

Cloudera Runtime 7.1.9

Troubleshooting Apache Hadoop YARN

Date published: 2020-07-28

Date modified: 2024-07-19

CLOUdera

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

| | |
|---|-----------|
| Troubleshooting Docker on YARN..... | 4 |
| Troubleshooting on YARN..... | 11 |
| YARN Queue Manager UI behavior in mixed resource allocation mode..... | 11 |
| Troubleshooting for mixed resource allocation mode in YARN Queue Manager..... | 13 |
| Troubleshooting Linux Container Executor..... | 13 |

Troubleshooting Docker on YARN

A list of common Docker on YARN related problem and how to resolve them.

Docker is not enabled

Problem statement

Started an application on Docker, but the containers are running as regular containers.

Root cause

Docker is not enabled.

Resolution

Enable Docker in Cloudera Manager.

YARN_CONTAINER_RUNTIME_TYPE runtime environment variable is not provided during Application submission

Problem statement

Started an application on Docker, but the containers are running as regular containers.

Root cause

YARN_CONTAINER_RUNTIME_TYPE runtime environment variable is not provided during Application submission.

Resolution

Provide the environment variable when submitting the application.

LCE enforces running user to be nobody in an unsecure cluster

Problem statement

On an unsecure cluster, Appattempt exited with exitCode -1000 with diagnostic message:

```
[...]
main : run as user is nobody
main : requested yarn user is yarn
Can't create directory /yarn/nm/usercache/yarn/appcache/applic
ation_1570626013274_0001 - Permission denied
```

Root cause

LCE enforces running user to be nobody in an unsecure cluster if `yarn.nodemanager.linux-container-executor.nonsecure-mode.limit-users` is set.

Resolution

In Cloudera Manager, add the following configuration to the YARN Service Advanced Configuration Snippet (Safety Valve) for `yarn-site.xml` safety-valve by clicking the plus icon:

- Key: `yarn.nodemanager.linux-container-executor.nonsecure-mode.limit-users`
- Value: `false`

Then use a user who has the correct permissions or add more permissive access to these folders for the nobody user. For more information, see [YARN force nobody user on all jobs](#).

The Docker binary is not found

Problem Statement

Container launch fails with the following message:

```
Container launch fails
Exit code: 29
Exception message: Launch container failed
Shell error output: sh: <docker binary path, /usr/bin/docker by
default>: No such file or directory
Could not inspect docker network to get type /usr/bin/docker net
work inspect host --format='{{.Driver}}'.
Error constructing docker command, docker error code=-1, error
message='Unknown error'
```

Root cause

The Docker binary is not found.

Resolution

The Docker binary is either not installed or installed to a different folder. Install Docker binary and provide the path to the binaries by specifying it using the Docker Binary Path (docker.binary) property in Cloudera Manager.

The Docker daemon is not running or does not respond

Problem statement

Container launch fails with the following message:

```
[timestamp] Exception from container-launch.
Container id: container_e06_1570629976081_0004_01_000003
Exit code: 29
Exception message: Launch container failed
Shell error output: Cannot connect to the Docker daemon at unix:/
/var/run/docker.sock. Is the docker daemon running?
Could not inspect docker network to get type /usr/bin/docker ne
twork inspect host --format='{{.Driver}}'.
Error constructing docker command, docker error code=-1, error me
ssage='Unknown error'
```

Root cause

The Docker daemon is not running or does not respond.

Resolution

Start or restart the Docker daemon with the dockerd command.

Docker rpm misses some symbolic link

Problem statement

On Centos 7.5 container launch fails with the following message:

```
[...]
[layer hash]: Pull complete
[layer hash]: Pull complete
Digest: sha256:[sha]
Status: Downloaded newer image for [image]
/usr/bin/docker-current: Error response from daemon: shim error:
docker-runc not installed on system.
```

Root cause

Docker rpm misses some symbolic link.

Resolution

Create the missing symbolic link using the following command in a terminal: `sudo ln -s /usr/libexec/docker/docker-runc-current /usr/bin/docker-runc`

YARN_CONTAINER_RUNTIME_DOCKER_IMAGE is not set

Problem statement

Container launch fails with the following message:

```
[timestamp]Exception from container-launch.
Container id: container_e06_1570629976081_0004_01_000003
Exit code: -1
Exception message: YARN_CONTAINER_RUNTIME_DOCKER_IMAGE not set!
Shell error output: <unknown>
Shell output: <unknown>
```

Root cause

YARN_CONTAINER_RUNTIME_DOCKER_IMAGE is not set.

Resolution

Set the YARN_CONTAINER_RUNTIME_DOCKER_IMAGE environment variable when submitting the application.

Image is not trusted

Problem statement

Container launch fails with the following message:

```
[timestamp] Exception from container-launch.
Container id: container_e06_1570629976081_0004_01_000003
Exit code: 127
Exception message: Launch container failed
Shell error output: image: [image] is not trusted.
Disable mount volume for untrusted image
image: library/ibmjava:8 is not trusted.
Disable cap-add for untrusted image
Docker capability disabled for untrusted image
[...]
```

Root cause

The image is not trusted.

Resolution

Add the image's registry to the list of trusted registries (`docker.trusted.registries`). For example in case of `library/ubuntu:latest`, add the "library" registry to that list.

Docker image does not include the Snappy library

Problem statement

Running the `hadoop-mapreduce-examples pi` job fails with the following error:

```
[...]
[timestamp] INFO mapreduce.Job: map 0% reduce 0%
[timestamp] INFO mapreduce.Job: Task Id : attempt_1570629976081_0001_m_000000_0, Status : FAILED
Error: org/apache/hadoop/util/NativeCodeLoader.buildSupportsSnappy()Z
```

Root cause

The provided Docker image does not include the Snappy library. MapReduce needs this if compression is used and the Snappy codec is chosen for compression.

Resolution

Either add the Snappy library to the image or change the “Compression Codec of MapReduce Map Output” to some other codec

Hadoop UserGroupInformation class does not have access to the user permissions in the host system

Problem statement

Container fails shortly after start with the following exception:

```
Exception in thread "main" org.apache.hadoop.security.KerberosAuthException: failure to login: javax.security.auth.login.LoginException: java.lang.NullPointerException: invalid null input: name
  At com.sun.security.auth.UnixPrincipal.<init>(UnixPrincipal.java:71)
  at com.sun.security.auth.module.UnixLoginModule.login(UnixLoginModule.java:133)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

Root cause

The Hadoop UserGroupInformation class does not have access to the user permissions in the host system.

Resolution

Mount the /etc/passwd to the image. More configuration issues can be found in upstream Hadoop 3.2 documentation: [Launching Applications Using Docker Containers upstream documentation](#).

Kerberos configuration is not mounted for Docker containers

Problem Statement

MapReduce and Spark jobs fail with Docker on a secure cluster. It cannot get Kerberos realm.

```
user@<hostname> /l$ cd /yarn/container-logs/application_1573764921308_0002/container_e147_1573764921308_0002_01_000005
[user@<hostname> container_e147_1573764921308_0002_01_000005]$ ll
total 8
-rw-r--r-- 1 systest yarn    0 Nov 14 12:57 prelaunch.err
-rw-r--r-- 1 systest yarn   70 Nov 14 12:57 prelaunch.out
-rw-r--r-- 1 systest yarn    0 Nov 14 12:57 stderr
-rw-r----- 1 systest yarn    0 Nov 14 12:57 stderr.txt
-rw-r--r-- 1 systest yarn    0 Nov 14 12:57 stdout
-rw-r----- 1 systest yarn    0 Nov 14 12:57 stdout.txt
-rw-r--r-- 1 systest yarn  892 Nov 14 12:57 syslog
[user@<hostname> container_e147_1573764921308_0002_01_000005]$
cat syslog
2019-11-14 20:57:41,765 ERROR [main] org.apache.hadoop.yarn.YarnUncaughtExceptionHandler: Thread Thread[main,5,main] threw an Exception.
java.lang.IllegalArgumentException: Can't get Kerberos realm
  at org.apache.hadoop.security.HadoopKerberosName.setConfiguration(HadoopKerberosName.java:71)
  at org.apache.hadoop.security.UserGroupInformation.initialize(UserGroupInformation.java:330)
```

```

    at org.apache.hadoop.security.UserGroupInformation.setConfig
uration(UserGroupInformation.java:381)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:80)
Caused by: java.lang.IllegalArgumentException
    at javax.security.auth.kerberos.KerberosPrincipal.<init>(Ker
berosPrincipal.java:136)
    at org.apache.hadoop.security.authentication.util.KerberosUt
il.getDefaultRealm(KerberosUtil.java:108)
    at org.apache.hadoop.security.HadoopKerberosName.setConfig
uration(HadoopKerberosName.java:69)
    ... 3 more
[user@<hostname> container_e147_1573764921308_0002_01_000005]$

```

Root cause

Kerberos configuration is not mounted for Docker containers.

Resolution

In case of MapReduce job, add the following environment variable when running the job: `-Dmapreduce.reduce.env=YARN_CONTAINER_RUNTIME_DOCKER_MOUNTS=/etc/krb5.conf:/etc/krb5.conf:ro`

Ensure to add `/etc/krb5.conf` to the Allowed Read-Only Mounts in Cloudera Manager configuration.

Example:

```

yarn jar /opt/cloudera/parcels/CDH-7.0.3-1.cdh7.0.3.p0.1616399/1
ib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi -Dmapreduce
.map.env="YARN_CONTAINER_RUNTIME_TYPE=docker,YARN_CONTAINER_RUNT
IME_DOCKER_IMAGE=library/ibmjava:8,YARN_CONTAINER_RUNTIME_DOCKER
_DELAYED_REMOVAL=true,YARN_CONTAINER_RUNTIME_DOCKER_MOUNTS=/etc/
krb5.conf:/etc/krb5.conf:ro" -Dmapreduce.reduce.env="YARN_CONTAI
NER_RUNTIME_TYPE=docker,YARN_CONTAINER_RUNTIME_DOCKER_IMAGE=libr
ary/ibmjava:8,YARN_CONTAINER_RUNTIME_DOCKER_DELAYED_REMOVAL=true
,YARN_CONTAINER_RUNTIME_DOCKER_MOUNTS=/etc/krb5.conf:/etc/krb5.c
onf:ro" 1 40000

```

In case of Spark job, ensure that mount is added as read-only for `/etc/krb5.conf` as `spark.appMasterEnv` and `spark.executorEnv`:

```

--conf spark.yarn.appMasterEnv.YARN_CONTAINER_RUNTIME_DOCKER_MO
UNTS=/etc/passwd:/etc/passwd:ro,/opt/cloudera/parcels:/opt/cloud
era/parcels:ro,/etc/krb5.conf:/etc/krb5.conf:ro \

```

```

--conf spark.executorEnv.YARN_CONTAINER_RUNTIME_DOCKER_MOUNTS="
/etc/passwd:/etc/passwd:ro,/opt/cloudera/parcels:/opt/cloudera/
parcels:ro,/etc/krb5.conf:/etc/krb5.conf:ro"

```

The `ssl-client.xml` file and the `truststore` file is not mounted for Docker containers using MapReduce

Problem statement

Reducer cannot connect to the shuffle service due to SSL handshake issues.

CLI logs:

```

19/11/15 03:26:02 INFO impl.YarnClientImpl: Submitted application
application_1573810028869_0004
19/11/15 03:26:02 INFO mapreduce.Job: The url to track the job:
<URL>
19/11/15 03:26:02 INFO mapreduce.Job: Running job: job_1573810
028869_0004

```



```

19/11/15 03:26:12 INFO mapreduce.Job: Job job_1573810028869_0004
  running in uber mode : false
19/11/15 03:26:12 INFO mapreduce.Job: map 0% reduce 0%
19/11/15 03:26:23 INFO mapreduce.Job: map 100% reduce 0%
19/11/15 03:27:30 INFO mapreduce.Job: Task Id : attempt_15738100
28869_0004_r_000000_0, Status : FAILED
Error: org.apache.hadoop.mapreduce.task.reduce.Shuffle$ShuffleErr
or: error in shuffle in fetcher#2
    at org.apache.hadoop.mapreduce.task.reduce.Shuffle.run(Shu
ffle.java:136)
    at org.apache.hadoop.mapred.ReduceTask.run(ReduceTask.java:37
7)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:17
4)
    at java.security.AccessController.doPrivileged(AccessControll
er.java:770)
    at javax.security.auth.Subject.doAs(Subject.java:570)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserG
roupInformation.java:1876)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:168
)
Caused by: java.io.IOException: Exceeded MAX_FAILED_UNIQUE_FET
CHES; bailing-out.
    at org.apache.hadoop.mapreduce.task.reduce.ShuffleSchedulerIm
pl.checkReducerHealth(ShuffleSchedulerImpl.java:396)
    at org.apache.hadoop.mapreduce.task.reduce.ShuffleScheduler
Impl.copyFailed(ShuffleSchedulerImpl.java:311)
    at org.apache.hadoop.mapreduce.task.reduce.Fetcher.openShuffl
eUrl(Fetcher.java:291)
    at org.apache.hadoop.mapreduce.task.reduce.Fetcher.copyFromHo
st(Fetcher.java:330)
    at org.apache.hadoop.mapreduce.task.reduce.Fetcher.run(Fetc
her.java:198)

```

NodeManager logs:

```

2019-11-15 03:30:16,323 INFO org.apache.hadoop.yarn.server.nodem
anager.NodeStatusUpdaterImpl: Removed completed containers from
NM context: [container_e149_1573810028869_0004_01_000005]
2019-11-15 03:30:50,812 ERROR org.apache.hadoop.mapred.Shuffle
Handler: Shuffle error:
javax.net.ssl.SSLException: Received fatal alert: certificate_un
known
    at sun.security.ssl.Alerts.getSSLException(Alerts.java
:208)
    at sun.security.ssl.SSLEngineImpl.fatal(SSLEngineImpl.
java:1666)
    at sun.security.ssl.SSLEngineImpl.fatal(SSLEngineImpl.jav
a:1634)
    at sun.security.ssl.SSLEngineImpl.recvAlert(SSLEngineImp
l.java:1800)
    at sun.security.ssl.SSLEngineImpl.readRecord(SSLEngineI
mpl.java:1083)
    at sun.security.ssl.SSLEngineImpl.readNetRecord(SSLEngine
Impl.java:907)
    at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.ja
va:781)
    at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:624)
    at org.jboss.netty.handler.ssl.SslHandler.unwrap(SslHa
ndler.java:1218)
    at org.jboss.netty.handler.ssl.SslHandler.decode(SslHan
dler.java:852)

```

```

    at org.jboss.netty.handler.codec.frame.FrameDecoder.callDecode(FrameDecoder.java:425)
    at org.jboss.netty.handler.codec.frame.FrameDecoder.messageReceived(FrameDecoder.java:303)
    at org.jboss.netty.channel.SimpleChannelUpstreamHandler.handleUpstream(SimpleChannelUpstreamHandler.java:70)
    at org.jboss.netty.channel.DefaultChannelPipeline.sendUpstream(DefaultChannelPipeline.java:564)
    at org.jboss.netty.channel.DefaultChannelPipeline.sendUpstream(DefaultChannelPipeline.java:559)
    at org.jboss.netty.channel.Channels.fireMessageReceived(Channels.java:268)
    at org.jboss.netty.channel.Channels.fireMessageReceived(Channels.java:255)
    at org.jboss.netty.channel.socket.nio.NioWorker.read(NioWorker.java:88)
    at org.jboss.netty.channel.socket.nio.AbstractNioWorker.process(AbstractNioWorker.java:108)
    at org.jboss.netty.channel.socket.nio.AbstractNioSelector.run(AbstractNioSelector.java:337)
    at org.jboss.netty.channel.socket.nio.AbstractNioWorker.run(AbstractNioWorker.java:89)
    at org.jboss.netty.channel.socket.nio.NioWorker.run(NioWorker.java:178)
    at org.jboss.netty.util.ThreadRenamingRunnable.run(ThreadRenamingRunnable.java:108)
    at org.jboss.netty.util.internal.DeadLockProofWorker$1.run(DeadLockProofWorker.java:42)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
2019-11-15 03:30:50,812 ERROR org.apache.hadoop.mapred.ShuffleHandler: Shuffle error [id: 0xf95ad8ab, /10.65.53.21:44366 => /10.65.53.21:13562] EXCEPTION: javax.net.ssl.SSLException: Received fatal alert: certificate_unknown
2019-11-15 03:30:51,156 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.ContainerManagerImpl: Stopping container with container Id: container_e149_1573810028869_0004_01_000006

```

NodeManager logs (Exception):

```

2019-11-15 03:30:50,812 ERROR org.apache.hadoop.mapred.ShuffleHandler: Shuffle error:
javax.net.ssl.SSLException: Received fatal alert: certificate_unknown
[...]
2019-11-15 03:30:50,812 ERROR org.apache.hadoop.mapred.ShuffleHandler: Shuffle error [id: 0xf95ad8ab, /10.65.53.21:44366 => /10.65.53.21:13562] EXCEPTION: javax.net.ssl.SSLException: Received fatal alert: certificate_unknown
2019-11-15 03:30:51,156 INFO org.apache.hadoop.yarn.server.nodemanager.containermanager.ContainerManagerImpl: Stopping container with container Id: container_e149_1573810028869_0004_01_000006

```

Root cause

For normal containers, the file `ssl-client.xml` defines the SSL settings and it is on the classpath (normally under directory: `/etc/hadoop/conf.cloudera.YARN-1/ssl-client.xml`). Therefore, it has to be mounted for Docker containers using MapReduce. Since the `ssl-client.xml` file refers to the truststore file as well, that also had to be mounted.

Resolution

Add the following when running the job:

```
-Dmapreduce.reduce.env=YARN_CONTAINER_RUNTIME_DOCKER_MOUNTS="/etc/hadoop/conf.cloudera.YARN-1/ssl-client.xml:/etc/hadoop/conf.cloudera.YARN-1/ssl-client.xml:ro,/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks:/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks:ro"
```

Ensure to add `/etc/hadoop/conf.cloudera.YARN-1/ssl-client.xml` and `/var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks` to the Allowed Read-Only Mounts in Cloudera Manager.

Note, that the location of the truststore can vary, so verify its location from the `ssl-client.xml` file. You can access that file in Clouder Manager through the Processes view for NodeManager.

Troubleshooting on YARN

General troubleshooting procedures to diagnose some of the commonly encountered issues in YARN.

The Kill application button does not display in the YARN UI

Problem statement

The YARN UI does not display the Kill application button.

Root cause

Kerberos is not enabled.

Resolution

Enable Kerberos in order to view the Kill application button.

With the application state API, you can query the state of a submitted app as well as kill a running app by modifying the state of a running app using a PUT request with the state set to "KILLED".

To perform the PUT operation, authentication has to be set up for the Resource Manager (RM) web services. See [Enabling Kerberos Authentication for CDP](#) linked below.

Related Information

[Enabling Kerberos Authentication for CDP](#)

YARN Queue Manager UI behavior in mixed resource allocation mode

The mixed resource allocation mode in YARN is only supported through safety valves. If you open the Queue Manager UI or try to access Queue Manager APIs when mixed calculation mode is turned on, Queue Manager blocks the UI or APIs and informs you that mixed calculation mode is turned on, Queue Manager is inaccessible until this feature is fully compatible.

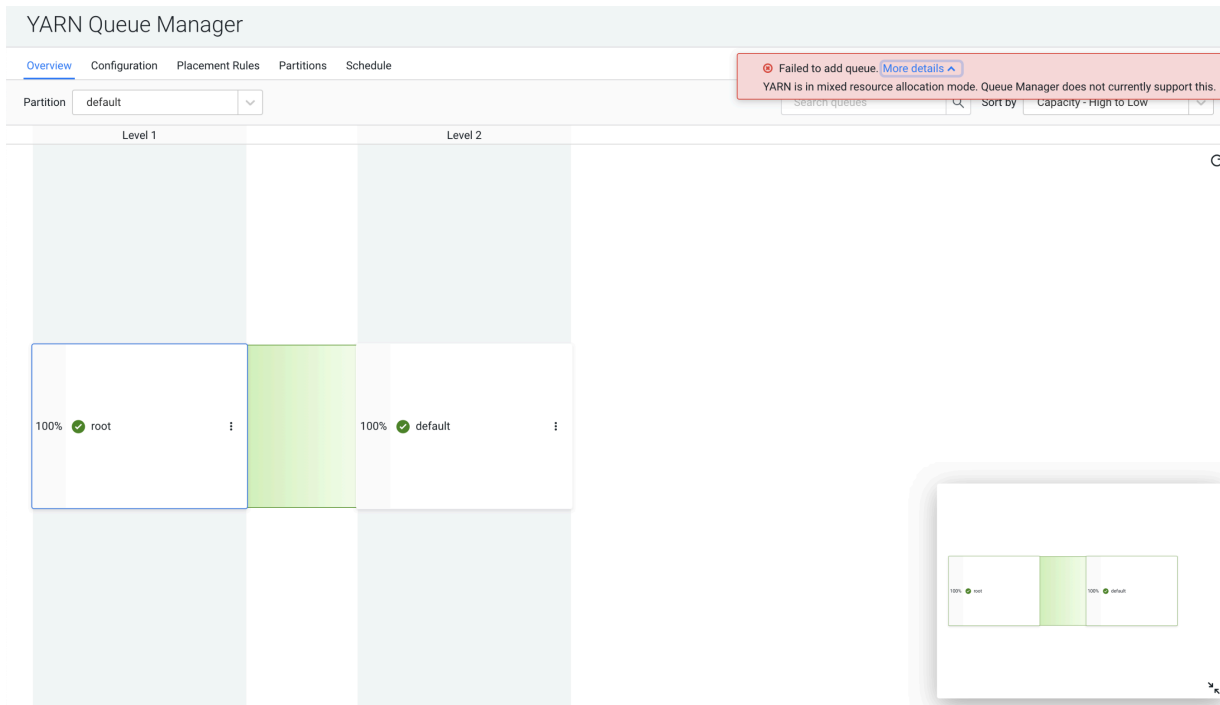
If the flag ``yarn.scheduler.capacity.legacy-queue-mode.enabled`` is not set or the property is missing, mixed resource allocation mode is not enabled in YARN. The detection of mixed allocation mode is determined by this flag in the capacity scheduler safety valve XML.

For more details on Mixed resource allocation, see the links provided below.

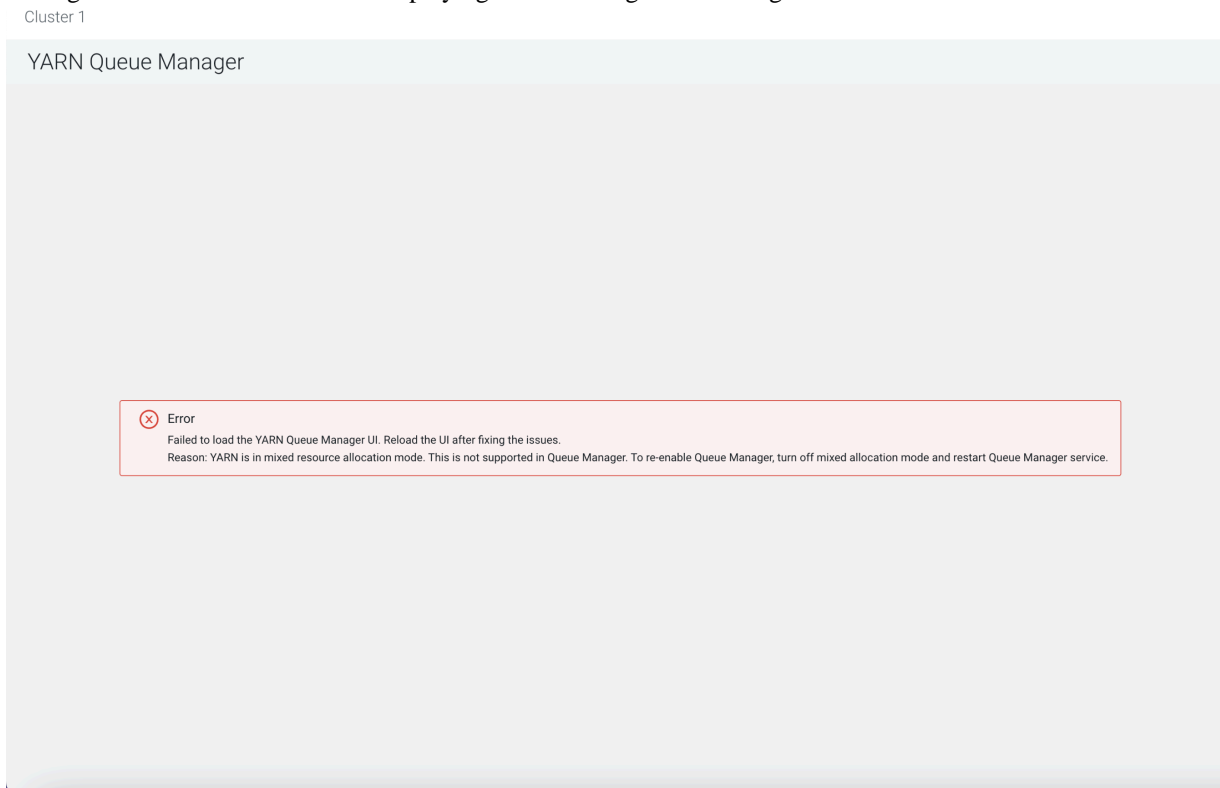
Scenarios

The following scenarios describe how the YARN Queue Manager UI behaves in mixed resource allocation mode in YARN.

- If YARN mixed resource allocation mode is activated or detected while the YARN Queue Manager is already running, it causes YARN Queue Manager UI to fail to load, resulting in an error message being displayed in the UI.



- If you start the YARN Queue Manager while YARN is already in mixed resource calculation mode, Queue Manager UI and APIs are blocked displaying the following error message.



Related Information[Mixed resource allocation mode \(Technical Preview\)](#)

Troubleshooting for mixed resource allocation mode in YARN Queue Manager

You can reset YARN Queue Manager UI and restore its functionality to troubleshoot YARN Queue Manager UI behavior in mixed resource allocation mode from Cloudera Manager.

Procedure

1. In Cloudera Manager, select the YARN service.
2. Go to the Configuration tab.
3. If the flag is not set or the property is missing, mixed resource allocation mode will not be enabled in YARN. Set the following property `yarn.scheduler.capacity.legacy-queue-mode.enabled` value to 'true' in the Capacity Scheduler Configuration Advanced Configuration Snippet (Safety Valve) field to use the mixed allocation mode:

```
<property>
                                <name>yarn.scheduler.capacity.legacy-queue-m
ode.enabled</name>
                                <value>>false</value>
</property>
```

4. Click Save Changes.
5. Click Actions Restart to restart YARN.
6. Click Actions Restart to restart YARN Queue Manager.

Troubleshooting Linux Container Executor

A list of numeric error codes communicated by the container-executor to the NodeManager that appear in the `/var/log/hadoop-yarn NodeManager` log.

Table 1: Numeric error codes that are applicable to the container-executor in YARN, but are used by the LinuxContainerExecutor only.

| Numeric Code | Name | Description |
|--------------|------------------------------------|--|
| 1 | INVALID_ARGUMENT_NUMBER | <ul style="list-style-type: none"> • Incorrect number of arguments provided for the given container-executor command • Failure to initialize the container localizer |
| 2 | INVALID_USER_NAME | The user passed to the container-executor does not exist. |
| 3 | INVALID_COMMAND_PROVIDED | The container-executor does not recognize the command it was asked to run. |
| 5 | INVALID_NM_ROOT | The passed NodeManager root does not match the configured NodeManager root (<code>yarn.nodemanager.local-dirs</code>), or does not exist. |
| 6 | SETUID_OPER_FAILED | Either could not read the local groups database, or could not set UID or GID |
| 7 | UNABLE_TO_EXECUTE_CONTAINER_SCRIPT | The container-executor could not run the container launcher script. |

| Numeric Code | Name | Description |
|--------------|------------------------------------|--|
| 8 | UNABLE_TO_SIGNAL_CONTAINER | The container-executor could not signal the container it was passed. |
| 9 | INVALID_CONTAINER_PID | The PID passed to the container-executor was negative or 0. |
| 18 | OUT_OF_MEMORY | The container-executor couldn't allocate enough memory while reading the container-executor.cfg file, or while getting the paths for the container launcher script or credentials files. |
| 20 | INITIALIZE_USER_FAILED | Couldn't get, stat, or secure the per-user NodeManager directory. |
| 21 | UNABLE_TO_BUILD_PATH | The container-executor couldn't concatenate two paths, most likely because it ran out of memory. |
| 22 | INVALID_CONTAINER_EXEC_PERMISSIONS | The container-executor binary does not have the correct permissions set. |
| 24 | INVALID_CONFIG_FILE | The container-executor.cfg file is missing, malformed, or has incorrect permissions. |
| 25 | SETSID_OPER_FAILED | Could not set the session ID of the forked container. |
| 26 | WRITE_PIDFILE_FAILED | Failed to write the value of the PID of the launched container to the PID file of the container. |
| 255 | Unknown Error | <p>This error has several possible causes. Some common causes are:</p> <ul style="list-style-type: none"> User accounts on your cluster have a user ID less than the value specified for the min.user.id property in the container-executor.cfg file. The default value is 1000 which is appropriate on Ubuntu systems, but may not be valid for your operating system. For information about setting min.user.id in the container-executor.cfg file. This error is often caused by previous errors; look earlier in the log file for possible causes. |

Table 2: Exit status codes apply to all containers in YARN. These exit status codes are part of the YARN framework and are in addition to application specific exit codes that can be set.

| Numeric Code | Name | Description |
|--------------|----------------------|--|
| 0 | SUCCESS | Container has finished successfully. |
| -1000 | INVALID | Initial value of the container exit code. A container that does not have a COMPLETED state will always return this status. |
| -100 | ABORTED | Containers killed by the framework, either due to being released by the application or being 'lost' due to node failures, for example. |
| -101 | DISKS_FAILED | Container exited due to local disks issues in the NodeManager node. This occurs when the number of good nodemanager-local-directories or nodemanager-log-directories drops below the health threshold. |
| -102 | PREEMPTED | Containers preempted by the framework. This does not count towards a container failure in most applications. |
| -103 | KILLED_EXCEEDED_VMEM | Container terminated because of exceeding allocated virtual memory limit. |

| Numeric Code | Name | Description |
|--------------|-----------------------------|--|
| -104 | KILLED_EXCEEDED_PMEM | Container terminated because of exceeding allocated physical memory limit. |
| -105 | KILLED_BY_APPMASTER | Container was terminated on request of the application master. |
| -106 | KILLED_BY_RESOURCEMANAGER | Container was terminated by the resource manager. |
| -107 | KILLED_AFTER_APP_COMPLETION | Container was terminated after the application finished. |