# Pulp - Task #7178

## Recommended installation layout

07/21/2020 02:05 PM - ekohl

| | | | |
|---|---|---|---|
| **Status:** | POST | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0:00 hour |
| **Sprint/Milestone:** | | | |
| **Platform Release:** | | **Tags:** | Dev Environment, Documentation, Katello, SELinux |
| **Groomed:** | No | **Sprint:** | |
| **Sprint Candidate:** | No | **Quarter:** | |

**Description**

While trying to address https://projects.theforeman.org/issues/30423 I was looking at the recommended layout. There are a few things I'd suggest to do different.

https://docs.pulpproject.org/settings.html#media-root suggests /var/lib/pulp with mode 750 and SELinux context var_lib_t. I disagree with all of these.

The directory should be its own directory (so /var/lib/pulp/media) because it means all files within it are owned by Pulpcore. This means you can use unreferenced_files from django-extensions without adjustment.

If the directory is /var/lib/pulp then mode 750 means Apache can't read files from the directory. This conflicts with Pulp 2 and serving assets directly via Apache. This is not a concern when /var/lib/pulp/media is used.

The SELinux context should be pulpcore_var_lib_t so Apache and other services are denied access to media files. Again, this can only be done when it's in a subdirectory.

Looking at https://github.com/pulp/pulpcore/blob/09d05da6f38c74e3574d3d256f890b23d76cb3d0/pulpcore/app/settings.py#L36-L43 there are various settings that are derived from MEDIA_ROOT, which is IMHO incorrect.
https://docs.djangoproject.com/en/2.2/ref/settings/#media-root states

> Absolute filesystem path to the directory that will hold user-uploaded files

STATIC_ROOT should be part of MEDIA_ROOT. A common pattern is to introduce a setting (like ROOT_DIR or similar) and derive all locations based on that. Then in production mode you can set ROOT_DIR to /var/lib/pulp and automatically get all recommended directories correct.

Then there is the SELinux policy. As mentioned previously, it doesn't set the SELinux type for all the MEDIA_ROOT, but only the artifact directory. This is IMHO incorrect.

It also sets the SELinux type for assets (which is the default directory for STATIC_ROOT) to pulpcore_var_lib_t but Apache isn't allowed to serve that. However, it is the most efficient way to serve these files. Perhaps this could use the type httpd_sys_content_t but I don't know if that's appropriate. A more experience SELinux dev should weigh in on this.

Lastly there's the SELinux types for bin/gunicorn and bin/rq. In the Katello RPM packaging these live in /usr/bin and don't get a label. I don't think it's appropriate to label those files pulpcore_exec_t, but the result is that the Pulp services run unconfined in the Katello deployment. A common pattern is to use /usr/libexec to create wrappers. The systemd services could then call these wrappers with the correct context.

**Subtasks:**

Task # 7482: pulp_installer change(s) for Recommended installation layout **NEW**

### History

**#1 - 07/21/2020 04:51 PM - pulpbot**

*- Status changed from NEW to POST*

PR: https://github.com/pulp/pulpcore/pull/799

#### #2 - 07/21/2020 05:49 PM - rchan

*- Project changed from File Support to Pulp*

moving from File project to Pulp.

#### #3 - 07/21/2020 10:03 PM - mdepaulo@redhat.com

ekohl wrote:

> If the directory is /var/lib/pulp then mode 750 means Apache can't read files from the directory. This conflicts with Pulp 2 and serving assets directly via Apache. This is not a concern when /var/lib/pulp/media is used.

This is incorrect.

1. We want to use group access, not world access, for security.
2. We  add the "apache" user to the "pulp" group, and run chgrp, and run basically "chmod 770" (we add group access) for pulp 2 upgrades. So that apache does have rw access: https://pulp.plan.io/issues/5553
   https://github.com/pulp/pulp_installer/blob/master/roles/pulp_common/tasks/install.yml#L85
3. The problem for new installs, where the mode is intentionally 750, is a regression bug in pulp_installer. The group on the dir is "users" rather than "pulp". This should be addressed via the bug report I filed recently: https://pulp.plan.io/issues/7173

   The SELinux context should be pulpcore_var_lib_t so Apache and other services are denied access to media files. Again, this can only be done when it's in a subdirectory.

I don't follow. I thought apache and pulp were supposed to both have access to certain directories, per the aforementioned issue: https://pulp.plan.io/issues/5553

> Lastly there's the SELinux types for bin/gunicorn and bin/rq. In the Katello RPM packaging these live in /usr/bin and don't get a label. I don't think it's appropriate to label those files pulpcore_exec_t, but the result is that the Pulp services run unconfined in the Katello deployment. A common pattern is to use /usr/libexec to create wrappers. The systemd services could then call these wrappers with the correct context.

This is a really good idea IMO.

#### #4 - 07/22/2020 11:32 AM - ekohl

mdepaulo@redhat.com wrote:

> ekohl wrote:
>
> > If the directory is /var/lib/pulp then mode 750 means Apache can't read files from the directory. This conflicts with Pulp 2 and serving assets directly via Apache. This is not a concern when /var/lib/pulp/media is used.
>
> This is incorrect.
>
> 1. We want to use group access, not world access, for security.

I'd argue this is less secure. By adding Apache to the Pulp group, they can read all file as opposed to only the static files. IIRC in the Katello Pulp 2 deployment this was never done either so it was even less secure.

In the layout I proposed, only a few directories are world readable and those are served without authentication by Apache. By having the MEDIA_ROOT in a subdirectory, you can actually guarantee it's private with simple directory permissions. Your proposal solely relies on SELinux to prevent Apache from reading the media files directly.

In https://github.com/theforeman/puppet-pulpcore/pull/115 I added documentation for the layout implementation by printing the directory structure and which variables point to them.

```
/
├── [drwxr-xr-x root     root    ]  etc
│   └── [drwxr-xr-x root     pulp    ]  pulp ($config_dir)
│       └── [-rw-r----- root     pulp    ]  settings.py
└── [drwxr-xr-x root     root    ]  var
    └── [drwxr-xr-x root     root    ]  lib
        └── [drwxrwxr-x pulp     pulp    ]  pulp ($user_home)
            ├── [drwxrwxr-x pulp     pulp    ]  docroot ($apache_docroot)
            ├── [drwxrwx--- pulp     pulp    ]  media ($media_root)
            ├── [drwxrwxr-x pulp     pulp    ]  static ($static_root)
            ├── [drwxrwx--- pulp     pulp    ]  tmp ($cache_dir)
            └── [drwxrwx--- pulp     pulp    ]  upload ($chunked_upload_dir)
```

> 1. We  add the "apache" user to the "pulp" group, and run chgrp, and run basically "chmod 770" (we add group access) for pulp 2 upgrades. So that apache does have rw access: https://pulp.plan.io/issues/5553
> https://github.com/pulp/pulp_installer/blob/master/roles/pulp_common/tasks/install.yml#L85

In the Foreman/Katello installer we never never done that. Reading the code, I think it would be a bad idea for a pure Pulp 3 deployment. In the Pulp 2 deployment that Katello uses, the application runs as the Apache user. Thus /var/lib/pulp does need to be writable by Apache. It would be simpler to make /var/lib/pulp owned by apache:pulp and mode 0755 rather than adding the group **if** it's a mixed Pulp 2/3 deployment.

> 1. The problem for new installs, where the mode is intentionally 750, is a regression bug in pulp_installer. The group on the dir is "users" rather than "pulp". This should be addressed via the bug report I filed recently: https://pulp.plan.io/issues/7173
>
>    The SELinux context should be pulpcore_var_lib_t so Apache and other services are denied access to media files. Again, this can only be done when it's in a subdirectory.

> I don't follow. I thought apache and pulp were supposed to both have access to certain directories, per the aforementioned issue: https://pulp.plan.io/issues/5553

My understanding is that Apache only accesses static files, not media files. For the latter there's the content application.

#### #5 - 07/27/2020 07:29 PM - ekohl

On the matter of /var/lib/pulp/static vs /varlib/pulp/assets. After playing more I've learned that due to the use of the whitenoise application (which reads the static files @ application startup), you have to make the media files available to the API application. If they're in static (as opposed to assets) and the Pulp application runs confined, it's not allowed to read the files and you get a denial. Since it's already serving those, it doesn't make a lot of sense to then directly serve them via Apache - it just complicates matters and it's not like they're served a lot anyway. This means the current practice of using assets is probably the best one.

#### #6 - 07/28/2020 04:40 PM - fao89

*- Tracker changed from Issue to Task*

*- % Done set to 0*

*- Severity deleted (2. Medium)*

*- Triaged deleted (No)*