

Docker Support - Story #1209

As user, I can upload docker v2 blobs and v2 manifest

08/24/2015 10:58 AM - jluza

Status:	CLOSED - WONTFIX	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:		Tags:	Pulp 2
Platform Release:		Sprint:	
Target Release - Docker:		Quarter:	
Groomed:	No		
Sprint Candidate:	No		

Description

As user, I can upload docker v2 blobs and v2 manifest directly to pulp without performing sync operation. The solution will consists of 2 new importers: docker_blob_importer, docker_manifest_importer. Each of them allows user to upload different kind of content directly from local machine.

new types:

+BLOB_TYPE_ID = 'docker_blob'

+BLOB_IMPORTER_TYPE_ID = 'docker_blob_importer'

+BLOB_IMPORTER_CONFIG_FILE_NAME = 'server/plugins.conf.d/docker_blob_importer.json'

+MANIFEST_TYPE_ID = 'docker_manifest' # <- not sure if this will be needed

+MANIFEST_IMPORTER_TYPE_ID = 'docker_manifest_importer'

+MANIFEST_IMPORTER_CONFIG_FILE_NAME = 'server/plugins.conf.d/docker_manifest_importer.json'

docker_blob type will be implemented as models.Blob (already in the upstream code)

blob_importer import sequence (similar as in IsoImporter):

- init_unit
- move file
- save_unit

manifest_importer import steps (this is how I image it)

- parse manifest
- set specified repo attributes
- validation?

History

#1 - 08/24/2015 11:02 AM - ipanova@redhat.com

- Subject changed from [RFE] raw content importers to [RFE] As user, I can upload docker v2 blobs and v2 manifest

#2 - 08/26/2015 05:18 PM - bmbouter

- Subject changed from [RFE] As user, I can upload docker v2 blobs and v2 manifest to As user, I can upload docker v2 blobs and v2 manifest

- Description updated

#3 - 08/26/2015 05:19 PM - bmbouter

[jluza](#): I don't have much to add on the content of the story, but once this story gets enough comments on it, the next step will be to set the 'Sprint Candidate' flag to True. Then it will go through a final round of discussion w/ Pulp team before it receives the 'Groomed' flag.

#4 - 08/26/2015 06:02 PM - acarter@redhat.com

bmbouter wrote:

[jluza](#): I don't have much to add on the content of the story, but once this story gets enough comments on it, the next step will be to set the 'Sprint Candidate' flag to True. Then it will go through a final round of discussion w/ Pulp team before it receives the 'Groomed' flag.

[bmbouter](#) I think [jluza](#) primarily is looking for feedback on the design since he may need to write a patch for our internal instance well before you guys could get to this (like next week); we want to make sure that the change is something you'd be willing to pull into Pulp later

#5 - 08/27/2015 09:23 AM - jluza

acarter@redhat.com wrote:

bmbouter wrote:

[jluza](#): I don't have much to add on the content of the story, but once this story gets enough comments on it, the next step will be to set the 'Sprint Candidate' flag to True. Then it will go through a final round of discussion w/ Pulp team before it receives the 'Groomed' flag.

[bmbouter](#) I think [jluza](#) primarily is looking for feedback on the design since he may need to write a patch for our internal instance well before you guys could get to this (like next week); we want to make sure that the change is something you'd be willing to pull into Pulp later

Yes. What she said.

#6 - 08/27/2015 11:26 PM - mhrivnak

Implementation wise, this could be one importer. The upload method can accept both types and do the "right thing" for each.

I don't think this is a feature that other pulp users will want to take advantage of. It requires having a docker registry running, using some custom tool to get manifests and blobs from its API (and recording each manifest's digest somewhere, which is only provided in a custom header), and then using more custom code to upload those to pulp. I think nearly all users will find it easier to just have pulp sync from the local registry. I understand that you have a requirement to store build artifacts outside of pulp, which is what is motivating this RFE, but most pulp users do not have that requirement.

Given that, since these are plugins, you could have your own custom importer that uses ours as a base class, and implements the upload feature any way you like.

I'll again suggest that if your goal is to store the build artifact in a separate system, I would consider this workflow:

1. build an image
2. start a fresh local registry in a docker container, and mount in storage to the correct `/var/lib/...` location
3. push the image
4. stop the container
5. tar up the mounted-in storage and save it

Now any time you want to access that image, you only need to untar, and run step 2.

#9 - 08/31/2015 10:49 AM - jluza

Let's face the facts. Would it be difficult to implement this? I will give you answer: it wouldn't. Because I was able to implement blob importer in 2 days - and I've never done this before so I had to find you how plugins in pulp work. That notwithstanding, I would be one who would do the code. Would it be difficult to maintain? Well, my version of blob importer takes literally 3-5 lines of active code, everything else are just layers, constants, plugin configurations. And what is probably the most important fact, there are no new code needed for this. Everything is already in pulp_docker. We don't need anything what isn't already part of sync process. So only change is in way how pulp will get image files.

Regarding to other users interests and special tool needed for fetching the content. I'm not sure if I understood correctly your plans with pulp and it's docker support. I thought pulp - in docker scope - should be alternative for official v2 registry. I expected it will support full docker v2 API. And I can imagine that with crane, pulp can support most of the API mentioned here: <http://docs.docker.com/registry/spec/api/> But obviously uploading content to registry is not covered. The only way how to do that in pulp is via sync. But official API gives user chance to download content of image to local disk and from there upload it again to registry. As a user I would be probably disappointed pulp doesn't give me that feature.

That's why I don't understand your argument of special tool for download content. This special tool could be called curl or wget. Just because there's no cmd command in official docker client doesn't mean that users won't want to be able to do this.

#10 - 08/31/2015 05:40 PM - mhrivnak

I sense some frustration, but let's keep in mind that we all want to find the best solution. I think there is a communication gap, so I will spell out in more detail what I believe is your proposal, and what problems I see.

Pulp and crane implement a read-only docker registry API. Pulp does not support a user doing "docker push" direct to pulp. There has been extensive discussion elsewhere about why, but it comes down to this; mapping authn and authz from docker onto pulp was impossible with v1 and very difficult with v2, and in general implementing and maintaining multiple write APIs that are conceptually different is exceptionally complex. Trying to do those things in a Satellite deployment multiplies the complexity even further.

Pulp is not trying to be a drop-in replacement for the upstream docker registry. Pulp provides value as a repository management tool that lets you manage many different kinds of content with one API, and lets you achieve more advanced workflows like promotion.

Hopefully that clarifies pulp's goals with regard to docker functionality, but let me know if you have questions.

Now I'll describe what I think is your proposal. Please correct me if I'm not understanding it.

A build workflow will happen, the details of which are unimportant. A docker daemon will have the newly built image. Because the docker engine still stores content in v1 data structures, it does not create a manifest until the "push" workflow. The only way to get the manifest and blobs out of the daemon is to initiate a "push" to a running registry.

Once the manifest and blobs have been pushed to a registry, now your goal is to get the individual manifest and blob files. I am only aware of two options.

- 1) Try to retrieve them directly from the registry's filesystem. I assume this is possible, but would be wise to avoid, since their filesystem layout may not be a public API and is thus subject to change.
- 2) Use the REST API to retrieve the manifest and blobs. You will first need to know the name of the repository and the manifest's tag name. Then you do a GET for that manifest. (Note that the v2 API only provides the manifest's digest in a custom header, so you would need to save that value separately.) Now you must parse the manifest (which is JSON) to determine which blobs it references. With that parsing complete, you must do a GET request for each blob. Note that manifests sometimes reference the same blob ID more than once, which makes this workflow a bit more complex.

If you have a simpler idea in mind, please share. But as far as I can tell, you would need to write a custom tool to do one of the above options, and that tool would need to be smart enough to parse and read a manifest, then construct blob URLs.

Then you would upload each individual file to pulp, which depends on the new plugin code that you would have to write.

If you think that is the best workflow for your purposes, I am happy to support that. But I do not think that is a desirable user story for upstream pulp; an upstream pulp user would just have pulp do a sync from the registry. Pulp was designed to be pluggable so that users like you can extend existing plugins or make your own, so I would suggest doing that.

If you have a better idea, I would be very happy to hear it. Does this clarify things? Please let me know what additional questions you have.

#11 - 09/09/2015 03:15 AM - mhrivnak

- Status changed from NEW to CLOSED - WONTFIX

I think you have decided to go a different direction, so I am closing this request.

#12 - 05/26/2017 08:10 PM - tomckay@redhat.com

<http://rhelblog.redhat.com/2017/05/11/skopeo-copy-to-the-rescue/>

```
$ sudo skopeo copy docker://devel.example.com:5000/openshift3/logging-deployment:3.1.0 dir://home/vagrant/tmp/saved-images
Getting image source signatures
Copying blob sha256:9eeb1e6338117fc918ccd0e1c2096c3b6ed57307123d384d9b6a1592a2e03a86
 67.52 MB / 67.52 MB [=====]
Copying blob sha256:fe6e7d1bf96b21ee3321e555724b38a8d049cd3c391482d600eb33ac8686dd9f
 19.28 MB / 22.49 MB [=====>-----]
Copying blob sha256:597ff2da8719b2239409cf962b73a34993f6974c2837defb7637d20720e95508
 17.78 MB / 30.10 MB [=====>-----]
Copying blob sha256:093eb51f9eded16957baefff0fe18cc632e2a1f3c2cb851743ff755c4b0ec257
 54.34 MB / 54.34 MB [=====]
Writing manifest to image destination
Storing signatures
[vagrant@devel foreman]$ ls ~/tmp/saved-images/
093eb51f9eded16957baefff0fe18cc632e2a1f3c2cb851743ff755c4b0ec257.tar
597ff2da8719b2239409cf962b73a34993f6974c2837defb7637d20720e95508.tar
9eeb1e6338117fc918ccd0e1c2096c3b6ed57307123d384d9b6a1592a2e03a86.tar
fe6e7d1bf96b21ee3321e555724b38a8d049cd3c391482d600eb33ac8686dd9f.tar
manifest.json
```

Is this relevant? Could the output from skopeo be uploaded?

Is adding full push endpoint viable in current environment of containers?

#13 - 04/15/2019 10:44 PM - bmbouter

- Tags Pulp 2 added