YARN Localization Support for Docker Image (YARN-3854)

Zhankun 2016/08

1. Introduction

YARN now supports Docker to launch containers through the LinuxContainerExecutor. Currently YARN will delegate the preparation of the Docker image to the Docker engine if the image is not present on the target host. But there are more issues to be considered in a YARN cluster when it comes to having the Docker images ready before launching Docker. We filed this design document for discussion of in-band control of Docker images in YARN (YARN-3854).

2. Problem Definition

As we all know, Docker will try to pull images from docker hub or private repository if no local image found. So currently, to ensure that LinuxContainerExecutor can launch container successfully, one must configure public network for Docker daemon, set up own private repo or prepare the image in advance by “docker load” manually/semi-automatically.

Each above choice needs more thinking in it, let’s check them one by one. Firstly, if a user enables public network, the docker pull speed will be a concern. The task will timeout if docker pull operation costs much because of sufficient large image or unstable networking. I guess unstable public network prevent us from using it in production environment.

Secondly, investments on own private repository in an internal network is a good choice and most big company will do this. This private repo is not only for YARN but also CI/CD integration with other existing system to use Docker. We agree with this solution if the user can afford to maintain a private repo. One issue for implicit docker pull in launch phase is that YARN cannot know what’s happening if the pull process is stuck for unknown reason. An explicit docker pull before launch phase could be better.

Thirdly, a user can manually load images for cluster in advance or develop own YARN application to do this step. But doing this load stuff manually is obviously unacceptable. And although a YARN application which spawn a separate task can achieve this, it needs the submitting user has the privilege of docker group. This also has potential security issue. Another potential way to do this is utilizing HDFS distributed cache as as the vehicle for sharing Docker images and then “docker load” it into Docker local repository. This needs the application “docker save” the image to tarball and upload it to HDFS. Then YARN would do the docker load for the application. The biggest issue with “HDFS + docker load” is that we cannot ensure the application provided tarball is trusted to avoid the applications override each other’s image in a naming conflict accident.

To sum up, although there are various approaches that could be used here with different trade-offs/issues : image archives on HDFS + docker load , docker pull during the localization phase or (automatic) docker pull during the run/launch phase, this proposal prefer to enable the docker image localization by “docker pull” during localization phase. We can split it into three questions and provide possible solutions for each question in next section:

1. What’s the interface for application that let YARN know where it needs to pull a Docker image during localization from?
2. Where to execute the “docker pull”?
3. What’s the implications? For instance, clean up images.

3. Possible solutions

3.1 Docker image localization interface

Option 1. Add a “DOCKERIMAGE” type in LocalResourceType and specify image name in “[registry/image:tag]” format in “URL”
This interface is straightforward and try to uniform Docker image into a new kind of LocalResource. The application sets “LocalResourceType.DOCKERIMAGE” to indicate it needs a Docker image resource localization and specify image name in “URL” (without protocal schema). Then YARN can just check the type to know it needs to docker pull the image specified.

3.2 Where to execute the “docker pull”?

Option 1. Extend the “FSDownload.java” to execute docker pull
Currently no additional information is needed for docker pull to work when FSDownload is spawned. It’s natural to implement this docker pull here.

3.3 Integration implications of YARN “docker pull”

Image cleanup

The visibility of resource still apply for Docker image. We need to keep current resource life-cycles policy working for Docker image. New deletion operation in “DeletionService” is needed for Docker image deletion.

Localization timeout
It seems YARN-2175 is for this timeout. And we won’t solve this in this JIRA because it can be detected by application and the private repo is assumed to be fast and stable.

Input Docker credentials and store option
We may need to handle interactive user/password prompt when docker pull against secure private repository if no credentials found in Docker client config file. Docke will store your credentials in “~/.docker/config.json” by default (currently no expiration).
If specific keychain or external store is specified in config.json, Docker would use it to get the credentials and pull the image.

YARN-5428 means to enables us to specify the config file which potentially contains user credentials. This can make docker pull operation automatic if the administrator stores local repository credentials in it.

Credentials store refer to:
https://docs.docker.com/engine/reference/commandline/login/#/credentials-store

Unsecure docker repository
Docker assume all repository is secure by default. If an insecure registry is not marked as insecure thru docker daemon option, “docker pull” will result in an error message prompting. We assume the repository should be configured correctly by administrator because YARN won’t fix it but just raise this failure.

Details refer to
https://docs.docker.com/engine/reference/commandline/dockerd/#/insecure-registries

4. Opinion on best solution

We don’t describe each solutions in detail but list the pros and cons below to compare different solutions:

Solution	Pros	Cons
Image archives on HDFS + docker load	● No dependencies on Docker daemon and no need to maintain private repo	● Potential security issue caused by image name/tag conflicts when docker load ● The application/admin

		should upload the image archives to HDFS and manage it
Docker pull during the localization phase	● Fast Docker image download and delegate the management to public/private registry ● Avoid hung implicitly in pull operation in launch phase	● Managing “docker login” related stuffs may require YARN to evolve with Docker private repository. e.g. if current authentication in Docker changed, YARN needs changes too.
Docker pull during the launch phase	● No changes needed in localization	● Docker pull operation is a black box to YARN.

We think the “docker pull” during the localization phase is a good balance between HDFS as a rough registry and Docker registry as a black box. It has benefits from Docker registry but still have control on it in certain extent.

5. Details on how the solution should be implemented

This proposal enables the interface for YARN application to specify a “DOCKERIMAGE” resource thru type and URL.
We prefer that Docker repository credentials are ready before YARN executes docker pull. This make our design simple and protect YARN from interaction with “docker pull” prompt. Then the implementation parts involves below steps:

1. Add support for docker pull operation. Aka. DockerPullCommand
2. Extend LocalResourceType and utilize DockerPullCommand in FSDownload. After image downloaded, leave a hint file in corresponding resource local directory to record the image name.
3. Add support for docker image deletion operation. Aka. DockerRmiCommand
4. Extend deletion service for removing unused Docke images

6. Future work

TBD