

MongoDB "Connection Timed Out" Error on CCO

Contents

[Introduction](#)

[Problem](#)

[Solution](#)

Introduction

This document describes how to troubleshoot if you end up with "connection timed out" error on CloudCenter Orchestrator (CCO).

Problem

After you configure the MongoDB for CCO, it is likely to fail when it is not able to connect properly with MongoDB. This issue can arise due to many factors, however, mentioned scenario is with the ports that are blocked by IPTABLES.

To diagnose the issue, it is always advisable to look the log files and check the actual cause of the error. Here the error is caused due to connection timeout.

```
2017-05-25 17:35:53,340 ERROR context.ContextLoader [localhost-startStop-1] - Context
initialization failed
com.mongodb.MongoTimeoutException: Timed out after 30000 ms while waiting for a server that
matches ReadPreferenceServerSelector{readPreference=primary}. Client view of cluster state is
{type=UNKNOWN, servers=[{address=localhost:27017, type=UNKNOWN, state=CONNECTING,
exception={com.mongodb.MongoSocketOpenException: Exception opening socket}, caused by
{java.net.ConnectException: Connection refused (Connection refused)}}]
    at com.mongodb.connection.BaseCluster.createTimeoutException(BaseCluster.java:369)
    at com.mongodb.connection.BaseCluster.selectServer(BaseCluster.java:101)
    at
com.mongodb.binding.ClusterBinding$ClusterBindingConnectionSource.<init>(ClusterBinding.java:75)
    at
com.mongodb.binding.ClusterBinding$ClusterBindingConnectionSource.<init>(ClusterBinding.java:71)
    at com.mongodb.binding.ClusterBinding.getReadConnectionSource(ClusterBinding.java:63)
    at com.mongodb.operation.OperationHelper.withConnection(OperationHelper.java:210)
    at com.mongodb.operation.FindOperation.execute(FindOperation.java:480)
    at com.mongodb.operation.FindOperation.execute(FindOperation.java:77)
    at com.mongodb.Mongo.execute(Mongo.java:773)
    at com.mongodb.Mongo$2.execute(Mongo.java:760)
    at com.mongodb.DBCursor.initializeCursor(DBCursor.java:851)
    at com.mongodb.DBCursor.hasNext(DBCursor.java:152)
    at com.hazelcast.spring.mongodb.MongoMapStore.loadAllKeys(MongoMapStore.java:142)
    at com.hazelcast.spring.mongodb.MongoMapStore.loadAllKeys(MongoMapStore.java:47)
    at com.hazelcast.map.impl.MapStoreWrapper.loadAllKeys(MapStoreWrapper.java:119)
    at
com.hazelcast.map.impl.mapstore.BasicMapStoreContext.loadAllKeys(BasicMapStoreContext.java:179)
    at com.hazelcast.map.impl.MapKeyLoader.sendKeysInBatches(MapKeyLoader.java:249)
    at com.hazelcast.map.impl.MapKeyLoader.access$200(MapKeyLoader.java:65)
    at com.hazelcast.map.impl.MapKeyLoader$1.call(MapKeyLoader.java:152)
    at com.hazelcast.map.impl.MapKeyLoader$1.call(MapKeyLoader.java:149)
```

```
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at com.hazelcast.util.executor.CompletableFutureTask.run(CompletableFutureTask.java:57)
at
com.hazelcast.util.executor.CachedExecutorServiceDelegate$Worker.run(CachedExecutorServiceDelega
te.java:209)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:748)
at
com.hazelcast.util.executor.HazelcastManagedThread.executeRun(HazelcastManagedThread.java:76)
at com.hazelcast.util.executor.HazelcastManagedThread.run(HazelcastManagedThread.java:92)
```

Application Version

CloudCenter 4.6.x / 4.7.x / 4.8.0

Solution

Add an exception for MongoDB in the IPTABLES and execute this command:

```
# iptables -A INPUT -ptcp --dport 27017 -j ACCEPT
```

Then restart the MongoDB service and restart tomcat service:

```
# /etc/init.d/mongod restart
# /etc/init.d/tomcat restart
```