03/17/2016 06:24 PM - jcline@redhat.com

Okay, so I've done more digging. I filed a report upstream[0] which I recommend anyone who wants to understand the problem read. Sadly, it was a known issue[1] I just missed because I was looking for CA-related problems. It turns out we've got more than one problem:

1. Using one Session for downloads that require more than one client certificate across them (quite likely in the streamer) will not work, no matter what we do to Nectar.
2. Using one Session across Nectar downloads is not currently possible because of the way Nectar configures the Session.

Here are some potential solutions I've thought of:

1. Drop connection pooling on the streamer and continue to use Nectar. This may or may not result in frequent download failures with the streamer, and it will certainly hurt its performance in a very, very big way.
2. Modify Nectar in some way to avoid writing the certificates and keys to temporary files repeatedly. I don't think Nectar should be responsible for this task though, and getting it done reasonably in the platform may or may not be easy.
3. Stop using Nectar in the streamer and do either 3a or 3b.
3a. configure requests directly and manage where we put the certs and keys so that we don't write and delete them for every request AND also make sure we clear the connection pool any time a new client cert/key gets used. We'll get some connection pooling, but not as much.
3b. Use the library provided by twisted instead of Nectar or requests.
4. Change how CA certificates and client certificates/keys are stored in Pulp, and until the issue is addressed upstream create and manage the lifecycle of a pool of Session objects in the streamer.

[0] https://github.com/kennethreitz/requests/issues/3058
[1] https://github.com/kennethreitz/requests/issues/2863

#4 - 03/17/2016 08:03 PM - mhrivnak

It sounds like no matter what we do, we have to start storing ssl certs (both CAs and client certs) on the filesystem in a permanent location, so it can be referenced and re-referenced at any time on the filesystem by urllib3.

There is the additional problem that urllib3 does not account for different ssl settings (certs included) when managing its connection pool. Here is a potential short-term solution that could be useful until the problem gets fixed in urllib3. (This is number 4 on the list above.)

We could manage our own pool of session objects, and create a unique session for each unique collection of ssl settings. It would go something like this:

1. Create a namedtuple that contains a field for each ssl setting that we manage.
2. Create a dictionary where keys are instances of the above namedtuple, and values are a Session object
3. When making a request, create a namedtuple with the current ssl settings, use that to find-or-create a Session object, and then use it.
4. Something would need to keep track of how long it's been since a particular session was used, and evict old ones from the pool. That might require some cleanup, but perhaps it is sufficient to just let normal garbage collection happen.

#5 - 03/17/2016 09:32 PM - jcline@redhat.com

I've mulled on this a bit more and I've thought of another thing we need to consider when managing our own pool. What happens when a client asks for a big file? I'm talking several gigabytes or more. How long do we keep Sessions in the pool? An hour without new requests? A day? I can't think of a good way to figure out if there is an active download in the Session so we risk killing long-running downloads.

I'm not saying there's not a solution to that problem, but it's a problem.

#6 - 03/18/2016 02:43 PM - mhrivnak

- Priority changed from Normal to High

- Severity changed from 2. Medium to 3. High

- Platform Release set to 2.8.1

- Triaged changed from No to Yes

#7 - 03/18/2016 03:25 PM - mhrivnak

Good point. The way I'm hoping we can address that is by not killing the session when we remove it from the pool. The key question is:

When you're done using a session, do you have to kill it or otherwise do cleanup? Or can you just let it fall out of scope, and let normal garbage collection deal with it? The latter is the far more likely case, by virtue of being pythonic.

If that's correct, then we can just remove a session from the pool, and any existing requests using it will finish up without a problem.

In theory we could end up with a small number of redundant Session objects this way, but I don't think it's enough that it would be likely to cause problems.

#8 - 03/18/2016 06:53 PM - jcline@redhat.com

- Priority changed from High to Normal

- Severity changed from 3. High to 2. Medium

- Platform Release deleted (2.8.1)

- Triaged changed from Yes to No

There is cleanup (a call to `close()` on the session). It looks like it closes each adapter, which in turn cleans up the connection pools.

We could let it drop out of scope without performing cleanup, but I think we would be leaking TCP connections. Now, these *should* be cleaned up eventually since HTTP servers have a Keep-Alive timeout (Apache does, at any rate). These timeout values are usually fairly low (a few seconds to a few minutes), but of course they are configurable so potentially it could be much longer.

#9 - 03/18/2016 06:53 PM - jcline@redhat.com

- Priority changed from Normal to High

- Severity changed from 2. Medium to 3. High

- Platform Release set to 2.8.1

- Triaged changed from No to Yes

#10 - 03/18/2016 07:04 PM - mhrivnak

I see that it has a close() method, but nothing says it has to be called, and the examples don't call it. We should get confirmation of 1) do we have to call close(), and 2) what happens if it just falls out of scope when it's not being used anymore, perhaps by asking them.

#11 - 03/18/2016 07:24 PM - jcline@redhat.com

Well, not to be too pedantic, but nothing says you need to close files or anything else, but I don't think that's a good reason to not do it. They've implemented the `__enter__` and `__exit__` methods, so it's safe to assume (I think) that users should either use `with` or call `close`.

Luckily, I've taken a look at how urllib3 cleans up and it looks like it lets connections that are in use finish up before closing shop, so we can probably safely call `close` on a session, let it drop out of scope, and carry on.

#12 - 03/18/2016 07:38 PM - mhrivnak

That sounds good. If there are any questions remaining about whether it is safe to remove a session from a pool, do some cleanup, and let existing requests finish gracefully, let's get in touch with upstream and work with them to get firm answers. Hopefully we won't need to make any assumptions.

#13 - 03/21/2016 02:18 PM - jcline@redhat.com

- Description updated

- Status changed from ASSIGNED to NEW

- Assignee deleted (jcline@redhat.com)

#14 - 03/21/2016 04:42 PM - rbarlow

- Status changed from NEW to ASSIGNED

- Assignee set to rbarlow

#15 - 03/21/2016 04:43 PM - rbarlow

Should we s/mysteriously/miserably/ on the subject?

#20 - 03/23/2016 07:54 PM - semyers

- Platform Release changed from 2.8.1 to 2.8.2

#21 - 03/28/2016 07:32 PM - rbarlow

This PR isn't done so I'm not marking it POST yet, but it lives here:

https://github.com/pulp/pulp/pull/2499

#22 - 03/28/2016 11:07 PM - jortel@redhat.com

mhrivnak wrote:

> I see that it has a close() method, but nothing says it has to be called, and the examples don't call it. We should get confirmation of 1) do we have to call close(), and 2) what happens if it just falls out of scope when it's not being used anymore, perhaps by asking them.

Looks like nectar is just letting sessions fall out of scope and not closing the session explicitly. Since no connection leaks have been observed, I think it's safe to assume that the session pooling approach can do the same.

#23 - 03/29/2016 06:53 PM - rbarlow

- Project changed from Nectar to Pulp

#24 - 03/30/2016 05:40 PM - rbarlow

- Status changed from ASSIGNED to POST

#25 - 03/30/2016 11:08 PM - rbarlow

- Status changed from POST to MODIFIED

- % Done changed from 0 to 100

Applied in changeset pulp|0e2560e045dad843950461affad51d69c4b9f9d2.

#26 - 03/31/2016 12:24 AM - mmccune@redhat.com

- *Version set to 2.8.0*

I get this error trying to utilize the patch in the PR on a yum install from a client:

```
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) An unexpected error occurred while handling the request.
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) Traceback (most recent call last):
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) File
"/usr/lib/python2.7/site-packages/pulp/streamer/server.py", line 184, in _handle_get
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) self._download(catalog_entry, request, responder)
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) File
"/usr/lib/python2.7/site-packages/pulp/streamer/server.py", line 212, in _download
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) importer, config =
repo_controller.get_importer_by_id(catalog_entry.importer_id)
Mar 30 18:22:29 sat-r220-03 pulp_streamer: pulp.streamer.server:ERROR: (2428-46784) ValueError: too many values to unpack
Mar 30 18:22:29 sat-r220-03 pulp_streamer: [-] 127.0.0.1 - - [30/Mar/2016:22:22:28 +0000] "GET
/var/lib/pulp/content/units/rpm/81/69d22ce297a7688847ec8a30365b6055fec4b9613ae76bcc762953f8ac1a95/rpcbind-0.2.0-33.el7_2.x86_64.rpm
HTTP/1.1" 500 - "-" "urlgrabber/3.10 yum/3.4.3"
```

#27 - 03/31/2016 03:14 AM - jcline@redhat.com

I suspect you have an old version of the patch. The merged PR has 3 values (the proper amount) instead of the 2 in your traceback: https://github.com/pulp/pulp/pull/2499/files#diff-0401aafbc062d63f501d2648d64ae307L211

#28 - 03/31/2016 05:42 AM - mmccune@redhat.com

Ok, got passed the above error, thanks for the pointer.

Now I'm getting:

```
pulp.streamer.server:ERROR: (25128-63488) An unexpected error occurred while handling the request.
pulp.streamer.server:ERROR: (25128-63488) Traceback (most recent call last):
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/pulp/streamer/server.py", l
ine 184, in _handle_get
pulp.streamer.server:ERROR: (25128-63488)     self._download(catalog_entry, request, responder)
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/pulp/streamer/server.py", l
ine 215, in _download
pulp.streamer.server:ERROR: (25128-63488)     db_importer, catalog_entry.url, working_dir='/tmp')
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/pulp/plugins/importer.py",
line 82, in get_downloader_for_db_importer
pulp.streamer.server:ERROR: (25128-63488)     nectar_config = importer_to_nectar_config(importer, working_dir=
working_dir)
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/pulp/plugins/util/nectar_co
nfig.py", line 66, in importer_to_nectar_config
pulp.streamer.server:ERROR: (25128-63488)     return importer_config_to_nectar_config(config, working_dir, dow
nload_config_kwargs)
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/pulp/plugins/util/nectar_co
nfig.py", line 97, in importer_config_to_nectar_config
pulp.streamer.server:ERROR: (25128-63488)     download_config = DownloaderConfig(**download_config_kwargs)
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/nectar/config.py", line 136
, in __init__
pulp.streamer.server:ERROR: (25128-63488)     self._process_ssl_settings()
pulp.streamer.server:ERROR: (25128-63488)   File "/usr/lib/python2.7/site-packages/nectar/config.py", line 165
, in _process_ssl_settings
```

```
pulp.streamer.server:ERROR: (25128-63488)     raise AttributeError('Cannot read file: %%s' %% file_arg_value)
pulp.streamer.server:ERROR: (25128-63488) AttributeError: Cannot read file: /var/lib/pulp/importers/mmccune-or
g-Red_Hat_Enterprise_Linux_Server-Red_Hat_Enterprise_Linux_7_Server_RPMs_x86_64_7Server-yum_importer/pki/ca.cr
t
[-] 127.0.0.1 - - [31/Mar/2016:03:38:22 +0000] "GET /var/lib/pulp/content/units/rpm/a5/6a0f4a22ca8b911e4d01856
83aee7d1e75f9a85ad0231eccbd5f845a2ef6d0/yum-utils-1.1.31-34.el7.noarch.rpm HTTP/1.1" 500 - "-" "urlgrabber/3.1
0 yum/3.4.3"
```

is there some sort of migration I'm missing to get these ca.crt files on disk?

#29 - 03/31/2016 06:16 AM - mmccune@redhat.com

a bit more info, I setup new repositories after applying the new code, this time it creates the files:

```
# ls -Z /var/lib/pulp/importers/neworg-3-Red_Hat_Enterprise_Linux_Server-Red_Hat_Satellite_Tools_6_1_for_RHEL_
7_Server_RPMs_x86_64-yum_importer/pki/
-rw-------. apache apache system_u:object_r:httpd_sys_rw_content_t:s0 ca.crt
-rw-------. apache apache system_u:object_r:httpd_sys_rw_content_t:s0 client.crt
-rw-------. apache apache system_u:object_r:httpd_sys_rw_content_t:s0 client.key
```

but still get the same error as above:

```
 AttributeError: Cannot read file: /var/lib/pulp/importers/neworg-3-Red_Hat_Enterprise_Linux_Server-Red_Hat_Sa
tellite_Tools_6_1_for_RHEL_7_Server_RPMs_x86_64-yum_importer/pki/ca.crt
```

#30 - 03/31/2016 07:00 AM - mmccune@redhat.com

after re-restarting httpd and pulp_streamer as well as setting up new repositories I was able to lazy fetch content with this new routine without error.

that said, I'd imagine we need some sort of migration to populate the /pki dir for existing repos?

#31 - 03/31/2016 02:26 PM - jcline@redhat.com

I suppose it couldn't hurt to have a migration. The files are created when the importer is saved (and cleaned up when the importer is deleted) so even if you turned an existing repository into an on-demand one I believe the files will get created when that change is saved. The only case I can think of is people out there who already have lazy repositories (some community users, maybe) and don't change any configuration after upgrading from 2.8.z to 2.8.2

#32 - 03/31/2016 03:13 PM - rbarlow

- Status changed from MODIFIED to ASSIGNED

I'll create a migration to loop over all Importers and save their certs. Thanks for the feedback!

#33 - 03/31/2016 04:43 PM - mhrivnak

- Sprint/Milestone set to 19

#34 - 03/31/2016 07:14 PM - pthomas@redhat.com

Randy,
Could you also add detailed steps on how to verify this.
And point me to any pulp-smash test you might have written for this.

Thanks

#35 - 03/31/2016 09:58 PM - rbarlow

- File 0001-Add-a-migration-that-writes-Importer-TLS-files-to-lo.patch added

Hello Mike!

I'm attaching a patch to this ticket that contains a very simple migration that will get you unblocked. This patch will not be applied to upstream Pulp, as it is not future proof (i.e. it depends on specific behaviors of the code it imports that might not be the same in future versions of Pulp). For upstream, I will need to write a migration that assumes less about the Importer model's code.

I have this patch in a work-in-progress PR here, but I will soon change that PR to reflect the real change that needs to happen upstream:

https://github.com/pulp/pulp/pull/2505

#36 - 04/01/2016 03:05 PM - rbarlow

Preethi,

I just followed the steps that Jeremy listed in the first comment. I ended up creating and on-demand syncing the rhel-6-server repository. Once that was done, I looked in MongoDB for the storage path of an RPM. I ended up using 389-ds-base-libs, and requested it from the streamer:

```
curl -O http://127.0.0.1:8751/var/lib/pulp/content/units/rpm/66/0e7bf17e1bd386dcd61759fd2822cb01ab396b13d69f33a3e6b5d67ce29ff6/389-ds-base-libs-1.2.10.2-15.el6.i686.rpm
```

Then I waited a while (tea time!) and came back and did the same thing. Before the fix it should fail because it can't read the certificates/key, but with the fix it should still work.

#37 - 04/04/2016 10:58 PM - rbarlow

- Status changed from ASSIGNED to POST

The migration is complete to my satisfaction now, and is ready for review.

https://github.com/pulp/pulp/pull/2505

#38 - 04/06/2016 10:24 PM - semyers

- Platform Release changed from 2.8.2 to 2.8.3

#39 - 04/11/2016 09:20 PM - rbarlow

- Status changed from POST to MODIFIED

Applied in changeset pulp|f634c21b4881634745bfbb2a1200dcf390291d91.

#41 - 04/27/2016 12:38 AM - semyers

- Status changed from MODIFIED to 5

#42 - 05/09/2016 10:28 PM - pthomas@redhat.com

verified

```
[root@ibm-x3550m3-09 ~]# curl -O http://127.0.0.1:8751/var/lib/pulp/content/units/rpm/66/0e7bf17e1bd386dcd6175
9fd2822cb01ab396b13d69f33a3e6b5d67ce29ff6/389-ds-base-libs-1.2.10.2-15.el6.i686.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  377k  100  377k    0     0   363k      0  0:00:01  0:00:01 --:--:--  364k
[root@ibm-x3550m3-09 ~]# 
[root@ibm-x3550m3-09 ~]# 
[root@ibm-x3550m3-09 ~]# curl -O http://127.0.0.1:8751/var/lib/pulp/content/units/rpm/66/0e7bf17e1bd386dcd6175
9fd2822cb01ab396b13d69f33a3e6b5d67ce29ff6/389-ds-base-libs-1.2.10.2-15.el6.i686.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  377k  100  377k    0     0   257k      0  0:00:01  0:00:01 --:--:--  257k
[root@ibm-x3550m3-09 ~]# 
[root@ibm-x3550m3-09 ~]# 
[root@ibm-x3550m3-09 ~]# curl -O http://127.0.0.1:8751/var/lib/pulp/content/units/rpm/66/0e7bf17e1bd386dcd6175
9fd2822cb01ab396b13d69f33a3e6b5d67ce29ff6/389-ds-base-libs-1.2.10.2-15.el6.i686.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  377k  100  377k    0     0   571k      0 --:--:-- --:--:-- --:--:--  571k
[root@ibm-x3550m3-09 ~]# 
[root@ibm-x3550m3-09 ~]# 
[root@ibm-x3550m3-09 ~]# curl -O http://127.0.0.1:8751/var/lib/pulp/content/units/rpm/66/0e7bf17e1bd386dcd6175
9fd2822cb01ab396b13d69f33a3e6b5d67ce29ff6/389-ds-base-libs-1.2.10.2-15.el6.i686.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  377k  100  377k    0     0   302k      0  0:00:01  0:00:01 --:--:--  303k
```

#43 - 05/09/2016 10:28 PM - pthomas@redhat.com

- Status changed from 5 to 6

#44 - 05/17/2016 09:31 PM - semyers

- Status changed from 6 to CLOSED - CURRENTRELEASE

#48 - 03/08/2018 07:21 PM - bmbouter

- Sprint set to Sprint 1

#49 - 03/08/2018 07:21 PM - bmbouter

- Sprint/Milestone deleted (19)

#50 - 04/15/2019 10:33 PM - bmbouter

- Tags Pulp 2 added

Files

0001-Add-a-migration-that-writes-Importer-TLS-files-to-lo.patch	1.12 KB	03/31/2016	rbarlow