

ConnectionRefused

Connection Refused

You get a [ConnectionRefused](#) Exception when there is a machine at the address specified, but there is no program listening on the specific TCP port the client is using -and there is no firewall in the way silently dropping TCP connection requests. If you do not know what a TCP connection request is, please consult the [specification](#).

Unless there is a configuration error at either end, a common cause for this is the Hadoop service isn't running.

This stack trace is very common when the cluster is being shut down -because at that point Hadoop services are being torn down across the cluster, which is visible to those services and applications which haven't been shut down themselves. Seeing this error message during cluster shutdown is not anything to worry about.

If the application or cluster is not working, and this message appears in the log, then it is more serious.

The exception text declares both the hostname and the port to which the connection failed. The port can be used to identify the service. For example, port 9000 is the HDFS port. Consult the [Ambari port reference](#), and/or those of the supplier of your Hadoop management tools.

1. Check the hostname the client using is correct. If it's in a Hadoop configuration option: examine it carefully, try doing an ping by hand.
2. Check the IP address the client is trying to talk to for the hostname is correct.
3. Make sure the destination address in the exception isn't 0.0.0.0 -this means that you haven't actually configured the client with the real address for that service, and instead it is picking up the server-side property telling it to listen on every port for connections.
4. If the error message says the remote service is on "127.0.0.1" or "localhost" that means the configuration file is telling the client that the service is on the local server. If your client is trying to talk to a remote system, then your configuration is broken.
5. Check that there isn't an entry for your hostname mapped to 127.0.0.1 or 127.0.1.1 in `/etc/hosts` (Ubuntu is notorious for this).
6. Check the port the client is trying to talk to using matches that the server is offering a service on. The `netstat` command is useful there.
7. On the server, try a `telnet localhost <port>` to see if the port is open there.
8. On the client, try a `telnet <server> <port>` to see if the port is accessible remotely.
9. Try connecting to the server/port from a different machine, to see if it just the single client misbehaving.
10. If your client and the server are in different subdomains, it may be that the configuration of the service is only publishing the basic hostname, rather than the Fully Qualified Domain Name. The client in the different subdomain can be unintentionally attempt to bind to a host in the local subdomain —and failing.
11. If you are using a Hadoop-based product from a third party, -please use the support channels provided by the vendor.
12. Please do not file bug reports related to your problem, as they will be closed as [Invalid](#)

See also [Server Overflow](#)

None of these are Hadoop problems, they are hadoop, host, network and firewall configuration issues. As it is your cluster, [only you can find out and track down the problem](#).