Dockerize Hadoop

Proposed design by Marton Elek
Context/Background

This slides are some kind of design documents for HADOOP-14898.

An example implementation is available from here: https://github.com/elek/hadoop/tree/docker-2.8.0

Background: I have an own set of dockerized hadoop/spark images under the flokkr project (https://github.com/flokkr/flokkr) with more advanced capabilities. I have experiences to run them with kubernetes/docker-swarm/docker+nomad + consul based configuration management.
Use cases

- Start a cluster from the latest development build (See the HDFS-7240 for an example):
  - git checkout HDFS-7240
  - mvn clean install -Pdist
  - cd dev-support/compose/ozone
  - docker-compose up -d

- Start a hadoop cluster without the source and without build
  - copy-paste/download docker-compose.yaml from [https://hadoop.apache.org/docs](https://hadoop.apache.org/docs)
  - docker-compose up -d

For both use case we can support multiple setup (eg. metrics enabled, timeline server, etc.)
Decision points

The proposed method is marked with ❤️ in the next slides

- Automated/manual build
- From source or from binary
- Dedicated or shared branches
- Type of images
- Version support
- Configuration
- User/owner
Build

Automated build by dockerhub ❤

- Only the branch should be defined on dockerhub
- In case of changes on the branch the Dockerfile will be used

+ Container is flagged by dockerhbb
+ ASF supports this approach

- Not as flexible. Can’t build additional binaries without adding the build tools to the container

Build from source and push to dockerhub

+ More flexible. We can use Makefile, do matrix builds, build
- Not marked as automated (not as trustworthy)
- Harder to maintain Readme (should be updated manually)
From binary/source

Build the container using released apache artifacts

Dockerfile:
ADD https://...apache.org/...hadoop//hadoop-2.8.1.tar.gz

+ The hadoop image (eg. apache/hadoop:2.7) could contained exactly the same hadoop which was released
+ ASF policy requires to not distribute snapshot versions to the public audience

Build containers from the source

- Additional build tools should be presented in the image (could be handled by multi-stage builds)
- Hard to achieve with dockerhub based build (as you can use only one Dockerfile there)
- Not compatible with the ASF release policy
Branch usage

Use the same sources as the java code. (branch-2, trunk,...) Dockerfile would be part of the main directory structure

+ More visible, easier to understand
- Harder to release. If 3.0.0-beta1 is released but we see a dockerization related bug, the release should be revoked
- Support for version independent custom image (as the hadoop-runner, see later) is not supported

Dedicated branches for the Dockerfile.

For example, create branches:

- docker/2.7
- docker/2.8
- docker/runner

- More branches in the git
+ We can support older releases
+ More independent from the release process. We can fix the container scripts independent from the releases
+ Easier to debug/understand, only one Dockerfile and a script will be on the branches
Type of images

Use only one or two images ❤️

- One additional command definition for each container runtime
+ More visible what will be happened

I proposed to create two containers:

1. One version independent base image with the starter scripts (hadoop-runner)
2. One container for every hadoop version (apache/hadoop:2.7)

Create images for every component (namenode/datanode/scm/ksm/…)

+ More easy to start containers (don't need to define the starter command
+ Huge number of different containers (hard to to support everything: namenode, datanode, scm, ksm…)
Version support

Use only one branch/container per hadoop major release (hadoop:2.7,hadoop:2.8,hadoop:3.0)

+ More simple setup
+ Minor releases could be archived with docker specific tags in the git repository
- We can’t fix the script for older versions
  (For example if we fix something on docker/hadoop-2.7, it will fix the hadoop/apache:2.7.4 image but not the already released 2.7 image)

Create a different branch for every minor version (2.7.1,2.7.2,2.7.3)

+ More flexible. Could be defined to use a specific version
- Every script fix should be merged to every component
User usage

Run everything as root

+ Much more easier to use
+ No problem about mounting external dist directories

Create hdfs/yarn users to the base container

- More tricky starter script is required. If only one container will be created we need some `su - logic`
+ More realistic. Some features/problems only tested with different users (eg. group mapping)

I think we should support this use case, but the run everything as root could be an option to turn on.
Configuration

Define configuration key/values by environment variables (docker way), but convert them to hadoop xml

+ More easy to use
- Needs additional utility

I believe that it’s much more easier to use. See https://github.com/flokkr/runtime-compose for examples.

The current proposal includes a py script to transform envs to hadoop configs files

Hard-coded or mounted configuration

- Much more harder to maintain/provide configuration as examples

But anyway it’s always an option to mount external directories/configuration.