

## Pulp - Task #3065

### Replace the vagrant user with a user named 'pulp'

10/11/2017 06:56 PM - daviddavis

<b>Status:</b> CLOSED - WONTFIX	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0:00 hour
<b>Sprint/Milestone:</b> 3.0.0	<b>Tags:</b>
<b>Platform Release:</b>	<b>Sprint:</b>
<b>Groomed:</b> No	<b>Quarter:</b>
<b>Sprint Candidate:</b> No	

**Description**

We currently use a vagrant user that owns files (e.g. /var/lib/pulp) and run processes (e.g. pulp\_workers), etc. We'd like to rename this vagrant user to pulp to be consistent with ansible installs.

**Related issues:**

Related to Pulp - Issue #3050: get rid of "apache" user in dev environment **CLOSED - CURRENTRELEASE**

#### History

**#1 - 10/11/2017 06:56 PM - daviddavis**

- Related to Issue #3050: get rid of "apache" user in dev environment added

**#2 - 10/16/2017 09:07 PM - amacdona@redhat.com**

When I type "vagrant ssh", will that log into the vm as the "pulp" user?

I think that we should **not** hardcode the username, but rather continue to do this:

<https://github.com/pulp/devel/blob/3.0-dev/ansible/roles/pulp-user/tasks/main.yml#L18>

**#3 - 10/16/2017 09:27 PM - daviddavis**

Yea, the plan is that when you run "vagrant ssh", you'll ssh in as a user named pulp.

We set pulp\_user here:

<https://github.com/pulp/devel/blob/d44e53186495a19773d043cc80748e3c88df7855/ansible/pulp-from-source.yml#L10>

I think that will be the place we'll need to change to "pulp".

**#4 - 10/17/2017 04:36 PM - daviddavis**

- Description updated

**#5 - 10/17/2017 04:37 PM - amacdona@redhat.com**

- Tracker changed from Issue to Task

- % Done set to 0

**#6 - 10/18/2017 10:41 PM - Ichimonji10**

I love the idea of not hardcoding a username into our installers, but rather letting Pulp be run as any arbitrary user. It's certainly a more technically

correct way of approaching the problem of "how do I run a daemon?" And given that Pulp 3 is designed so that it can be installed into an arbitrary virtualenv, it seems like an especially important consideration to keep in mind. In practical terms, this is easy to do. We're using Ansible to install Pulp 3, and if our Ansible code is well written, it should be able to handle this without any issues at all.

This design decision highlights an important design point that, as far as I can tell, still hasn't been resolved: where is the boundary between Pulp and the outside world? In other words, at what point is data handed to a process owned by the "pulp" or "vagrant" user? This may not be the best place to bring up this discussion. Feel free to point me elsewhere. Rabbit trail alert.

I've seen one proposed solution so far. The solution is for a webserver (apache/nginx/lighttpd/etc) to run as the Pulp 3 user ("pulp", "vagrant", etc), where a (WSGI) process runs inside the webserver's memory space, and where that process is responsible for receiving requests and placing them in the celery queue for other Pulp processes to pick up.

I hope we can avoid this solution. It makes me go cock-eyed when I realize that the system-wide webserver that runs on standard ports like 80/443 won't be running as a standard user. There's a couple reasons for this:

- This is a highly unusual set-up. If I was hunting for a piece of file distribution software, and I was reading through Pulp's docs and thinking about whether to use it, that piece of information would make me immediately say "that's a ridiculous design." I'd move on to the next candidate.
- There may be file ownership issues that might now need to be addressed. For example, on my system, nginx owns files like /etc/nginx/..., /etc/logrotate.d/nginx, /usr/bin/nginx, /var/lib/nginx/proxy/, /var/log/nginx/, etc. Do I now need to change the ownership of all those files/dirs/etc? Once I do that, upgrades become a nightmare. A package manager typically knows who owns those files. Will the package manager leave ownership alone when upgrading the system?
- This may break security policies, like SELinux security policies.
- This design prevents the web server from easily being used for other applications. What if I want my web server to serve another web application too, or another static website? As a sysadmin, I will be wondering if installing Pulp 3 will break other applications, due to the strange-ness of this set-up.

A much more typical set-up could look like this:

client → reverse proxy → HTTP server → celery queue

In this case, the boundary between the outside world and Pulp 3 processes lies between the reverse proxy and the HTTP server. The reverse proxy can be any arbitrary application that's sufficiently capable, such as Apache, Nginx or LightTPD. That application can be installed as normal and run as the user that users expect, and it would require only a little bit of configuration, such as this:

```
# pulp3.example.com
server {
    listen 80;
    server_name pulp3.example.com;
    return 301 https://$host$request_uri;
}
server {
    listen 443 ssl;
    server_name pulp3.example.com;
    ssl_certificate ssl/pulp3.example.com.chained.crt;
    ssl_certificate_key ssl/pulp3.example.com.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # SSLv3 is insecure
    location / {proxy_pass http://127.0.0.1:8000;}
}
```

(This is adapted from a [real nginx config file](#))

For initial development purposes, the HTTP server could be the one built in to Django. And as development goes forward, y'all could switch to something like [Gunicorn](#). (Gunicorn integrates well with Django, as it's a WSGI HTTP server.)

**#7 - 05/16/2018 04:43 PM - bmbouter**

Recently I've seen that all Vagrant environments use the user vagrant, so I don't think having everything run in the devel environment as user 'vagrant' as a bad thing. I'm +1 to closing this as WONTFIX.

**#8 - 05/16/2018 04:49 PM - daviddavis**

- Status changed from NEW to CLOSED - WONTFIX

[bmbouter](#) agreed. I'll close this out and if any objects, they can reopen.

**#9 - 04/25/2019 06:46 PM - daviddavis**

- Sprint/Milestone set to 3.0.0

**#10 - 04/26/2019 10:37 PM - bmbouter**

- Tags deleted (Pulp 3)