Item 195
**git_comments:**

1. try and see if NodeManagers are up, otherwise the Flink job will not have enough resources to run

**git_commits:**

1. **summary:** [FLINK-10368] Harden Dockerized Kerberos tests by waiting for NM to be up
**message:** [FLINK-10368] Harden Dockerized Kerberos tests by waiting for NM to be up Before, we didn't wait for Yarn NodeManagers to be up. This meant that sometimes the Flink Job would not have enough resources to run.

**github_issues:**

**github_issues_comments:**

**github_pulls:**

**github_pulls_comments:**

**github_pulls_reviews:**

**jira_issues:**

1. **summary:** 'Kerberized YARN on Docker test' unstable
**description:** Running Kerberized YARN on Docker test end-to-end test failed on an AWS instance. The problem seems to be that the NameNode went into safe-mode due to limited resources. {code} SLF4J: Class path contains multiple SLF4J bindings. SLF4J: Found binding in [jar:file:/home/hadoop-user/flink-1.6.1/lib/slf4j-log4j12-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class] SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.8.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class] SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation. SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory] 2018-09-19 09:04:39,201 INFO org.apache.hadoop.security.UserGroupInformation - Login successful for user hadoop-user using keytab file /home/hadoop-user/hadoop-user.keytab 2018-09-19 09:04:39,453 INFO org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at master.docker-hadoop-cluster-network/172.22.0.3:8032 2018-09-19 09:04:39,640 INFO org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at master.docker-hadoop-cluster-network/172.22.0.3:10200 2018-09-19 09:04:39,656 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli - No path for the flink jar passed. Using the location of class org.apache.flink.yarn.YarnClusterDescriptor to locate the jar 2018-09-19 09:04:39,656 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli - No path for the flink jar passed. Using the location of class org.apache.flink.yarn.YarnClusterDescriptor to locate the jar 2018-09-19 09:04:39,901 INFO org.apache.flink.yarn.AbstractYarnClusterDescriptor - Cluster specification: ClusterSpecification{masterMemoryMB=2000, taskManagerMemoryMB=2000, numberTaskManagers=3, slotsPerTaskManager=1} 2018-09-19 09:04:40,286 WARN org.apache.flink.yarn.AbstractYarnClusterDescriptor - The configuration directory ('/home/hadoop-user/flink-1.6.1/conf') contains both LOG4J and Logback configuration files. Please delete or rename one of them. ------------------------------------------------------------ The program finished with the following exception: org.apache.flink.client.deployment.ClusterDeploymentException: Couldn't deploy Yarn session cluster at org.apache.flink.yarn.AbstractYarnClusterDescriptor.deploySessionCluster(AbstractYarnClusterDescriptor.java:420) at org.apache.flink.client.cli.CliFrontend.runProgram(CliFrontend.java:259) at org.apache.flink.client.cli.CliFrontend.run(CliFrontend.java:215) at org.apache.flink.client.cli.CliFrontend.parseParameters(CliFrontend.java:1044) at org.apache.flink.client.cli.CliFrontend.lambda$main$11(CliFrontend.java:1120) at java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.Subject.doAs(Subject.java:422) at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1836) at org.apache.flink.runtime.security.HadoopSecurityContext.runSecured(HadoopSecurityContext.java:41) at org.apache.flink.client.cli.CliFrontend.main(CliFrontend.java:1120) Caused by: org.apache.hadoop.hdfs.server.namenode.SafeModeException: Cannot create file/user/hadoop-user/.flink/application_1537266361291_0099/lib/slf4j-log4j12-1.7.7.jar. Name node is in safe mode. Resources are low on NN. Please add or free up more resources then turn off safe mode manually. NOTE: If you turn off safe mode before adding resources, the NN will immediately return to safe mode. Use "hdfs dfsadmin -safemode leave" to turn safe mode off. NamenodeHostName:master.docker-hadoop-cluster-network at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.newSafemodeException(FSNamesystem.java:1407) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkNameNodeSafeMode(FSNamesystem.java:1395) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.startFileInt(FSNamesystem.java:2278) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.startFile(FSNamesystem.java:2223) at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.create(NameNodeRpcServer.java:728) at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.create(ClientNamenodeProtocolServerSideTranslatorPB.java:413) at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java) at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:447) at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989) at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850) at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793) at java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.Subject.doAs(Subject.java:422) at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840) at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489) at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62) at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45) at java.lang.reflect.Constructor.newInstance(Constructor.java:423) at org.apache.hadoop.ipc.RemoteException.instantiateException(RemoteException.java:121) at org.apache.hadoop.ipc.RemoteException.unwrapRemoteException(RemoteException.java:88) at org.apache.hadoop.hdfs.DFSOutputStream.newStreamForCreate(DFSOutputStream.java:270) at org.apache.hadoop.hdfs.DFSClient.create(DFSClient.java:1274) at org.apache.hadoop.hdfs.DFSClient.create(DFSClient.java:1216) at org.apache.hadoop.hdfs.DistributedFileSystem$8.doCall(DistributedFileSystem.java:473) at org.apache.hadoop.hdfs.DistributedFileSystem$8.doCall(DistributedFileSystem.java:470) at org.apache.hadoop.fs.FileSystemLinkResolver.resolve(FileSystemLinkResolver.java:81) at org.apache.hadoop.hdfs.DistributedFileSystem.create(DistributedFileSystem.java:470) at org.apache.hadoop.hdfs.DistributedFileSystem.create(DistributedFileSystem.java:411) at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:929) at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:910) at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:807) at org.apache.hadoop.fs.FileUtil.copy(FileUtil.java:368) at org.apache.hadoop.fs.FileUtil.copy(FileUtil.java:341) at org.apache.hadoop.fs.FileSystem.copyFromLocalFile(FileSystem.java:2002) at org.apache.flink.yarn.Utils.setupLocalResource(Utils.java:162) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.setupSingleLocalResource(AbstractYarnClusterDescriptor.java:1139) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.access$000(AbstractYarnClusterDescriptor.java:111) at org.apache.flink.yarn.AbstractYarnClusterDescriptor$1.visitFile(AbstractYarnClusterDescriptor.java:1200) at org.apache.flink.yarn.AbstractYarnClusterDescriptor$1.visitFile(AbstractYarnClusterDescriptor.java:1188) at java.nio.file.Files.walkFileTree(Files.java:2670) at java.nio.file.Files.walkFileTree(Files.java:2742) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.uploadAndRegisterFiles(AbstractYarnClusterDescriptor.java:1188) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.startAppMaster(AbstractYarnClusterDescriptor.java:800) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.deployInternal(AbstractYarnClusterDescriptor.java:542) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.deploySessionCluster(AbstractYarnClusterDescriptor.java:413) ... 9 more Caused by: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.hdfs.server.namenode.SafeModeException): Cannot create file/user/hadoop-

user/.flink/application_1537266361291_0099/lib/slf4j-log4j12-1.7.7.jar. Name node is in safe mode. Resources are low on NN. Please add or free up more resources then turn off safe mode manually. NOTE: If you turn off safe mode before adding resources, the NN will immediately return to safe mode. Use "hdfs dfsadmin -safemode leave" to turn safe mode off. NamenodeHostName:master.docker-hadoop-cluster-network at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.newSafemodeException(FSNamesystem.java:1407) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkNameNodeSafeMode(FSNamesystem.java:1395) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.startFileInt(FSNamesystem.java:2278) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.startFile(FSNamesystem.java:2223) at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.create(NameNodeRpcServer.java:728) at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.create(ClientNamenodeProtocolServerSideTranslatorPB.java:413) at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java) at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:447) at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:989) at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:850) at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:793) at java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.Subject.doAs(Subject.java:422) at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840) at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2489) at org.apache.hadoop.ipc.Client.getRpcResponse(Client.java:1489) at org.apache.hadoop.ipc.Client.call(Client.java:1435) at org.apache.hadoop.ipc.Client.call(Client.java:1345) at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:227) at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:116) at com.sun.proxy.$Proxy14.create(Unknown Source) at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.create(ClientNamenodeProtocolTranslatorPB.java:297) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:498) at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:409) at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invokeMethod(RetryInvocationHandler.java:163) at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invoke(RetryInvocationHandler.java:155) at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invokeOnce(RetryInvocationHandler.java:95) at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:346) at com.sun.proxy.$Proxy15.create(Unknown Source) at org.apache.hadoop.hdfs.DFSOutputStream.newStreamForCreate(DFSOutputStream.java:265) ... 33 more Running the Flink job failed, might be that the cluster is not ready yet. We have been trying for 795 seconds, retrying ... {code} I think it would be good to harden the test.

**jira_issues_comments:**

1. [~till.rohrmann] said we should at checks to the code that see if the cluster could be brought up and then exit the test gracefully without failing when it can't be brought up.
2. I encountered another problem: {code}The program finished with the following exception: org.apache.flink.client.deployment.ClusterDeploymentException: Couldn't deploy Yarn session cluster at org.apache.flink.yarn.AbstractYarnClusterDescriptor.deploySessionCluster(AbstractYarnClusterDescriptor.java:419) at org.apache.flink.client.cli.CliFrontend.runProgram(CliFrontend.java:261) at org.apache.flink.client.cli.CliFrontend.run(CliFrontend.java:215) at org.apache.flink.client.cli.CliFrontend.parseParameters(CliFrontend.java:1035) at org.apache.flink.client.cli.CliFrontend.lambda$main$9(CliFrontend.java:1111) at java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.Subject.doAs(Subject.java:422) at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1844) at org.apache.flink.runtime.security.HadoopSecurityContext.runSecured(HadoopSecurityContext.java:41) at org.apache.flink.client.cli.CliFrontend.main(CliFrontend.java:1111) Caused by: org.apache.flink.configuration.IllegalConfigurationException: The number of requested virtual cores per node 1 exceeds the maximum number of virtual cores 0 available in the Yarn Cluster. Please note that the number of virtual cores is set to the number of task slots by default unless configured in the Flink config with 'yarn.containers.vcores.' at org.apache.flink.yarn.AbstractYarnClusterDescriptor.isReadyForDeployment(AbstractYarnClusterDescriptor.java:299) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.deployInternal(AbstractYarnClusterDescriptor.java:491) at org.apache.flink.yarn.AbstractYarnClusterDescriptor.deploySessionCluster(AbstractYarnClusterDescriptor.java:412) ... 9 more {code} Apparently, the cluster thinks that it has only {{0}} vcores. This might be a setup issue.
3. dawidwys opened a new pull request #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965 Hardened the kerberized yarn e2e test, especially having in mind environments poor in resources. Changes: * wait for whole bootstrapping script to execute on master node before submitting job * added check that all containers are up and running before submitting job * reduced memory requirements for the kerberized yarn test ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
4. **body:** I've tried to harden the test by: * waiting for whole bootstrapping script to execute on master node before submitting job - the bootstrapping script performs multiple operations on hdfs, we should catch initialization exceptions sooner and potentially fail gracefully, we also limit the number of false exceptions during job submission * adding check that all containers are up and running before submitting job - that should catch cases when not enough cores are available, that might happen if slave nodes failed * reducing memory requirements for the kerberized yarn test - that should decrease the chance that the cluster will be short on resources [~till.rohrmann] Do you think those changes are sufficient?
   **label:** code-design
5. aljoscha commented on a change in pull request #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#discussion_r230008139 ########## File path: flink-end-to-end-tests/test-scripts/test_yarn_kerberos_docker.sh ########## @@ -60,19 +64,41 @@ function cluster_shutdown { trap cluster_shutdown INT trap cluster_shutdown EXIT -until docker cp $FLINK_TARBALL_DIR/$FLINK_TARBALL master:/home/hadoop-user/; do - # we're retrying this one because we don't know yet if the container is ready - echo "Uploading Flink tarball to docker master failed, retrying ..." - sleep 5 +# wait for kerberos to be set up +start_time=$(date +%s) +until docker logs master 2>&1 | grep -q "Finished master initialization"; do + current_time=$(date +%s) + time_diff=$((current_time - start_time)) + + if [ $time_diff -ge $MAX_RETRY_SECONDS ]; then + echo "ERROR: Could not start hadoop cluster. Aborting..." + exit 0 + else + echo "Waiting for hadoop cluster to come up. We have been trying for $time_diff seconds, retrying ..." + sleep 10 + fi done +# perform health checks +if ! { [ $(docker inspect -f '{{.State.Running}}' master 2>&1) = 'true' ] && + [ $(docker inspect -f '{{.State.Running}}' slave1 2>&1) = 'true' ] && + [ $(docker inspect -f '{{.State.Running}}' slave2 2>&1) = 'true' ] && + [ $(docker inspect -f '{{.State.Running}}' kdc 2>&1) = 'true' ]; }; +then + echo "ERROR: Could not start hadoop cluster. At least one of the containers failed. Aborting..." + exit 0 Review comment: Isn't exit code `1` the exit code for failure? ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
6. aljoscha commented on a change in pull request #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#discussion_r230008106 ########## File path: flink-end-to-end-tests/test-scripts/test_yarn_kerberos_docker.sh ########## @@ -60,19 +64,41 @@ function cluster_shutdown { trap cluster_shutdown INT trap cluster_shutdown EXIT -until docker cp $FLINK_TARBALL_DIR/$FLINK_TARBALL master:/home/hadoop-user/; do - # we're retrying this one because we don't know yet if the container is ready - echo "Uploading Flink tarball to docker master failed, retrying ..." - sleep 5 +# wait for kerberos to be set up +start_time=$(date +%s) +until docker logs master 2>&1 | grep -q "Finished master initialization"; do + current_time=$(date +%s) + time_diff=$((current_time - start_time)) + + if [ $time_diff -ge $MAX_RETRY_SECONDS ]; then + echo "ERROR: Could not start hadoop cluster. Aborting..." + exit 0 Review comment: Isn't exit code `1` the exit code for failure? ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
7. dawidwys commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435020917 Yes, you are right, I've used this error code to gracefully abort the test. Wasn't it the suggested idea in JIRA ticket? I wouldn't say it should happen often. I've tried running the test with resources limited for docker containers and it generally passed. I don't know which is the better, I can see both pros & cons of both approaches. ---------------------------------------------------------------- This is an

automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

8. aljoscha commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435022418 Ah, you're right! It was even my who said that Till proposed this. If @tillrohrmann is alright with this approach I'd say this is good to merge. ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
9. tillrohrmann commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435340617 Would it work to retry the deployment logic and fail if we could not deploy the cluster after several attempts? ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
10. dawidwys commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435393642 At first I thought it wouldn't change much (as the cluster starting consists only of bringing up docker containers), but actually there is some kerberos setting up logic, so it might actually make sense. Just to be clear we should fail the test meaning (exit 1). ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
11. tillrohrmann commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435398287 That sounds promising. Let us know once you've updated the PR. ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
12. dawidwys commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435802842 @tillrohrmann @aljoscha I've updated the test. Could you have final look? ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
13. dawidwys edited a comment on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435802842 @tillrohrmann @aljoscha I've updated the test. Could you have the final look? ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
14. aljoscha commented on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435811883 I think this looks good to merge now. :👌 ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
15. aljoscha edited a comment on issue #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965#issuecomment-435811883 I think this looks good to merge now. 👌 ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
16. dawidwys closed pull request #6965: [FLINK-10368][e2e] Hardened kerberized yarn e2e test URL: https://github.com/apache/flink/pull/6965 This is a PR merged from a forked repository. As GitHub hides the original diff on merge, it is displayed below for the sake of provenance: As this is a foreign pull request (from a fork), the diff is supplied below (as it won't show otherwise due to GitHub magic): diff --git a/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/bootstrap.sh b/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/bootstrap.sh index 7b5e50ba439..5b98b96e51d 100755 --- a/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/bootstrap.sh +++ b/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/bootstrap.sh @@ -124,6 +124,7 @@ elif [ "$1" == "master" ]; then hdfs dfs -chown hadoop-user:hadoop-user /user/hadoop-user kdestroy + echo "Finished master initialization" while true; do sleep 1000; done elif [ "$1" == "worker" ]; then diff --git a/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/config/yarn-site.xml b/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/config/yarn-site.xml index 9b17acc1656..c7736a69489 100644 --- a/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/config/yarn-site.xml +++ b/flink-end-to-end-tests/test-scripts/docker-hadoop-secure-cluster/config/yarn-site.xml @@ -21,6 +21,11 @@ under the License. <value>mapreduce_shuffle</value> </property> + <property> + <name>yarn.nodemanager.vmem-pmem-ratio</name> + <value>3</value> + </property> + <!-- this is ignored by the default scheduler but we have to set it because Flink would complain if we didn't have it --> <property> @@ -33,12 +38,12 @@ under the License. are scheduled on one NM, which wouldn't provoke a previously fixed Kerberos keytab bug. --> <property> <name>yarn.nodemanager.resource.memory-mb</name> - <value>4100</value> + <value>2500</value> </property> <property> <name>yarn.scheduler.minimum-allocation-mb</name> - <value>2000</value> + <value>1000</value> </property> <property> diff --git a/flink-end-to-end-tests/test-scripts/test_yarn_kerberos_docker.sh b/flink-end-to-end-tests/test-scripts/test_yarn_kerberos_docker.sh index c9ef15d3dc5..5f2dea2ea6a 100755 --- a/flink-end-to-end-tests/test-scripts/test_yarn_kerberos_docker.sh +++ b/flink-end-to-end-tests/test-scripts/test_yarn_kerberos_docker.sh @@ -24,7 +24,8 @@ FLINK_TARBALL_DIR=$TEST_DATA_DIR FLINK_TARBALL=flink.tar.gz FLINK_DIRNAME=$(basename $FLINK_DIR) -MAX_RETRY_SECONDS=800 +MAX_RETRY_SECONDS=120 +CLUSTER_SETUP_RETRIES=3 echo "Flink Tarball directory $FLINK_TARBALL_DIR" echo "Flink tarball filename $FLINK_TARBALL" @@ -33,20 +34,6 @@ echo "End-to-end directory $END_TO_END_DIR" docker --version docker-compose --version -mkdir -p $FLINK_TARBALL_DIR -tar czf $FLINK_TARBALL_DIR/$FLINK_TARBALL -C $(dirname $FLINK_DIR) . - -echo "Building Hadoop Docker container" -until docker build --build-arg HADOOP_VERSION=2.8.4 -f $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/Dockerfile -t flink/docker-hadoop-secure-cluster:latest $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/; do - # with all the downloading and ubuntu updating a lot of flakiness can happen, make sure - # we don't immediately fail - echo "Something went wrong while building the Docker image, retrying ..." - sleep 2 -done - -echo "Starting Hadoop cluster" -docker-compose -f $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/docker-compose.yml up -d - # make sure we stop our cluster at the end function cluster_shutdown { # don't call ourselves again for another signal interruption @@ -60,12 +47,71 @@ function cluster_shutdown { trap cluster_shutdown INT trap cluster_shutdown EXIT -until docker cp $FLINK_TARBALL_DIR/$FLINK_TARBALL master:/home/hadoop-user/; do - # we're retrying this one because we don't know yet if the container is ready - echo "Uploading Flink tarball to docker master failed, retrying ..." - sleep 5 +function start_hadoop_cluster() { + echo "Starting Hadoop cluster" + docker-compose -f $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/docker-compose.yml up -d + + # wait for kerberos to be set up + start_time=$(date +%s) + until docker logs master 2>&1 | grep -q "Finished master initialization"; do + current_time=$(date +%s) + time_diff=$((current_time - start_time)) + + if [ $time_diff -ge $MAX_RETRY_SECONDS ]; then + return 1 + else + echo "Waiting for hadoop cluster to come up. We have been trying for $time_diff seconds, retrying ..." + sleep 10 + fi + done + + # perform health checks + if ! { [ $(docker inspect -f '{{.State.Running}}' master 2>&1) = 'true' ] && + [ $(docker inspect -f '{{.State.Running}}' slave1 2>&1) = 'true' ] && + [ $(docker inspect -f '{{.State.Running}}' slave2 2>&1) = 'true' ] && + [ $(docker inspect -f '{{.State.Running}}' kdc 2>&1) = 'true' ]; }; + then + return 1 + fi + + return 0 +} + +mkdir -p $FLINK_TARBALL_DIR +tar czf $FLINK_TARBALL_DIR/$FLINK_TARBALL -C $(dirname $FLINK_DIR) . + +echo "Building Hadoop Docker container" +until docker build --build-arg HADOOP_VERSION=2.8.4 \ + -f $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/Dockerfile \ + -t flink/docker-hadoop-secure-cluster:latest \ + $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/; +do + # with all the downloading and ubuntu updating a lot of flakiness can happen, make sure + # we don't immediately fail + echo "Something went wrong while building the Docker image, retrying ..." + sleep 2 +done + +CLUSTER_STARTED=1 +for (( i = 0; i < $CLUSTER_SETUP_RETRIES; i++ )) +do + if start_hadoop_cluster; then + echo "Cluster started successfully." + CLUSTER_STARTED=0 + break #continue test, cluster set up succeeded + fi + + echo "ERROR: Could not start hadoop cluster. Retrying..." + docker-compose -f $END_TO_END_DIR/test-scripts/docker-hadoop-secure-cluster/docker-compose.yml down done +if [[ ${CLUSTER_STARTED} -ne 0 ]]; then + echo "ERROR: Could not start hadoop cluster. Aborting..." + exit 1 +fi + +docker cp $FLINK_TARBALL_DIR/$FLINK_TARBALL master:/home/hadoop-user/ + # now, at least the container is ready docker exec -it master bash -c "tar xzf /home/hadoop-user/$FLINK_TARBALL --directory /home/hadoop-user/" @@ -73,6 +119,7 @@ docker exec -it master bash -c "tar xzf /home/hadoop-user/$FLINK_TARBALL --direc docker exec -it master bash -c "echo \"security.kerberos.login.keytab: /home/hadoop-user/hadoop-user.keytab\" > /home/hadoop-user/$FLINK_DIRNAME/conf/flink-conf.yaml" docker exec -it master bash -c "echo \"security.kerberos.login.principal: hadoop-user\" >> /home/hadoop-user/$FLINK_DIRNAME/conf/flink-conf.yaml" docker exec -it master bash -c "echo \"slot.request.timeout: 60000\" >> /home/hadoop-user/$FLINK_DIRNAME/conf/flink-conf.yaml" +docker exec -it master bash -c "echo \"containerized.heap-cutoff-min: 100\" >> /home/hadoop-

user/$FLINK_DIRNAME/conf/flink-conf.yaml" echo "Flink config:" docker exec -it master bash -c "cat /home/hadoop-user/$FLINK_DIRNAME/conf/flink-conf.yaml" @@ -84,33 +131,28 @@ OUTPUT_PATH=hdfs:///user/hadoop-user/wc-out-$RANDOM start_time=$(date +%s) # it's important to run this with higher parallelism, otherwise we might risk that # JM and TM are on the same YARN node and that we therefore don't test the keytab shipping -until docker exec -it master bash -c "export HADOOP_CLASSPATH=\`hadoop classpath\` && /home/hadoop-user/$FLINK_DIRNAME/bin/flink run -m yarn-cluster -yn 3 -ys 1 -ytm 2000 -yjm 2000 -p 3 /home/hadoop-user/$FLINK_DIRNAME/examples/streaming/WordCount.jar --output $OUTPUT_PATH"; do - current_time=$(date +%s) - time_diff=$((current_time - start_time)) - - if [ $time_diff -ge $MAX_RETRY_SECONDS ]; then - echo "We tried running the job for $time_diff seconds, max is $MAX_RETRY_SECONDS seconds, aborting" - mkdir -p $TEST_DATA_DIR/logs - echo "Hadoop logs:" - docker cp master:/var/log/hadoop/* $TEST_DATA_DIR/logs/ - for f in $TEST_DATA_DIR/logs/*; do - echo "$f:" - cat $f - done - echo "Docker logs:" - docker logs master - exit 1 - else - echo "Running the Flink job failed, might be that the cluster is not ready yet. We have been trying for $time_diff seconds, retrying ..." - sleep 5 - fi -done - -docker exec -it master bash -c "kinit -kt /home/hadoop-user/hadoop-user.keytab hadoop-user" -docker exec -it master bash -c "hdfs dfs -ls $OUTPUT_PATH" -OUTPUT=$(docker exec -it master bash -c "hdfs dfs -cat $OUTPUT_PATH/*") -docker exec -it master bash -c "kdestroy" -echo "$OUTPUT" +if docker exec -it master bash -c "export HADOOP_CLASSPATH=\`hadoop classpath\` && \ + /home/hadoop-user/$FLINK_DIRNAME/bin/flink run -m yarn-cluster -yn 3 -ys 1 -ytm 1000 -yjm 1000 \ + -p 3 /home/hadoop-user/$FLINK_DIRNAME/examples/streaming/WordCount.jar --output $OUTPUT_PATH"; +then + docker exec -it master bash -c "kinit -kt /home/hadoop-user/hadoop-user.keytab hadoop-user" + docker exec -it master bash -c "hdfs dfs -ls $OUTPUT_PATH" + OUTPUT=$(docker exec -it master bash -c "hdfs dfs -cat $OUTPUT_PATH/*") + docker exec -it master bash -c "kdestroy" + echo "$OUTPUT" +else + echo "Running the job failed." + mkdir -p $TEST_DATA_DIR/logs + echo "Hadoop logs:" + docker cp master:/var/log/hadoop/* $TEST_DATA_DIR/logs/ + for f in $TEST_DATA_DIR/logs/*; do + echo "$f:" + cat $f + done + echo "Docker logs:" + docker logs master + exit 1 +fi if [[ ! "$OUTPUT" =~ "consummation,1" ]]; then echo "Output does not contain (consummation, 1) as required" @@ -139,7 +181,10 @@ fi echo "Running Job without configured keytab, the exception you see below is expected" docker exec -it master bash -c "echo \"\" > /home/hadoop-user/$FLINK_DIRNAME/conf/flink-conf.yaml" # verify that it doesn't work if we don't configure a keytab -OUTPUT=$(docker exec -it master bash -c "export HADOOP_CLASSPATH=\`hadoop classpath\` && /home/hadoop-user/$FLINK_DIRNAME/bin/flink run -m yarn-cluster -yn 3 -ys 1 -ytm 1200 -yjm 800 -p 3 /home/hadoop-user/$FLINK_DIRNAME/examples/streaming/WordCount.jar --output $OUTPUT_PATH") +OUTPUT=$(docker exec -it master bash -c "export HADOOP_CLASSPATH=\`hadoop classpath\` && \ + /home/hadoop-user/$FLINK_DIRNAME/bin/flink run \ + -m yarn-cluster -yn 3 -ys 1 -ytm 1000 -yjm 1000 -p 3 \ + /home/hadoop-user/$FLINK_DIRNAME/examples/streaming/WordCount.jar --output $OUTPUT_PATH") echo "$OUTPUT" if [[ ! "$OUTPUT" =~ "Hadoop security with Kerberos is enabled but the login user does not have Kerberos credentials" ]]; then ---------------------------------------------------------------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

17. Fixed in master via: 413a77157caf25dbbfb8b0caaf2c9e12c7374d98 Fixed in 1.7 via: 08cd6ea7cd8afa8d2761dde521eb9a7bf21ec5e6 Fixed in 1.6.3 via: ddcdfa5b8e89a7fb9bfe065bae376ff8571abf85 Fixed in 1.5.6 via: 42a84c7ea96e95ec9bfdf5fceb41ad841f44ba80
18. The test seems still to be unstable. It failed running it on an AWS instance: {code} Successfully built 48a8281421be Starting Hadoop cluster Creating network "docker-hadoop-cluster-network" with the default driver Creating kdc ... done Creating master ... done Creating slave2 ... done Creating slave1 ... done Waiting for hadoop cluster to come up. We have been trying for 0 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 10 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 20 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 30 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 41 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 51 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 61 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 71 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 81 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 91 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 101 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 111 seconds, retrying ... ERROR: Could not start hadoop cluster. Retrying... Stopping slave1 ... done Stopping slave2 ... done Stopping master ... done Stopping kdc ... done Removing slave1 ... done Removing slave2 ... done Removing master ... done Removing kdc ... done Removing network docker-hadoop-cluster-network Starting Hadoop cluster Creating network "docker-hadoop-cluster-network" with the default driver Creating kdc ... done Creating master ... done Creating slave2 ... done Creating slave1 ... done Waiting for hadoop cluster to come up. We have been trying for 1 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 11 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 21 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 31 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 41 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 51 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 61 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 71 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 81 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 91 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 101 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 111 seconds, retrying ... ERROR: Could not start hadoop cluster. Retrying... Stopping slave1 ... done Stopping slave2 ... done Stopping master ... done Stopping kdc ... done Removing slave1 ... done Removing slave2 ... done Removing master ... done Removing kdc ... done Removing network docker-hadoop-cluster-network Starting Hadoop cluster Creating network "docker-hadoop-cluster-network" with the default driver Creating kdc ... done Creating master ... done Creating slave2 ... done Creating slave1 ... done Waiting for hadoop cluster to come up. We have been trying for 0 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 10 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 20 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 30 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 41 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 51 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 61 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 71 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 81 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 91 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 101 seconds, retrying ... Waiting for hadoop cluster to come up. We have been trying for 111 seconds, retrying ... ERROR: Could not start hadoop cluster. Retrying... Stopping slave1 ... done Stopping slave2 ... done Stopping master ... done Stopping kdc ... done Removing slave1 ... done Removing slave2 ... done Removing master ... done Removing kdc ... done Removing network docker-hadoop-cluster-network ERROR: Could not start hadoop cluster. Aborting... Removing network docker-hadoop-cluster-network WARNING: Network docker-hadoop-cluster-network not found. [FAIL] Test script contains errors. Checking for errors... No errors in log files. Checking for exceptions... No exceptions in log files. Checking for non-empty .out files... grep: /home/admin/flink-1.7.0/log/*.out: No such file or directory No non-empty .out files. [FAIL] 'Running Kerberized YARN on Docker test ' failed after 15 minutes and 33 seconds! Test exited with exit code 1 {code}
19. From [~tzulitai]: lately the tests are failing for a different reason. See https://api.travis-ci.org/v3/job/564925127/log.txt, where we have: {code} Caused by: org.apache.flink.runtime.jobmanager.scheduler.NoResourceAvailableException: Could not allocate all requires slots within timeout of 60000 ms. Slots required: 7, slots allocated: 3, previous allocation IDs: [], execution status: completed: Attempt #0 (Source: Collection Source (1/1)) @ org.apache.flink.runtime.jobmaster.slotpool.SingleLogicalSlot@41102513 - [SCHEDULED], completed exceptionally: java.util.concurrent.CompletionException: java.util.concurrent.CompletionException: java.util.concurrent.TimeoutException/java.util.concurrent.CompletableFuture@3948e0b9[Completed exceptionally], completed: Attempt #0 (Flat Map (2/3)) @ org.apache.flink.runtime.jobmaster.slotpool.SingleLogicalSlot@4c66bf85 - [SCHEDULED], completed: Attempt #0 (Flat Map (3/3)) @ org.apache.flink.runtime.jobmaster.slotpool.SingleLogicalSlot@4e0d1ec1 - [SCHEDULED], incomplete: java.util.concurrent.CompletableFuture@e7592ae[Not completed, 1 dependents], incomplete: java.util.concurrent.CompletableFuture@70c6f6f1[Not completed, 1 dependents], incomplete: java.util.concurrent.CompletableFuture@925e5fb[Not completed, 1 dependents] {code}
20. Resolved on master in 41e9f7831ae0eb5924cb21a4a2a7a64e9cb16bc6 515325694618d5b2dc8728bf40261f73d41512d7
21. Resolved on release-1.9 in 2544a04889535244f2e7153732bc7151b0fcd070 a01d2421336ee950beee7d8bfdc614c0137a6a24
22. Resolved on release-1.8 in 94415058a3e71ff53b7d3985fa038fa1c4e4aefa 954f3c0fb33185c34fca485ccf47a2d0de587d72