

Item 370

## **git\_comments:**

## **git\_commits:**

1. **summary:** ARROW-9334: [Dev][Archery] Push ancestor docker images  
**message:** ARROW-9334: [Dev][Archery] Push ancestor docker images Closes #7681 from kszucs/ARROW-9334 Authored-by: Krisztián Szűcs <szucs.krisztian@gmail.com> Signed-off-by: Krisztián Szűcs <szucs.krisztian@gmail.com>

## **github\_issues:**

## **github\_issues\_comments:**

## **github\_pulls:**

1. **title:** ARROW-9334: [Dev][Archery] Push ancestor docker images  
**body:**

## **github\_pulls\_comments:**

1. <https://issues.apache.org/jira/browse/ARROW-9334>

## **github\_pulls\_reviews:**

## **jira\_issues:**

1. **summary:** [Dev][Archery] Push ancestor docker images  
**description:** The "debian-c-glib" and "ubuntu-c-glib" docker-compose configurations fail with the following message: {code:java} CMake Error at /usr/share/cmake-3.13/Modules/FindPackageHandleStandardArgs.cmake:137 (message): Could NOT find utf8proc (missing: UTF8PROC\_LIB UTF8PROC\_INCLUDE\_DIR) Call Stack (most recent call first): /usr/share/cmake-3.13/Modules/FindPackageHandleStandardArgs.cmake:378 (\_FPHSA\_FAILURE\_MESSAGE) cmake\_modules/Findutf8proc.cmake:41 (find\_package\_handle\_standard\_args) cmake\_modules/ThirdpartyToolchain.cmake:159 (find\_package) cmake\_modules/ThirdpartyToolchain.cmake:2096 (resolve\_dependency) CMakeLists.txt:467 (include {code})

## **jira\_issues\_comments:**

1. cc [~kou] [~kszucs]
2. Rebuilding the image fixes the issue, so it seems the uploaded images are not up-to-date?
3. Ah, they are old because {{archery docker push ubuntu-ruby}} pushes only {{ubuntu-ruby}} image. [https://github.com/apache/arrow/runs/842945967?check\\_suite\\_focus=true#step:9:2](https://github.com/apache/arrow/runs/842945967?check_suite_focus=true#step:9:2) [~kszucs] Could you update Archery to push ancestor images of the specified image too? The following patch may work but I didn't run it: {noformat} diff --git a/dev/archery/archery/docker.py b/dev/archery/archery/docker.py index 23448e6da..65fac3a12 100644 --- a/dev/archery/archery/docker.py +++ b/dev/archery/archery/docker.py @@ -271,6 +271,8 @@ class DockerCompose(Command): .format(image)) raise RuntimeError(msg) from None else: + for ancestor in self.nodes[image]: + self.\_execute\_compose('push', ancestor) self.\_execute\_compose('push', image) def images(self): {noformat}
4. It seems that similarly, {{conda-python}} is not up-to-date (at least with PYTHON=3.6).
5. Issue resolved by pull request 7681 [<https://github.com/apache/arrow/pull/7681>]
6. Are you sure this is actually fixed?
7. I got the following: {code} \$ archery docker run --no-build debian-c-glib Pulling debian-cpp ... done Pulling debian-c-glib ... done WARNING: Some service image(s) must be built from source by running: docker-compose build debian-c-glib [...] CMake Error at /usr/share/cmake-3.13/Modules/FindPackageHandleStandardArgs.cmake:137 (message): Could NOT find utf8proc (missing: UTF8PROC\_LIB UTF8PROC\_INCLUDE\_DIR) Call Stack (most recent call first): /usr/share/cmake-3.13/Modules/FindPackageHandleStandardArgs.cmake:378 (\_FPHSA\_FAILURE\_MESSAGE) cmake\_modules/Findutf8proc.cmake:41

(find\_package\_handle\_standard\_args) cmake\_modules/ThirdpartyToolchain.cmake:159 (find\_package)  
cmake\_modules/ThirdpartyToolchain.cmake:2096 (resolve\_dependency) CMakeLists.txt:484 (include) --  
Configuring incomplete, errors occurred! {code}

8. By passing `{{--no-build}}` you prevent it from rebuilding the image. Try with `{{archery docker run debian-c-glib}}`
9. That's the whole point. The upstream image should be up-to-date now.
10. Well, the push has happened <https://github.com/apache/arrow/runs/856279754#step:9:20> Additionally `{{archery docker run --no-build debian-c-glib}}` works for me locally, so it may be caused by other docker/docker-compose cache issues. I'm trying to reproduce it locally.
11. We're running the ubuntu-ruby builds on the CI not the debian ones, so debian-c-glib and debian-ruby are not up to date, only the ubuntu images. Executing `{{archery docker run --no-build ubuntu-c-glib}}` works as expected.
12. Hmm, thank you. It's a bit confusing that we have those Debian images which are not exercised nor updated...
13. We're exercising them as nightlies, but not pushing them to the apache docker account. I'm creating an issue about pushing from the nightly builds with our own credentials.  
<https://issues.apache.org/jira/browse/ARROW-9435>