Pulp - Task #2895

Task # 2868 (CLOSED - CURRENTRELEASE): Platform support for publishing.

As a user, I can have content served to me by Pulp

07/12/2017 06:28 PM - jortel@redhat.com

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	jortel@redhat.com	**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:	3.0.0		
Platform Release:		**Tags:**	
Groomed:	Yes	**Sprint:**	Sprint 25
Sprint Candidate:	Yes	**Quarter:**	

Description

Add content handling urls to serve conent. Some can be cannibalized from pulp2. Mainly parts of the ContentView. [0]

The handling URLs should be able to:

- Based on incoming request URL, determine the distribution
- Match the scheme against (http and https fields). 404 when scheme not enabled.
- Query the PublishedArtifact and PublishedMetadata tables (in that order) by URL path component to get the artifact or the metadata.
- respond with contents of the corresponding file

For this task the request and response should be handled with the normal WSGI application. There is a task to do enable this with apache later [1].

The content urls can be broken out separately later with magic in the settings.py and/or separate .wsgi files.

[0] https://github.com/pulp/pulp/tree/91a1e28c9e7d3dee418d5c7680dbf25c3e7adc63/server/pulp/server/content/web/views.py

[1] https://pulp.plan.io/issues/2914

History

#1 - 07/12/2017 06:30 PM - jortel@redhat.com

- Tags Pulp 3, Pulp 3 Plugin Writer Alpha added

What is involved with getting the test web server to route content URLs to the content app?

#2 - 07/12/2017 07:42 PM - mhrivnak

jortel@redhat.com wrote:

> What is involved with getting the test web server to route content URLs to the content app?

Assuming it's a separate django app, which I think it appropriate, we would just need to add it to INSTALLED_APPS in settings.py. It would be responsible for having its own urls.py and setting up its own routing there.

#3 - 07/12/2017 07:50 PM - bmbouter

Isn't this more than a django app installed to the pulp API's WSGI app, but it actually is it's own, distinct WSGI app? The content serving in pulp2 was its own WSGI app to not mix the content requests with the pulp API requests.

If that's the case it would get it's own wsgi entry point, and have its own urls.py and very few handlers. It would be called with runserver by specifying that settings.py application

#4 - 07/12/2017 07:53 PM - mhrivnak

I think it would be a good idea to add more detail to this task about exactly what functionality the app needs to have. I know the whole picture is spelled out in the other issue, but it's good to think through and write down what exactly this individual piece needs to do. Quick first thoughts:

- Based on incoming request URL, determine what publication is being accessed
- Using the publication, further determine which artifact or metadata file is being accessed
- respond with contents of the corresponding file
- or send through the streamer workflow (is this a refactor, rewrite, or mostly-stay-the-same situation?)
- somehow respond to requests for "listing" files (definitely not required for plugin API alpha)

#5 - 07/12/2017 07:57 PM - mhrivnak

bmbouter wrote:

> Isn't this more than a django app installed to the pulp API's WSGI app, but it actually is it's own, distinct WSGI app? The content serving in pulp2 was its own WSGI app to not mix the content requests with the pulp API requests.
>
> If that's the case it would get it's own wsgi entry point, and have its own urls.py and very few handlers. It would be called with runserver by specifying that settings.py application

Yes that's all correct. But the question was about using the test (dev) server, and for that I'm not sure how much it matters. We could require the developer to run two different dev servers: one for the rest api, and one for content. Or we could try to do it all in one process. I'm sure there are pros and cons of each, but neither seems completely objectionable just yet.

#6 - 07/12/2017 07:59 PM - bmbouter

mhrivnak that list looks great.

Also for the "respond with the contents of the file" that should actually stream the file using Django which allows for a streaming response type. I think we should file an add-on ticket that introduces a setting which would allow the view to return a mod xsendheader instead of streaming the actual bytes from the Django request handler. If enabled it responds with that xSendFile header set and an empty body so that Apache can stream it efficiently. I can file that if you want, just let me know.

We could do it in the same processes that is true. It certainly keeps it simple. The downside is that content data and api requests are mixed, but there is a nice upside of reduced memory usage and fewer qpid+database connections. The more I think about it the more I like it since apache will be serving production instances anyway.

#7 - 07/12/2017 09:36 PM - jortel@redhat.com

bmbouter wrote:

> mhrivnak that list looks great.
>
> Also for the "respond with the contents of the file" that should actually stream the file using Django which allows for a streaming response type. I think we should file an add-on ticket that introduces a setting which would allow the view to return a mod xsendheader instead of streaming the actual bytes from the Django request handler. If enabled it responds with that xSendFile header set and an empty body so that Apache can stream it efficiently. I can file that if you want, just let me know.
>
> We could do it in the same processes that is true. It certainly keeps it simple. The downside is that content data and api requests are mixed, but there is a nice upside of reduced memory usage and fewer qpid+database connections. The more I think about it the more I like it since apache will be serving production instances anyway.

In pulp2 the content app is a separate WSGI app[1]. One advantage to keep it a separate app is that it can be deployed on a separate server and configured with different resource allocations in Apache.

Is the new xsend setting something as simple as just adding:

```
XSEND = (true/false)
```

to settings.py.

[1] https://github.com/pulp/pulp/tree/master/server/pulp/server/content/web

#8 - 07/12/2017 09:40 PM - jortel@redhat.com

mhrivnak wrote:

> I think it would be a good idea to add more detail to this task about exactly what functionality the app needs to have. I know the whole picture is spelled out in the other issue, but it's good to think through and write down what exactly this individual piece needs to do. Quick first thoughts:
>
> - Based on incoming request URL, determine what publication is being accessed
> - Using the publication, further determine which artifact or metadata file is being accessed
> - respond with contents of the corresponding file
> - or send through the streamer workflow (is this a refactor, rewrite, or mostly-stay-the-same situation?)

There isn't much logic in the pulp2 ContentView but most of it can be generally reused.

- somehow respond to requests for "listing" files (definitely not required for plugin API alpha)

This list looks fine.

#9 - 07/13/2017 09:29 PM - jortel@redhat.com

After talking through this, it seems like the content app can be just a View in the platform/app. We can deploy separately with magic in the settings.py and/or separate .wsgi files.

#10 - 07/13/2017 09:49 PM - bmbouter

Yes I think that is fine for today and may be fine forever even. Production installs shouldn't use Django to stream bits so they shouldn't experience issues from content requests streaming large data back which would tie up the WSGI processes and could queue/block API requests.

#11 - 07/14/2017 02:34 AM - amacdona@redhat.com

- Subject changed from Port content WSGI applicaiton from pulp2 to pulp3. to Port content WSGI application from pulp2 to pulp3.

#12 - 07/17/2017 04:08 PM - bizhang

bizhang will groom

#13 - 07/17/2017 04:49 PM - bmbouter

Note this is to be split into two tasks: https://pulp.plan.io/issues/2868#note-22

#14 - 07/17/2017 05:54 PM - jortel@redhat.com

- Description updated

#15 - 07/17/2017 06:14 PM - jortel@redhat.com

- Subject changed from Port content WSGI application from pulp2 to pulp3. to Add Content WSGI application to server content.

- Description updated

#16 - 07/17/2017 08:35 PM - bizhang

- Description updated

#17 - 07/17/2017 08:50 PM - bizhang

- Groomed changed from No to Yes

- Sprint Candidate changed from No to Yes

#18 - 07/17/2017 09:14 PM - bmbouter

- Subject changed from Add Content WSGI application to server content. to As a user, I can have content served to me by Pulp

- Description updated

I retitled and edited it to reflect the comments where we decided to use the existing WSGI application instead of adding a new one.

#19 - 07/20/2017 04:21 PM - mhrivnak

- Priority changed from Normal to High

#20 - 07/20/2017 04:49 PM - mhrivnak

- Sprint/Milestone set to 42

#21 - 07/24/2017 05:42 PM - jortel@redhat.com

- Status changed from NEW to ASSIGNED

- Assignee set to jortel@redhat.com

#22 - 07/25/2017 11:08 PM - jortel@redhat.com

- Status changed from ASSIGNED to POST

https://github.com/pulp/pulp/pull/3100

#23 - 08/14/2017 04:36 PM - mhrivnak

- Sprint/Milestone changed from 42 to 43

#24 - 09/05/2017 06:04 PM - jortel@redhat.com

- Sprint/Milestone changed from 43 to 44

#25 - 09/12/2017 10:34 PM - jortel@redhat.com

- Status changed from POST to MODIFIED

#26 - 12/19/2017 04:23 PM - bmbouter

- Tags deleted (Pulp 3 Plugin Writer Alpha)

Cleaning up Redmine tags

#28 - 03/09/2018 12:23 AM - bmbouter

- Sprint set to Sprint 25

#29 - 03/09/2018 12:23 AM - bmbouter

- Sprint/Milestone deleted (44)

#30 - 04/25/2019 06:46 PM - daviddavis

- Sprint/Milestone set to 3.0.0

#31 - 04/26/2019 10:38 PM - bmbouter

- Tags deleted (Pulp 3)

#32 - 12/13/2019 06:16 PM - bmbouter

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE