

Docker Support - Issue #1375

'latest' tag out-dated in Crane Docker Registry V2 API

11/16/2015 06:53 PM - twaugh

Status:	CLOSED - CURRENTRELEASE	Start date:	
Priority:	High	Due date:	
Assignee:	rbarlow	Estimated time:	0:00 hour
Category:			
Sprint/Milestone:			
Severity:	2. Medium	Groomed:	No
Version - Docker:	master	Sprint Candidate:	No
Platform Release:	2.8.0	Tags:	Pulp 2
Target Release - Docker:	2.0.0	Sprint:	
OS:		Quarter:	
Triaged:	Yes		
Description			
<p>I've observed <code>/v2/<repo>/manifests/latest</code> listing out-dated blobs compared to the equivalent tag on the 'distribution' used for sync.</p> <p>I had previously created several manifests for the same set of blobs, with tags including '1.0', 'latest', and a unique tag. After this, a pulp sync was started.</p> <p>After the sync finished, more manifests were created, this time for a new set of blobs, with tags including '1.0', 'latest', and another unique tag.</p> <p>After another pulp sync was finished, <code>/v2/<repo>/manifests/latest</code> from Crane was showing the older manifest (incorrect), although <code>/v2/<repo>/manifests/1.0</code> was showing the new manifest (correct).</p>			
Related issues:			
Related to Docker Support - Story #1460: As a user, I can upload Tag units		CLOSED - CURRENTRELEASE	

Associated revisions

Revision `ac41d4fd` - 01/11/2016 04:50 PM - rbarlow

Pass the updated Mongoengine units to `Importer.remove_units()`.

The Docker Importer implements the `remove_units()` method, but the platform was still passing the old `RepoAssociation` objects instead of the proper Mongoengine units. This commit detects converted plugins and passes them the new units.

<https://pulp.plan.io/issues/1375>

re #1375

Revision `ac41d4fd` - 01/11/2016 04:50 PM - rbarlow

Pass the updated Mongoengine units to `Importer.remove_units()`.

The Docker Importer implements the `remove_units()` method, but the platform was still passing the old `RepoAssociation` objects instead of the proper Mongoengine units. This commit detects converted plugins and passes them the new units.

<https://pulp.plan.io/issues/1375>

re #1375

Revision `91e09ca0` - 01/11/2016 07:14 PM - rbarlow

Add a formal Tag Unit model to track repository tags.

This commit creates a new Tag model that is used to mark which Manifests are tagged in a repository. Users can upload, synchronize, and associate tags which allows them greater control over tags than was afforded in the v1 code.

<https://pulp.plan.io/issues/1375>

fixes #1375

Revision 91e09ca0 - 01/11/2016 07:14 PM - rbarlow

Add a formal Tag Unit model to track repository tags.

This commit creates a new Tag model that is used to mark which Manifests are tagged in a repository. Users can upload, synchronize, and associate tags which allows them greater control over tags than was afforded in the v1 code.

<https://pulp.plan.io/issues/1375>

fixes #1375

Revision 91e09ca0 - 01/11/2016 07:14 PM - rbarlow

Add a formal Tag Unit model to track repository tags.

This commit creates a new Tag model that is used to mark which Manifests are tagged in a repository. Users can upload, synchronize, and associate tags which allows them greater control over tags than was afforded in the v1 code.

<https://pulp.plan.io/issues/1375>

fixes #1375

Revision 91e09ca0 - 01/11/2016 07:14 PM - rbarlow

Add a formal Tag Unit model to track repository tags.

This commit creates a new Tag model that is used to mark which Manifests are tagged in a repository. Users can upload, synchronize, and associate tags which allows them greater control over tags than was afforded in the v1 code.

<https://pulp.plan.io/issues/1375>

fixes #1375

History

#1 - 11/16/2015 06:58 PM - rbarlow

- Project changed from Pulp to Docker Support
- Priority changed from Normal to High
- Version - Docker set to master
- Target Release - Docker set to 2.0.0
- Tags Easy Fix added

It's likely that deleting these lines will solve this issue:

https://github.com/pulp/pulp_docker/blob/docker_v2_api/plugins/pulp_docker/plugins/distributors/publish_steps.py#L135-L137

#2 - 11/18/2015 12:31 PM - tvaugh

The suggested change didn't fix the problem.

/v2/<repo>/manifests/latest from Crane disagrees with the Docker Distribution instance after a 'sync' and 'publish'.

Removing those lines did change which 'latest' I saw, but it still wasn't the right one. :-)

#3 - 11/20/2015 04:45 PM - mhrivnak

- Triaged changed from No to Yes

#4 - 11/20/2015 07:39 PM - rbarlow

- Tags deleted (Easy Fix)

tvaugh wrote:

The suggested change didn't fix the problem.

/v2/<repo>/manifests/latest from Crane disagrees with the Docker Distribution instance after a 'sync' and 'publish'.

You will need to also remove the images from the repository and then resync and republish. The code change prevents the data from ending up in this state in the future, but it will not fix data that is already in this state.

This code will not be our long term fix for the issue, but I do believe it will get you past your immediate hurdle. The problem that this code does not address is what would happen if the feed URL had "rolled back" a Manifest (i.e., an older manifest that had been latest and was replaced became latest again). To make that work we are going to need to track the tags in a more formal way than we are doing at this time.

#5 - 11/24/2015 02:38 PM - ttomecek

rbarlow wrote:

You will need to also remove the images from the repository and then resync and republish.

I really don't understand why the operation is called sync then. As I understand synchronization it should be an operation which makes content of two repositories identical -- something like `rsync --delete`. Why should be users responsible for managing the actual synchronization?

The way I would solve this is to make the process configurable:

- add an option to control overwriting
- add an option to delete content

#6 - 11/24/2015 04:51 PM - mhrivnak

I understand how that term could be interpreted differently. Perhaps a different word than "sync" should have been chosen, but the choice was made years ago, and we're not in a position right now to change it.

Pulp's sync workflow will continue to be additive by default. When new content appears in the remote feed, pulp will add it locally. When remote content disappears, pulp will retain that content. This is an important behavior that our users depend on.

Some content types have additional options to control and modify this behavior, and we're happy to consider adding something like that to the docker sync.

#7 - 11/24/2015 08:07 PM - ttomecek

mhrivnak wrote:

Pulp's sync workflow will continue to be additive by default. When new content appears in the remote feed, pulp will add it locally. When remote content disappears, pulp will retain that content. This is an important behavior that our users depend on.

Okay, I understand that if this feature is in pulp for years.

It would be really beneficial for us to have those two options implemented. Especially the content overwriting -- the reason why Tim created this issue.

#8 - 12/03/2015 04:55 PM - acarter@redhat.com

Checking in on this issue. Can you confirm that the changes requested will be prioritized or let us know if we need to have more conversations? This will be a blocker for Chain Rebuilds which is required by product management asap.

#9 - 12/03/2015 09:14 PM - rbarlow

acarter@redhat.com wrote:

Checking in on this issue. Can you confirm that the changes requested will be prioritized or let us know if we need to have more conversations?

It's high on my priority list. I will probably get to it in the next two weeks. In the meantime I believe the deleting those lines will work around the issue, though I will likely solve the problem a different way.

#10 - 12/21/2015 04:50 PM - rbarlow

- Status changed from NEW to ASSIGNED

- Assignee set to rbarlow

I have developed a plan to solve this issue. I will introduce a new type of Unit called Tag that will have three attributes: repo_id, name, manifest, and version. version will be 2 (representing Docker v2) for now. The repo_id and name fields will have a uniqueness constraint so that we can guarantee that a Tag only exists once per repository. The manifest will be a Manifest digest.

During a sync, whatever tags are found in the remote repository will either get created in the local repository (if missing), or the digests will be updated to reflect the remote digests (if already present). Any Tag objects in the local repository that are not found in the remote repository will remain for now (to be consistent with the defaults of other Pulp plugins, which do not remove data during syncs). We can consider adding a feature later to remove missing Tags, Manifests, and Blobs, similar to pulp_rpm's --remove-missing feature.

When users perform copy operations, it is the user's choice to copy a Tag, a Manifest, or a Blob. If the user wishes to promote a Tag from one repository to another they would simply use Pulp's copy feature. The Manifest referenced by the Tag will also automatically be copied, as well as any Blobs that it references. Any Manifest copied will recursively copy referenced Blobs as it does today, but it will not copy any tags. Copying a Blob will

not copy any Manifests at all, since the Blob is the lowest level Unit in Docker v2.

Another advantage of this design is that it will allow users to upload Tag units, which is an ability that was missing with our Docker v1 support. This means that existing Manifests in Pulp can be tagged by the user creating a Tag unit in the repository, which will only require specifying a name, a Manifest digest, and the repository that the user wishes the Tag to be uploaded to.

I believe our users will appreciate the flexibility of this design as it allows them explicit control over the tags which was not possible with pulp_docker < 2.0.

#11 - 12/21/2015 06:44 PM - tvaugh

This sounds like the right direction to me.

With the ability to upload Tag units, why would the user need to specify a name for the tag? Just because pulp_docker doesn't parse the Manifest? They won't want to create a Tag with a name different from what the Manifest says is its tag.

Also, to clarify: the Tag units will only be for V2 repositories?

#12 - 12/21/2015 09:21 PM - rbarlow

tvaugh wrote:

With the ability to upload Tag units, why would the user need to specify a name for the tag? Just because pulp_docker doesn't parse the Manifest? They won't want to create a Tag with a name different from what the Manifest says is its tag.

Hahaha, that is a great point! I'll make it so you just need to give the digest that you want to tag, and the Tag will automatically get the name from it. My excuse is that I didn't have my coffee yet this morning when I wrote this.

Also, to clarify: the Tag units will only be for V2 repositories?

Correct, we do not wish to alter the v1 workflow if possible.

#13 - 12/23/2015 09:12 PM - rbarlow

- Related to Story #1460: As a user, I can upload Tag units added

#14 - 01/11/2016 07:15 PM - rbarlow

- Status changed from ASSIGNED to MODIFIED

- % Done changed from 0 to 100

Applied in changeset [91e09ca00d50db8866aa7bf4e955946c137d7fe4](#).

#15 - 01/11/2016 07:24 PM - rbarlow

- Status changed from MODIFIED to ASSIGNED

We need to do this again for the docker-v2-support branch, so I'm sending it back to assigned.

#16 - 01/21/2016 09:08 PM - rbarlow

- Status changed from ASSIGNED to MODIFIED

#17 - 01/21/2016 09:09 PM - rbarlow

I moved this back to modified, as we agreed that this would only be done in upstream pulp_docker.

#18 - 02/11/2016 09:16 PM - rbarlow

- Status changed from MODIFIED to 5

#19 - 04/26/2016 06:16 PM - rbarlow

- Status changed from 5 to CLOSED - CURRENTRELEASE

- Platform Release set to 2.8.0

#20 - 04/15/2019 10:41 PM - bmbouter

- Tags Pulp 2 added