

Docker Support - Story #2368

As a user, I want to provide synced images with multiple "/" in them.

10/24/2016 06:33 PM - tomckay@redhat.com

Status:	CLOSED - CURRENTRELEASE	Start date:	
Priority:	Normal	Due date:	
Assignee:	tomckay@redhat.com	% Done:	0%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:		Tags:	Pulp 2
Platform Release:	2.13.0	Sprint:	
Target Release - Docker:	2.4.0	Quarter:	
Groomed:	No		
Sprint Candidate:	Yes		
Description			
Note that docker/distribution and openshift/registry both now support more than two parts to images. In katello I would like to sync "openshift3/node" and then provide it at "examplecorp/myproduct/openshift3/node" (or any other name).			

Associated revisions

Revision c9cce34b - 03/29/2017 04:58 PM - thomasmckay

provide synced images with multiple "/" in them.

<https://pulp.plan.io/issues/2368>

History

#1 - 10/25/2016 04:21 PM - semyers

- Tracker changed from Issue to Story

- % Done set to 0

Do you know of any documentation from docker (dockermentation) about this to help with the implementation?

#2 - 10/25/2016 04:49 PM - amacdona@redhat.com

Which Pulp field(s) needs to support multiple "/"s? I can't tell if you mean the `repo_id`, the `upstream-name` or something else.

#3 - 10/25/2016 05:01 PM - semyers

Austin identified this as the limiting factor here:

https://github.com/pulp/pulp_docker/blob/0381e9ecd0a46c05f83bb20d4e584ad7fa81893e/plugins/pulp_docker/plugins/distributors/configuration.py#L317

This should allow for any number of 'slashed' components, while still validating each part:

```
all(re.match(r"^[a-z0-9-_.]*$", piece) for piece in docker_id.split('/'))
```

That said, it would (still) be helpful to see docker-docs about what may have change with regard to the validity/invalidity of repo names.

#4 - 10/25/2016 06:48 PM - tomckay@redhat.com

<https://github.com/docker/distribution/blob/master/docs/spec/api.md#overview>

1. A repository name is broken up into path components. A component of a repository name must be at least one lowercase, alpha-numeric characters, optionally separated by periods, dashes or underscores. More strictly, it must match the regular expression `[a-z0-9]+(?:[._-][a-z0-9]+)*`.
2. If a repository name has two or more path components, they must be separated by a forward slash ("/").
3. The total length of a repository name, including slashes, must be less than 256 characters.

#5 - 10/25/2016 11:39 PM - mhrivnak

This may also requires changes to crane. Its v2 endpoints may also assume only one slash.

I'm surprised to see this ability, and I wonder if it gets much use in the wild, but sure enough it does work!

```
[mhrivnak@dhcp129-40 devel]$ sudo docker tag centos:7 foo/bar/baz
[mhrivnak@dhcp129-40 devel]$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/centos	7	980e0e4c79ec	7 weeks ago	196.7 MB
foo/bar/baz	latest	980e0e4c79ec	7 weeks ago	196.7 MB

#6 - 11/08/2016 10:47 PM - mhrivnak

- Sprint Candidate changed from No to Yes

#8 - 03/06/2017 11:26 PM - semyers

Given the rules in comment 4, this validation is insufficient:

semyers wrote:

```
all(re.match(r"^[a-z0-9-]*$", piece) for piece in docker_id.split('/'))
```

I put together this little script along with the regex provided in that comment to check "good" and "bad" docker repo names are correctly classified:

```
import re

good = (
    'this-is-good',
    'this/is/good',
    'this-is/good',
)

bad = (
    'bad' * 256, # 'bad' repeated 256 times, 768 chars total
    '.bad',
    '_bad',
    '-bad',
)

component_re = re.compile('[a-z0-9]+(?:[._-][a-z0-9]+)*')

def validate(name):
    # good names return True, bad names return False
    if len(name) >= 256:
        return False

    for piece in name.split('/'):
        if not re.match(component_re, piece):
            return False
```

```
return True

# The one-liner version:
# return len(name) < 256 and all(re.match(component_re, piece) for piece in name.split('/'))

assert all(map(validate, good))
assert not any(map(validate, bad))
```

The provided regex along with the body of the validate function, appears to correctly validate according to those rules. Add more strings to the "good" and "bad" tuples as desired to prove the validation to your satisfaction.

#9 - 03/31/2017 04:10 PM - semyers

- Status changed from NEW to MODIFIED
- Assignee set to tomckay@redhat.com

This fix was POSTed by Thomas Mckay, in PR https://github.com/pulp/pulp_docker/pull/189, which was then merged to docker master in https://github.com/pulp/pulp_docker/commit/4276534ca77e564a74f5a97374df629dfed61094

#10 - 04/12/2017 04:08 PM - pcreech

- Platform Release set to 2.13.0
- Target Release - Docker set to 2.4.0

#11 - 04/17/2017 03:17 PM - pcreech

- Status changed from MODIFIED to 5

#12 - 04/24/2017 07:39 PM - Ichimonji10

Fully verified against Pulp 2.13 beta on F24. Partially verified against F25 and RHEL 7.

To verify, I created, synced and published three repositories. The three repositories had their repo_registry_id options set to one, two and three-segment values. Crane successfully presented information about the three repositories. On F24, I also successfully executed docker pull localhost:5000/{repo_registry_id} for all three repositories.

#14 - 04/27/2017 03:41 PM - pcreech

- Status changed from 5 to CLOSED - CURRENTRELEASE

#15 - 04/15/2019 10:23 PM - bmbouter

- Tags Pulp 2 added