cloudera®

Docker on Hadoop

Daniel Templeton | Hadoop Committer @ Cloudera

Me

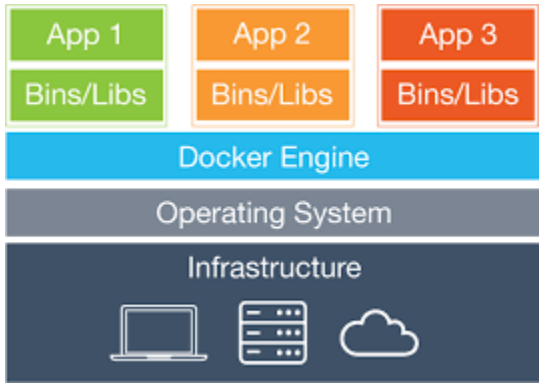
cloudera®

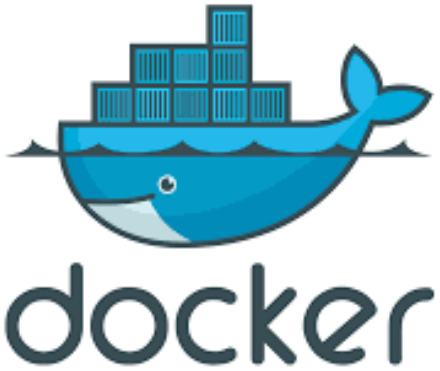
cloudera CERTIFIED
cloudera CONNECT

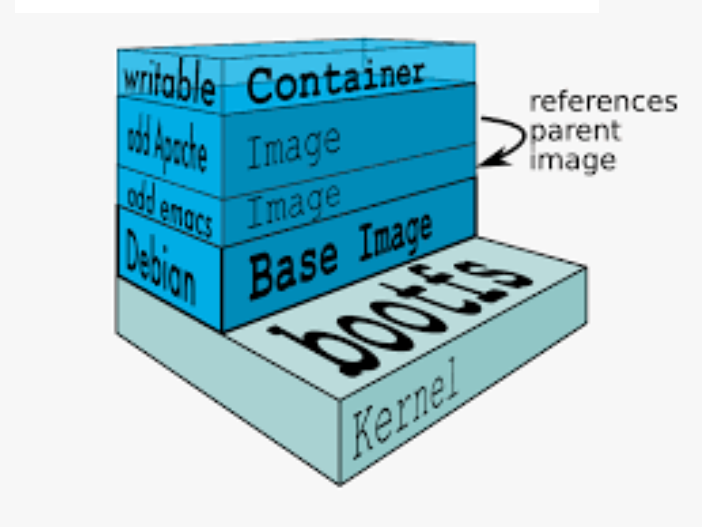
DRMAA
Distributed Resource Management Application API — www.drmaa.org

One Slide on Docker

- Same general idea as a VM
- BUT there's only one OS image
- Partitioned process space
- Layered images
- Image repo



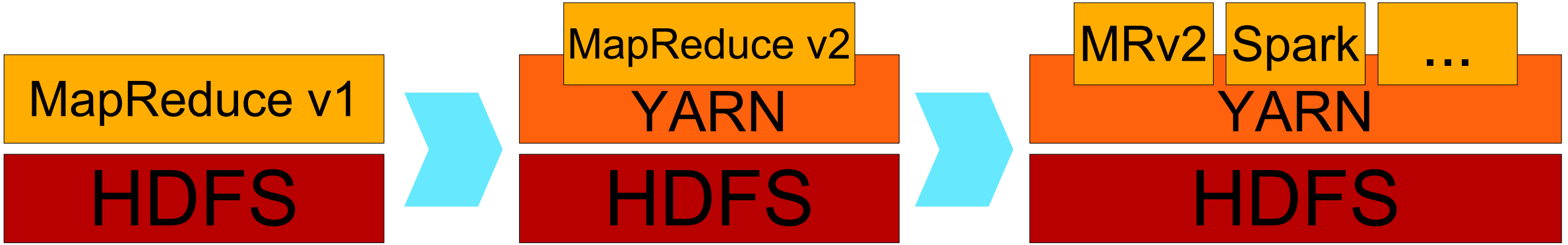




cloudera

One Slide on Hadoop

- Three core components
  - HDFS
  - YARN
  - MapReduce



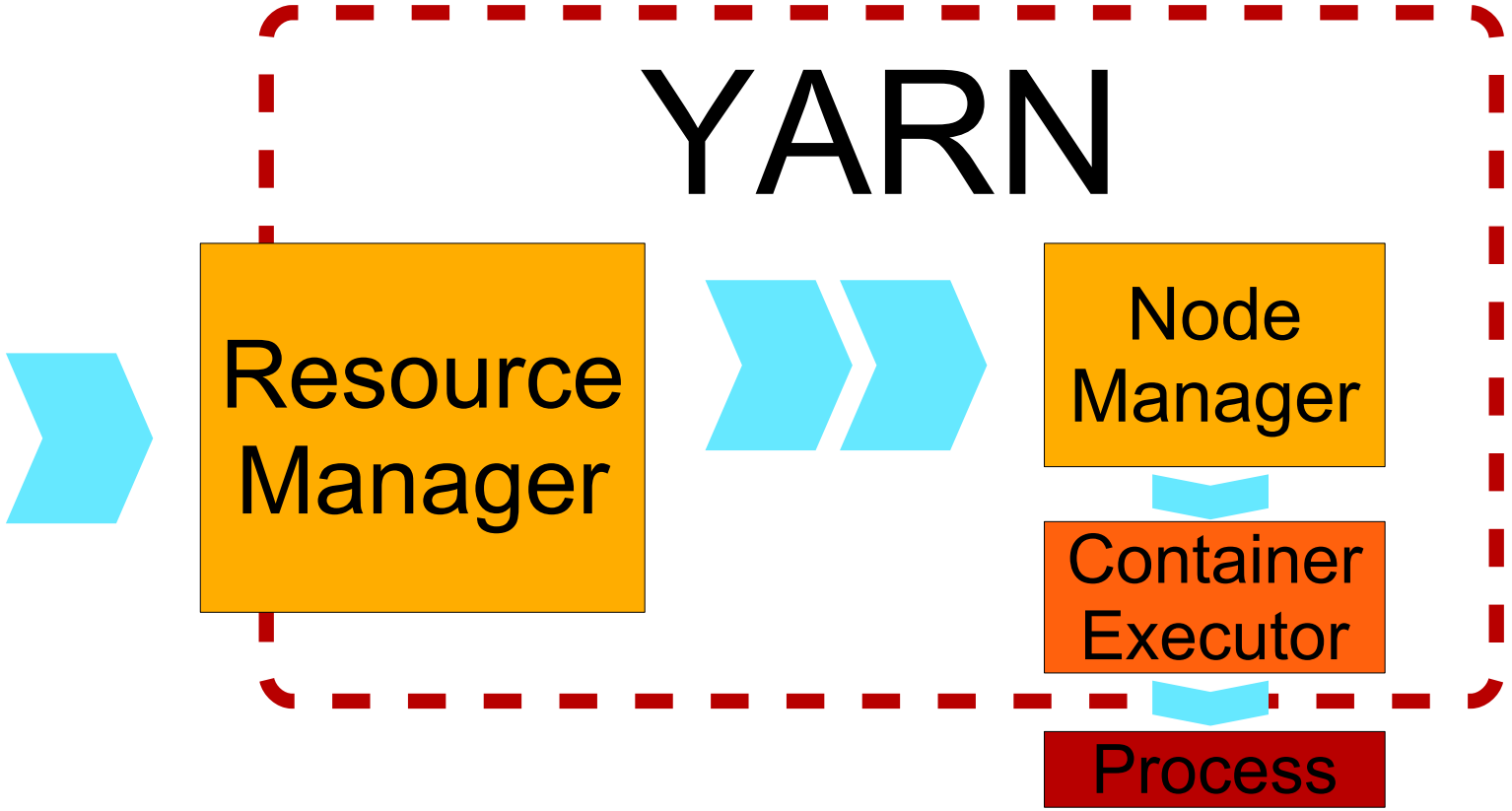
cloudera

Why Docker on Hadoop?

- Process isolation
  - CGroups for resource isolation
  - Adds process
- Environment isolation
  - Control execution environment
    - Libraries
    - JVM
    - OS
  - Unsafe operations

cloudera

Launching Jobs



cloudera

Container Executor

- DefaultContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
- LinuxContainerExecutor
  - Write a launch script
  - Launch native handler
    - Set UID
    - CGroups
    - Fork & exec
  - Required for secure

cloudera

Container Executor

- DefaultContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
- LinuxContainerExecutor
  - Write a launch script
  - Launch native handler
    - Set UID
    - CGroups
    - Fork & exec
  - Required for secure
- DockerContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
  - Docker run

cloudera

Container Executor

- DefaultContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
- LinuxContainerExecutor
  - Write a launch script
  - Launch native handler
    - **OR**
  - Launch Docker handler
    - *docker run*
  - Required for secure
- DockerContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
  - Docker run

cloudera

Container Executor

- DefaultContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
- LinuxContainerExecutor
  - Write a launch script
  - Launch native handler
    - **OR**
  - Launch Docker handler
    - *docker run*
  - Required for secure
- DockerContainerExecutor
  - Write a launch script
  - ProcessBuilder.start()
  - Docker run

cloudera

Secret Formula

How to run a Docker container through YARN

1. Setup LCE
2. Setup Docker
3. Configure yarn-site.xml
4. Configure container-executor.cfg
5. Prepare Docker image
6. Launch job

cloudera

Setup LCE

- LCE uses *container-executor* binary
  - Must be owned by root
  - Group must be same as node manager's group
  - Must have setuid and setgid bits set
  - Must be r+x only by the node manager's group
  - **Owner:** root, **Group:** hadoop, **Mode:** 6050
- Which relies on *container-executor.cfg*
  - Must not be writable by any other than root

Setup Docker

- Docker must be installed on all node manager nodes
- (OR node labels can be used to label the Docker nodes)

  Only capacity scheduler

  Only one label per host
- May be a good idea to pre-cache images that will be used

cloudera

Configure yarn-site.xml

- yarn.nodemanager.container-executor.class =

  org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor
- yarn.nodemanager.linux-container-executor.group =

  hadoop (or whatever group the node manager uses)
- yarn.nodemanager.linux-container-executor.nonsecure-mode.limit-users =

  false (typically)
- yarn.nodemanager.runtime.linux.docker.allowed-container-networks
- yarn.nodemanager.runtime.linux.docker.default-container-network
- yarn.nodemanager.runtime.linux.docker.privileged-containers.allowed
- yarn.nodemanager.runtime.linux.docker.privileged-containers.acl
- ...

cloudera

Configure container-executor.cfg

- yarn.nodemanager.linux-container-executor.group =

  hadoop (or whatever group the node manager uses)
- feature.docker.enabled =

  1 (i.e. true)
- min.user.id
- banned.users
- allowed.system.users
- docker.binary
- ...

cloudera

Prepare the Docker Image

- Application owner (UID) must exist
- Execution requirements

  Hadoop → JRE, Hadoop libraries, env vars

  Must be compatible with cluster and other images
- No entry point, no command

cloudera

Launch the Job

- Do whatever you normally do
- Use of Docker containers managed through env vars

  YARN_CONTAINER_RUNTIME_TYPE

  YARN_CONTAINER_RUNTIME_DOCKER_IMAGE

  YARN_CONTAINER_RUNTIME_DOCKER_RUN_OVERRIDE_DISABLE

  YARN_CONTAINER_RUNTIME_DOCKER_CONTAINER_NETWORK

  YARN_CONTAINER_RUNTIME_DOCKER_RUN_PRIVILEGED_CONTAINER

  YARN_CONTAINER_RUNTIME_DOCKER_LOCAL_RESOURCE_MOUNTS

cloudera

Example: MapReduce

```
$ vars="YARN_CONTAINER_RUNTIME_TYPE=docker”
$ vars=”$vars,YARN_CONTAINER_RUNTIME_DOCKER_IMAGE=hadoop"
$ hadoop jar hadoop-examples.jar pi \
  -Dyarn.app.mapreduce.am.env=$vars \
  -Dmapreduce.map.env=$vars \
  -Dmapreduce.reduce.env=$vars \
  10 100
```

cloudera

Example: Spark

```
$ spark-shell --master yarn \
  --conf spark.executorEnv.YARN_CONTAINER_RUNTIME_TYPE=docker \
  --conf spark.executorEnv.YARN_CONTAINER_RUNTIME_DOCKER_IMAGE=hadoop \
  --conf spark.yarn.AppMasterEnv.YARN_CONTAINER_RUNTIME_DOCKER_IMAGE=hadoop \
  --conf spark.yarn.AppMasterEnv.YARN_CONTAINER_RUNTIME_TYPE=docker
```

cloudera

Caveats

cloudera

Caveats

- Application owner must exist in Docker container
  - Limits flexibility of containers
  - Automatically mounts in /etc/passwd
    - Bad solution
    - Broken
    - Removed in Hadoop 2.9/3.0 (YARN-5394)
  - Discussion on ~~YARN-5360~~ and YARN-4266

cloudera

Caveats

- Application owner must exist in Docker container
- Hadoop artifacts must exist in Docker containers
  - Docker containers must be self-contained
  - HDFS access, deserializing tokens, etc.
  - Versions must be compatible
  - Complicates cluster upgrades
  - YARN-5534 will allow whitelisted volume mounts

Caveats

- Application owner must exist in Docker container
- Hadoop artifacts must exist in Docker containers
- Large images may fail
  - Images that aren't cached are implicitly pulled
  - Large images may take a while
  - MapReduce and Spark time out after 10 minutes
  - YARN-3854 is a step towards a solution

cloudera

Caveats

- Application owner must exist in Docker container
- Hadoop artifacts must exist in Docker containers
- Large images may fail
- No real support for secure image repos
    - Docker stores credentials in client config
    - Always set to $HOME/.docker/config.json
    - YARN-5428 will make the client config configurable

Caveats

- Application owner must exist in Docker container
- Hadoop artifacts must exist in Docker containers
- Large images may fail
- No real support for secure image repos
- Basic support for networks
  - Containers can request any configured network
  - No port mapping
  - No pods
  - No management of overlay networks

cloudera

Caveats

- Application owner must exist in Docker container
- Hadoop artifacts must exist in Docker containers
- Large images may fail
- No real support for secure image repos
- Basic support for networks
- Security implications
  - Privileged container execution
  - Setuid binary
  - Volume mounts (when YARN-3384 is complete)

Caveats

- Application owner must exist in Docker container
- Hadoop artifacts must exist in Docker containers
- Large images may fail
- No real support for secure image repos
- Basic support for networks
- Security Implications
- Not really useful before Hadoop 2.9/3.0
  - YARN-5298: Mounts localized file directories as volumes
  - YARN-4553: CGroups support
  - YARN-4007: Support different networking options
  - YARN-5258: Documentation

cloudera

Apache Slider

- YARN is traditionally a *job* scheduler
- What about services?
- Slider simplifies running a service on YARN
  - Is itself a YARN application
  - Declarative
- Docker support as of Slider 0.80
  - Slider agent calls *docker run*
  - Unrelated to YARN Docker support

Slider in YARN

- Slider core moving into YARN
  - YARN-5079: Native YARN framework layer for services and beyond
- Slider agent is not being integrated
  - Using YARN instead
  - Docker support through YARN
- Currently only in yarn-native-services branch
  - Merge date not set yet
- “Classic” Slider will continue to be available

cloudera

Summary

- Docker adds good things to YARN

  There are a few thorns

- YARN natively supports Docker

  Limited use until Hadoop 2.9/3.0

- Slider natively supports Docker

  Slider is moving into YARN and adopting YARN's Docker support

https://aajisaka.github.io/hadoop-project/hadoop-yarn/hadoop-yarn-site/DockerContainers.html

cloudera

Thank you

Daniel Templeton Cloudera, Inc.
daniel@cloudera.com @templedf