Jenkins slave cannot connect with master: Incorrect acknowledgement sequence

Asked 6 years, 3 months ago Modified 1 year, 6 months ago Viewed 27k times

Part of CI/CD Collective

17

After update of our Jenkins master installation to its latest LTS version 2.46.3 one of its slaves (Windows 7 machine, 32-bit) cannot connect with the master.

The error we're getting is:

```
java -jar slave.jar -jnlpUrl https://<jenkins-name>/computer/<node-name>/slave-
agent.jnlp -secret <secret-value>
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up slave: node-name
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [https://<jenkins-name>/]
Jun 22, 2017 1:19:05 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver
resolve
INFO: Remoting server accepts the following protocols: [JNLP3-connect, JNLP-
connect, CLI2-connect, Ping, CLI-connect, JNLP4-connect, JNLP2-c
onnect]
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: <jenkins-name>
  Agent port:    <jenkins-port>
  Identity:      <id:en:ti:ty>
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to <jenkins-name>:9150
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Jun 22, 2017 1:19:05 PM org.jenkinsci.remoting.protocol.impl.AckFilterLayer abort
WARNING: [JNLP4-connect connection to <our-proxy>/10.253.0.11:81] Incorrect
acknowledgement sequence, expected 0x0003414333 got 0x4854545044
Jun 22, 2017 1:19:05 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Protocol JNLP4-connect encountered an unexpected exception
java.util.concurrent.ExecutionException:
org.jenkinsci.remoting.protocol.impl.ConnectionRefusalException: Connection
closed before acknowled
gement sent
        at
org.jenkinsci.remoting.util.SettableFuture.get(SettableFuture.java:223)
        at hudson.remoting.Engine.innerRun(Engine.java:385)
        at hudson.remoting.Engine.run(Engine.java:287)
```

We spent a lot of time trying to fix the problem. Unfortunately without success.

Do you have an idea what could have caused the problem and how can it be solved?

jenkins jenkins-slave

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up ✕

Share Improve this question Follow

asked Jun 22, 2017 at 11:57
andreasgk
683 1 12 30

what was the previous version of Jenkins ? what is the authentication mechanism you are using ? LDAP , internal DB or something else ? – Mor Lajb Aug 24, 2017 at 4:28

1 Hey @andreasgk, did you manage to solve this issue? We're experiencing exactly the same error. – Thomas Stubbe Jun 17, 2021 at 15:15

7 Answers

Sorted by:

Highest score (default)

5

We recently hit this issue with our AWS-based Jenkins using JNLP for remote integration testing. The remote slave would call back to the Jenkins master, which failed with a similar error. The issue ended up being a dynamically generated AWS ELB of type HTTP (because the Kubernetes ELB provisioner presently doesn't support multi-protocol ELBs) for the Jenkins Master. We had to manually change the JNLP ingress port type of the ELB to `TCP` , while the web interface ingress 'instance port' was protocol `HTTP` and 'load balancer' was protocol `HTTPS` .

Share Improve this answer Follow

edited Feb 6, 2018 at 16:26

answered Feb 6, 2018 at 16:17
mars64
171 2 7

2 The JNLP protocol is binary, not HTTP; it's not obvious because the listener has a special case to send a dummy HTTP response if you send it a GET request. You can't put an HTTP proxy in front of the JNLP listener. If you do, "Incorrect acknowledgement sequence" is the error message you happen to get. – jim-minter Mar 2, 2018 at 22:11

Hey @mars64, can you give me some help? I have the same exact situation. I'm trying to expose JNLP from my ingress controller – Bernardo Vale May 3, 2018 at 12:37

For future references. Classic loadbalancer wasn't working with proxy protocol. I've disable proxy in my ingress controller and now everything works just fine – Bernardo Vale May 3, 2018 at 13:09

@jim-minter right you are! I mentioned this because we were using the cloud-provider annotations in Kubernetes on AWS to provision the ELB, which defaults to `HTTP` -- **every time you reprovision the service**. @BernardoVale - sorry to miss your message. Sounds like you got it resolved. Good luck! – mars64 Jun 13, 2018 at 18:56

had this exact issue with an elb classic service, and the TCP manual change worked for me (thanks). I am thinking of switching it to an ALB (ingress) or NLB balancer to use the new annotations and controller rather than the now "legacy" ELB version. If anyone has success with it before I go mucking around with it, and sees this let me know how you got on :D – Adam Comerford Nov 5, 2021 at 14:48

Is the Jenkins master instance running behind a load balancer? I had the same issue when

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up ✕

5

If so, then the acknowledgement sequence could get modified because of differing protocols in the Load balancer. JNLP requires TCP connection on port 50000 by default.

If your setup is on AWS, you could try creating a private hosted zone in Route53 with an Alias record for your Jenkins instance's private IP address.

For e.g: jenkins.example.com -> your Jenkins instance's private IP

Then, in **Jenkins UI -> Manage Jenkins -> Configure System -> Manage nodes and clouds -> Configure clouds ->** (under advanced settings)

Tunnel connection through : jenkins.example.com:50000

This avoids your slave agents to have to go through the load balancer to connect to the Jenkins Master.

Share Improve this answer Follow

answered Nov 12, 2020 at 12:52

Sidharth Ramesh
646 2 6 21

nice idea, this give me a hint to resolve my problem. there are slightly difference – Han.Oliver May 25, 2021 at 7:51

does not work for me unfortunately. The jenkins server is in AWS behind the NLB (443 > Ec2 TLS; 50000 > Ec2 TCP) and jenkins agent is a local macnini machine that can not resolvel private Route53 name and can not reach the jenkins server on private ip. It must work through the Network Load Balancer, but it does not =(– Артем Косенко Nov 11, 2022 at 16:06

works, though if you have fixed agent port in global configuration then you need to set that in this option with LB IP – MoonLight Dec 7, 2022 at 10:02

5

I had this issue, and i found a solution.

I have a jenkins master deployed via the jenkins helm chart on an EKS, and its exposed with an ingress controller, which is behind an ALB. I tried to connect an inbound agent node and i had this error above.

solution in short: just tick the "Use WebSocket" option in the agent node config (Manage Jenkins ==> Manage nodes and clouds ==> choose your inbound agent node ==> Configure ==> tick the "Use WebSocket")

after i did it - the agent could connect and the error was gone. this is the most elegant solution, i believe its also more secured because when you use it you dont need to keep the 50000 tcp port open, you can just keep using the main port of jenkins (443 usually i guess),

Note: you do need to make sure that the agent will have access to the jenkins main port (usually 443 or 80)

Join Stack Overflow to find the best answer to your technical question, help others answer theirs. Sign up

i found this: https://docs.cloudbees.com/docs/cloudbees-ci/latest/cloud-setup-guide/configure-ports-jnlp-agents

which led me to this: https://github.com/jenkinsci/jep/blob/master/jep/222/README.adoc

they explain there that when you expose jenkins through a load balancer, then you better use the websocket option (and even if not, using websocket is still better, because the websocket is more secured then the jnlp and the extra tcp port)

Share Improve this answer Follow

edited Mar 23, 2022 at 14:59

answered Mar 22, 2022 at 18:10
Ofir Yahav
81 1 6

awesome, "Use WebSocet" resolved for me also. thanks for posting this else its difficult do debug. i think any setup which advanced networking configuration, cloud or hybrid infra would be benefitted with this.. kudos man – nailed IT Aug 29 at 13:02

1

I encounter this kind of problem on gcp, jenkins master behind load balance, almost the same as Sidharth Ramesh reply.

in configuration -> manage jenkins -> configure global security, in the 'agent' part, you must config a specific port, never choose random. I choose 50222 as example,

below is agent protocols: there is a checkbox of "Inbound TCP Agent Protocol/4 (TLS encryption)", we must make enable. if not, there is an error message: "server reports protocol jnlp4-connect not supported skipping"

open the firewall of port from jenkins slave to jenkins master vm internal ip.

enjoy

Share Improve this answer Follow

answered May 25, 2021 at 7:50
Han.Oliver
525 5 8

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up

0

You need to check the secret key of the node is intact. If not proper, you have to download the slave.jar and also Run agent from command line with new jar file.

```
java -jar slave.jar -jnlpUrl
http://<ipaddress>:8080/computer/<computername>/slave-agent.jnlp -secret
340d54sdrgtjj334kelkahsdjkf83f1c5120dc2fb74939fcdb7f05e1926049f8d7991
```

Also to check the java version installed is > 7

Share Improve this answer Follow

edited Aug 29, 2017 at 13:01
Machavity ♦
30.9k 27 92 100

answered Aug 29, 2017 at 12:58
user8532358

0

This happened to us when a Windows Update or some other silent background update messed with the slave's environment variables. `HTTPS_PROXY` and `HTTP_PROXY` had to be re-added, and once that was done we were back in business.

Share Improve this answer Follow

answered Feb 21, 2019 at 22:59
kayleeFrye_onDeck
6,678 5 69 80

0

The message:

```
Incorrect acknowledgement sequence ...
```

happened for me when I had incorrectly configured a value for the Java property `hudson.TcpSlaveAgentListener.port` as the same port number as the HTTP port used by Jenkins. The [TcpSlaveAgentListener javadoc] indicates that is a misconfiguration when it says:

> **Aside from the HTTP endpoint**, Jenkins runs TcpSlaveAgentListener that listens on a TCP socket. Historically this was used for inbound connection from agents (hence the name), but over time it was extended and made generic, so that multiple protocols of different purposes can co-exist on the same socket. (emphasis added)

If the HTTP port was 8080 and the `hudson.TcpSlaveAgentListener.port` was also 8080, then my JNLP agents failed to connect. As soon as I assigned another value to `hudson.TcpSlaveAgentListener.port` (like 50000) and restarted Jenkins, my JNLP agents were able to connect.

Join Stack Overflow to find the best answer to your technical question, help others answer theirs. Sign up ✕

```
INFO: Trying protocol: JNLP4-connect
Mar 02, 2019 3:49:29 PM org.jenkinsci.remoting.protocol.impl.AckFilterLayer abort
WARNING: [JNLP4-connect connection to agent.example.com/172.16.16.113:8080]
Incorrect acknowledgement sequence, expected 0x000341434b got 0x485454502f
Mar 02, 2019 3:49:29 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Protocol JNLP4-connect encountered an unexpected exception
java.util.concurrent.ExecutionException:
org.jenkinsci.remoting.protocol.impl.ConnectionRefusalException: Connection
closed before acknowledgement sent
        at
org.jenkinsci.remoting.util.SettableFuture.get(SettableFuture.java:223)
        at hudson.remoting.Engine.innerRun(Engine.java:614)
        at hudson.remoting.Engine.run(Engine.java:474)
Caused by: org.jenkinsci.remoting.protocol.impl.ConnectionRefusalException:
Connection closed before acknowledgement sent
        at
org.jenkinsci.remoting.protocol.impl.AckFilterLayer.onRecvClosed(AckFilterLayer.java
        at
org.jenkinsci.remoting.protocol.FilterLayer.abort(FilterLayer.java:164)
        at
org.jenkinsci.remoting.protocol.impl.AckFilterLayer.abort(AckFilterLayer.java:130)
        at
org.jenkinsci.remoting.protocol.impl.AckFilterLayer.onRecv(AckFilterLayer.java:258)
        at
org.jenkinsci.remoting.protocol.ProtocolStack$Ptr.onRecv(ProtocolStack.java:668)
        at
org.jenkinsci.remoting.protocol.NetworkLayer.onRead(NetworkLayer.java:136)
        at
org.jenkinsci.remoting.protocol.impl.BIONetworkLayer.access$2200(BIONetworkLayer.ja
        at
org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader.run(BIONetworkLayer.java
        at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
        at hudson.remoting.Engine$1.lambda$newThread$0(Engine.java:93)
        at java.lang.Thread.run(Unknown Source)

Mar 02, 2019 3:49:29 PM hudson.remoting.jnlp.Main$CuiListener status
```

Share Improve this answer Follow

answered Mar 2, 2019 at 23:02

Mark Waite

1,351 11 13

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up ✕