# Pulp - Story #7791

## As a user I can re-upload artifacts if the file has gone missing or corrupted

11/04/2020 06:33 PM - ipanova@redhat.com

| | | | |
|---|---|---|---|
| **Status:** | NEW | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0:00 hour |
| **Sprint/Milestone:** | | | |
| **Platform Release:** | | **Tags:** | |
| **Groomed:** | No | **Sprint:** | |
| **Sprint Candidate:** | No | **Quarter:** | |

**Description**

If an artifact has gone missing or corrupted(bit rot) there is no way to re-upload it back to the filesystem

1. upload an artifact
2. rm /var/lib/pulp/artifacts/<some_artifact> or corrupt it
3. upload same artifact
4. 400 error with { "non_field_errors": [ "sha384 checksum must be unique." ] }

While we have /repair/ endpoint it will not work for the operations where artifact has no  remoteartifact.

# Problem statements:

## 1. If a file is missing it is impossible to upload a new one

- when saving artifact add try/except, look for existing one
- verify whether storage_path is an existing location if not update it with the newly uploaded bits
- Issue 400 due to duplicated artifact, but in addition **return the href** of the existing artifact.

## 2. If a file is corrupted it is impossible to re-upload and replace it with a valid one

- **option1** Running repair can find corrupted files. It should remove the corrupted file to get back to the case outlined in 1. .
  - Repair can be run against specific repo version, potentially can extend the functionality to repair a specific artifact/content
- **option2**  We could recalculate the checksum on all upload attempts.
  - Might be a lot of overhead for a rare failure
- **option3** Introduce a flag which will be specified at upload time E.g. --repair, or --force, or --validate-checksum. It will replace broken bits if checksum of the newly uploaded file matches a checksum in the DB. The recalculation of a checksum will happen on_demand this way and not for every upload attempt.

**Related issues:**

| | |
|---|---|
| Related to Pulp - Story #7114: Improve Artifact upload experience | **NEW** |
| Copied from Container Support - Story #7790: As a user I can re-upload artifa... | **NEW** |

**History**

**#1 - 11/04/2020 06:33 PM - ipanova@redhat.com**

*- Copied from Story #7790: As a user I can re-upload artifacts if the file has gone missing or corrupted added*

**#2 - 12/09/2020 06:50 PM - ipanova@redhat.com**

*- Description updated*

**#3 - 12/16/2020 02:10 PM - ttereshc**

1. If a file is missing it is impossible to upload a new one when saving artifact add try/except, look for existing one verify whether storage_path is an existing location if not update it with the newly uploaded bits

I would explicitly mention here that the checksums should match, the one in the DB and the checksum of newly uploaded file (checksum in the DB should not be updated, just used for comparison)

1. If a file is corrupted it is impossible to re-upload and replace it with a valid one

how about option3 (a more explicit version of option2): introduce a flag which will be specified at upload time E.g. --repair, or --force, or --validate-checksum.
It will replace broken bits if checksum of the newly uploaded file matches a checksum in the DB.
The recalculation of a checksum will happen on_demand this way and not for every upload attempt.

#### #4 - 12/16/2020 03:22 PM - ipanova@redhat.com

ttereshc wrote:

> 1. If a file is missing it is impossible to upload a new one when saving artifact add try/except, look for existing one verify whether storage_path is an existing location if not update it with the newly uploaded bits
>
> I would explicitly mention here that the checksums should match, the one in the DB and the checksum of newly uploaded file (checksum in the DB should not be updated, just used for comparison)

I was imagining the artifat._init_and_validate would calculate all the checksums of the upload in course and when saving it in case of existing artifact we would handle the IntegrityError

> 1. If a file is corrupted it is impossible to re-upload and replace it with a valid one
>
> how about option3 (a more explicit version of option2): introduce a flag which will be specified at upload time E.g. --repair, or --force, or --validate-checksum.
> It will replace broken bits if checksum of the newly uploaded file matches a checksum in the DB.
> The recalculation of a checksum will happen on_demand this way and not for every upload attempt.

This is a good compromise for the user experience improvement.

#### #5 - 12/16/2020 03:26 PM - ttereshc

ipanova@redhat.com wrote:

> ttereshc wrote:
>
> > 1. If a file is missing it is impossible to upload a new one when saving artifact add try/except, look for existing one verify whether storage_path is an existing location if not update it with the newly uploaded bits
> >
> > I would explicitly mention here that the checksums should match, the one in the DB and the checksum of newly uploaded file (checksum in the DB should not be updated, just used for comparison)
>
> I was imagining the artifat._init_and_validate would calculate all the checksums of the upload in course and when saving it in case of existing artifact we would handle the IntegrityError

Right, it's a part of the try/except you mention, ok. Disregard my comment then.

#### #6 - 12/16/2020 04:47 PM - ttereshc

After some offline discussion, it turned out that this ticket implies that the upload of existing artifact (which is good and NOT corrupted or missing) will also return the existing artifact and won't fail.
This directly relates to #7114.
If it's done this way, we ought to be consistent across all resources. At least, content upload needs to be adjusted as well.

We also need to be sure that it's only affecting upload workflows and not the artifact creation during sync.

#### #7 - 12/16/2020 04:48 PM - ttereshc

*- Related to Story #7114: Improve Artifact upload experience added*

#### #8 - 12/23/2020 02:49 PM - daviddavis

ttereshc wrote:

> After some offline discussion, it turned out that this ticket implies that the upload of existing artifact (which is good and NOT corrupted or missing) will also return the existing artifact and won't fail.
> This directly relates to #7114.
> If it's done this way, we ought to be consistent across all resources. At least, content upload needs to be adjusted as well.
>
> We also need to be sure that it's only affecting upload workflows and not the artifact creation during sync.

#7114 asks the pulp href be returned if the artifact already exists. It does not ask that the end point should succeed and no longer fail.

Also, IMO this proposal would break semantic versioning. Changing the response when a user uploads the exact same artifact twice would be a backwards incompatible change.

#### #9 - 12/23/2020 04:55 PM - daviddavis

> how about option3 (a more explicit version of option2): introduce a flag which will be specified at upload time E.g. --repair, or --force, or --validate-checksum.

To me this makes the most sense. It's backwards compatible and is clear to the user what's happening. Fixing an artifact by calling the create endpoint without a special param seems a bit too strange/magical/clever IMO.

#### #10 - 01/04/2021 12:58 PM - ipanova@redhat.com

*- Description updated*

#### #11 - 01/04/2021 01:05 PM - ipanova@redhat.com

daviddavis wrote:

> > how about option3 (a more explicit version of option2): introduce a flag which will be specified at upload time E.g. --repair, or --force, or --validate-checksum.
>
> To me this makes the most sense. It's backwards compatible and is clear to the user what's happening. Fixing an artifact by calling the create endpoint without a special param seems a bit too strange/magical/clever IMO.

The caveat of this option is if a client is trying get content into pulp, like podman, there is no way to specify this option on the client side, since the pulp upload layer is in the middle of the process. I am not advocating to adjust the whole proposal based on this specific usecase, but we'd need to whether hardcode the 'force' flag in the plugin code during the upload or come up with something else to enable container plugin to deal with the corrupted content.

#### #12 - 01/04/2021 03:00 PM - daviddavis

ipanova@redhat.com wrote:

> The caveat of this option is if a client is trying get content into pulp, like podman, there is no way to specify this option on the client side, since the pulp upload layer is in the middle of the process. I am not advocating to adjust the whole proposal based on this specific usecase, but we'd need to whether hardcode the 'force' flag in the plugin code during the upload or come up with something else to enable container plugin to deal with the corrupted content.

That's a good point.

Just to call it out: another option is to bump the pulpcore major version. We have some other changes in our queue that would benefit from being able to make some backwards incompatible changes to the API (e.g. https://pulp.plan.io/issues/7762)