Cloudera Runtime 7.3.2

Troubleshooting Apache Hadoop YARN

Date published: 2020-07-28
Date modified: 2026-03-31

https://docs.cloudera.com/

Legal Notice

Contents

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 Cloudera linked below.

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.

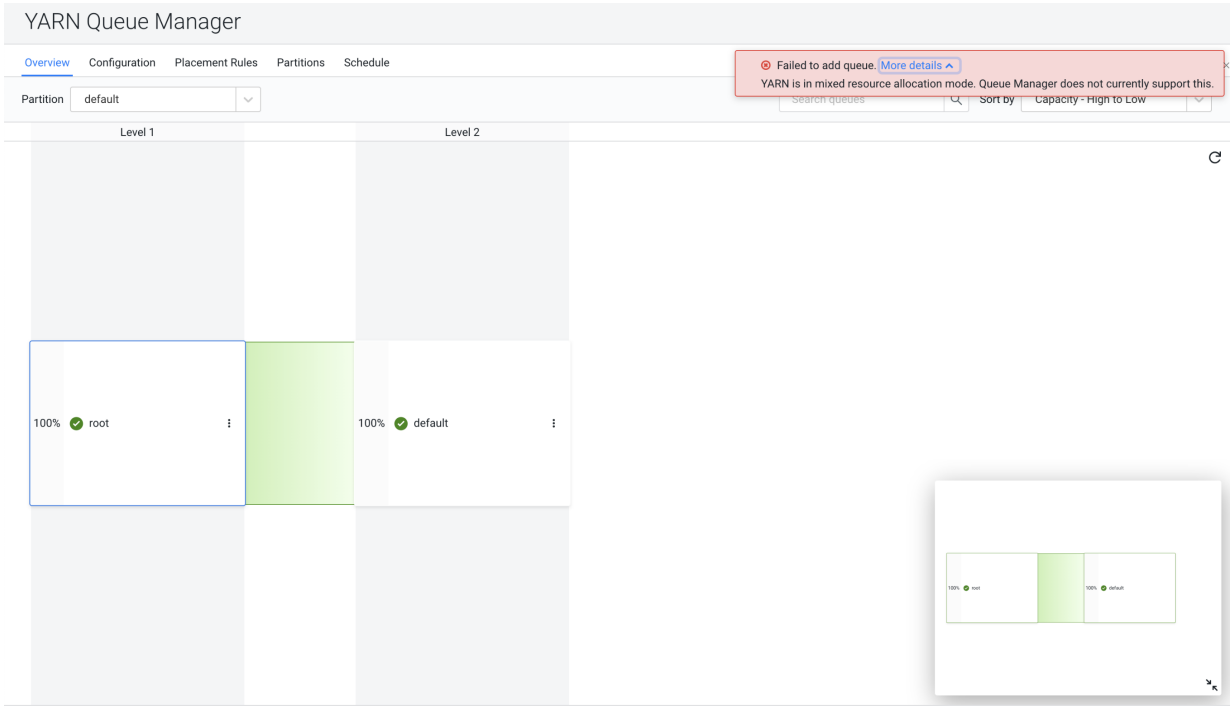
If the flag `yarn.scheduler.capacity.legacy-queue-mode.enabled` is not set or the property is missing, mixed resource allocation mode is not enabled in YARN. The detection of mixed allocation mode is determined by this flag in the capacity scheduler safety valve XML.

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

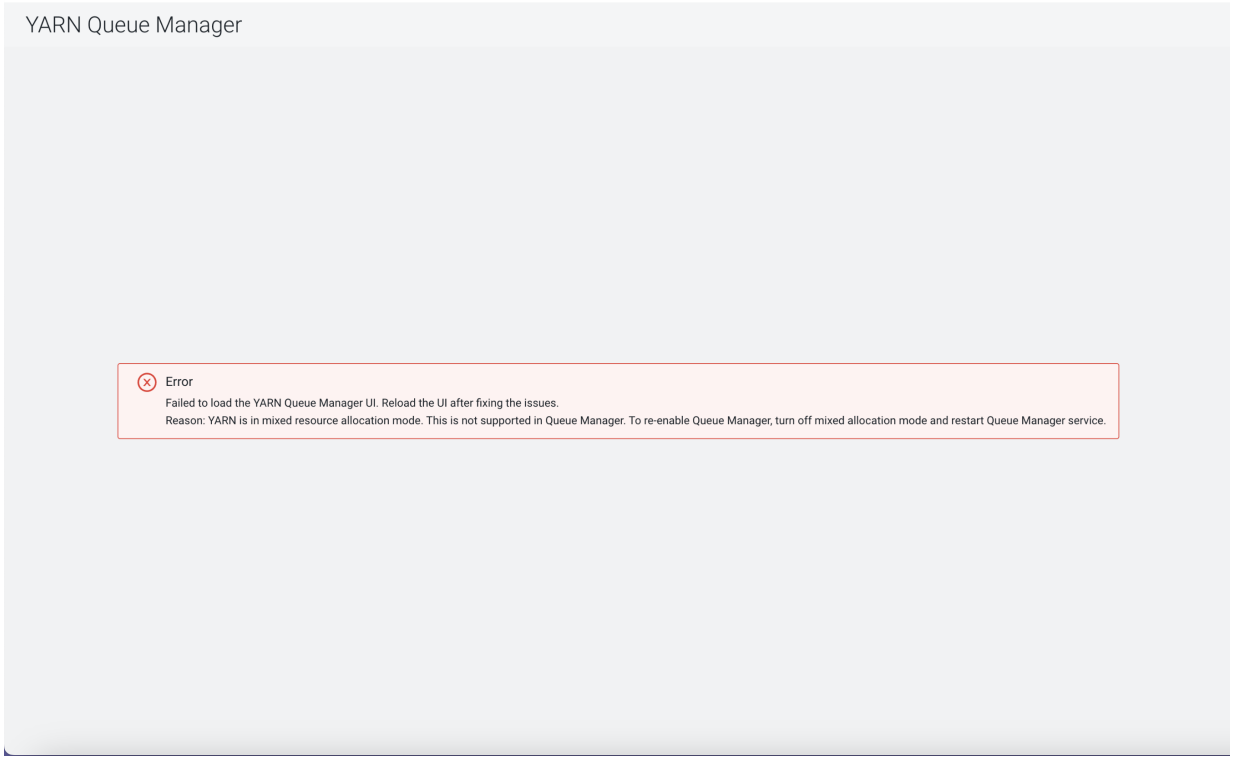
Scenarios

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-mode.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	• 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 (yarn.nodemanager.local-dirs), 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.
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.

Numeric Code	Name	Description
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.
24	Error starting NodeManager	NodeManager can fail to start up if the nosuid option is set on the file system where the container-executor binary resides. nosuid prevents the setuid bit on executable from taking effect. This means that the container-executor binary that has the setuid bit set with "root" privileges, is unable to access the container-executor.cfg configuration file owned by "root" and results in error.
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	This error has several possible causes. Some common causes are: • User accounts on your cluster have a user ID less than the value specified for the min.user.id property in the cont ainer-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 succesfully.
-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.

Numeric Code	Name	Description
-103	KILLED_EXCEEDED_VMEM	Container terminated because of exceeding allocated virtual memory limit.
-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.