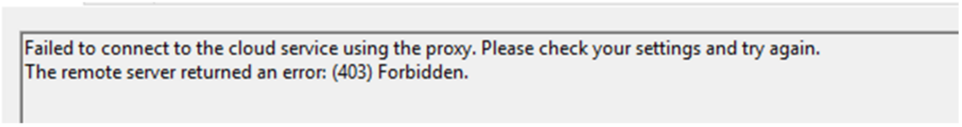
Self-Hosted node failed to connect to Frond-End Servers

Last updated by | Veena Pachauri | Mar 8, 2023 at 11:10 PM PST

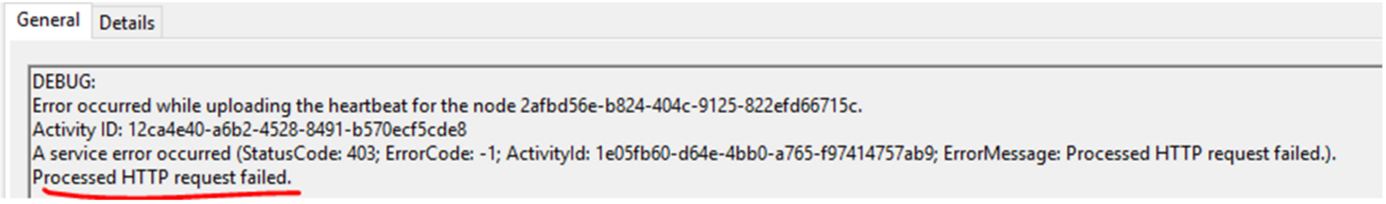
Contents

Issue

Self-IR failed with error:



Failed to connect to the cloud service using the proxy. Please check your settings and try again.
The remote server returned an error: (403) Forbidden.



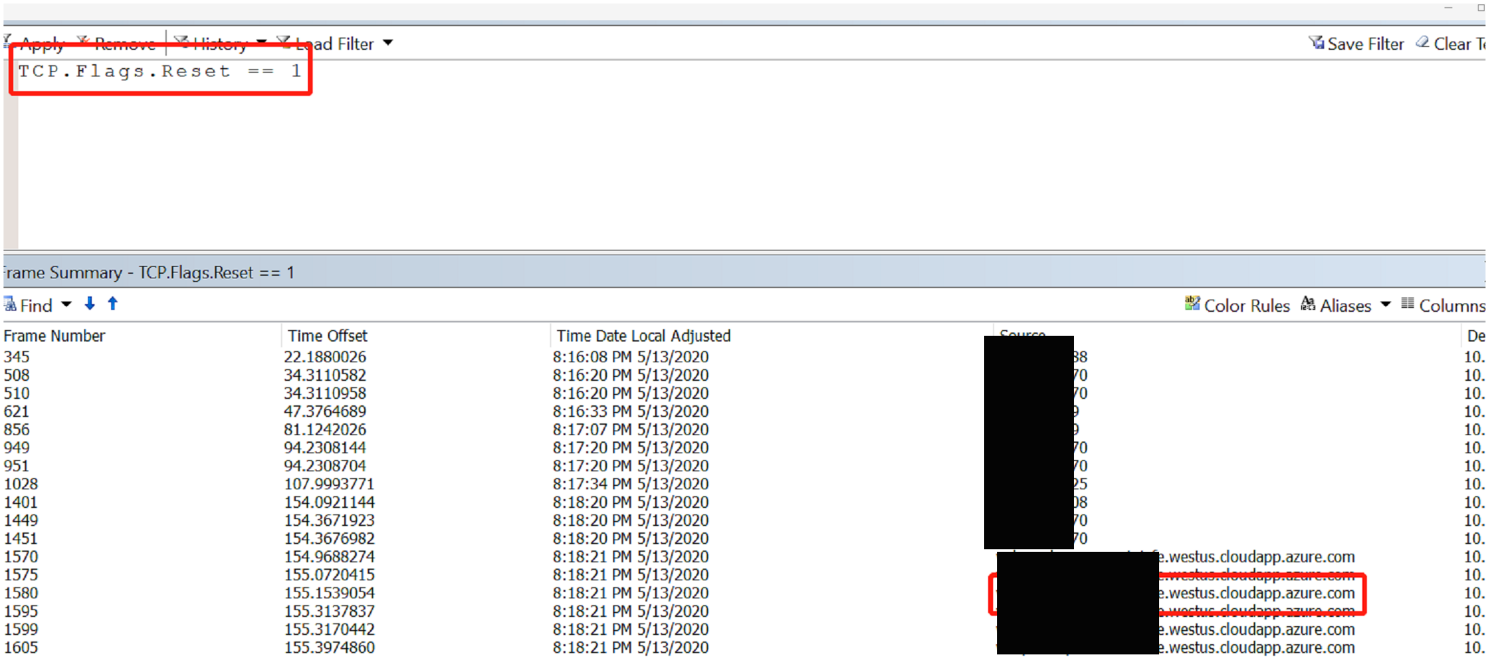
```
DEBUG:
Error occurred while uploading the heartbeat for the node 2afbd56e-b824-404c-9125-822efd66715c.
Activity ID: 12ca4e40-a6b2-4528-8491-b570ecf5cde8
A service error occurred (StatusCode: 403; ErrorCode: -1; ActivityId: 1e05fb60-d64e-4bb0-a765-f97414757ab9; ErrorMessage: Processed HTTP request failed.).
Processed HTTP request failed.
```

It can apply to any connector issues from self-IR about connectivity issues as well.

Troubleshoot

Took the netmon trace and analyzed further.

Firstly, you can set the filter to see any reset there from the server to the client side. From this example, you can see the server side is fe server.



```
TCP.Flags.Reset == 1
```

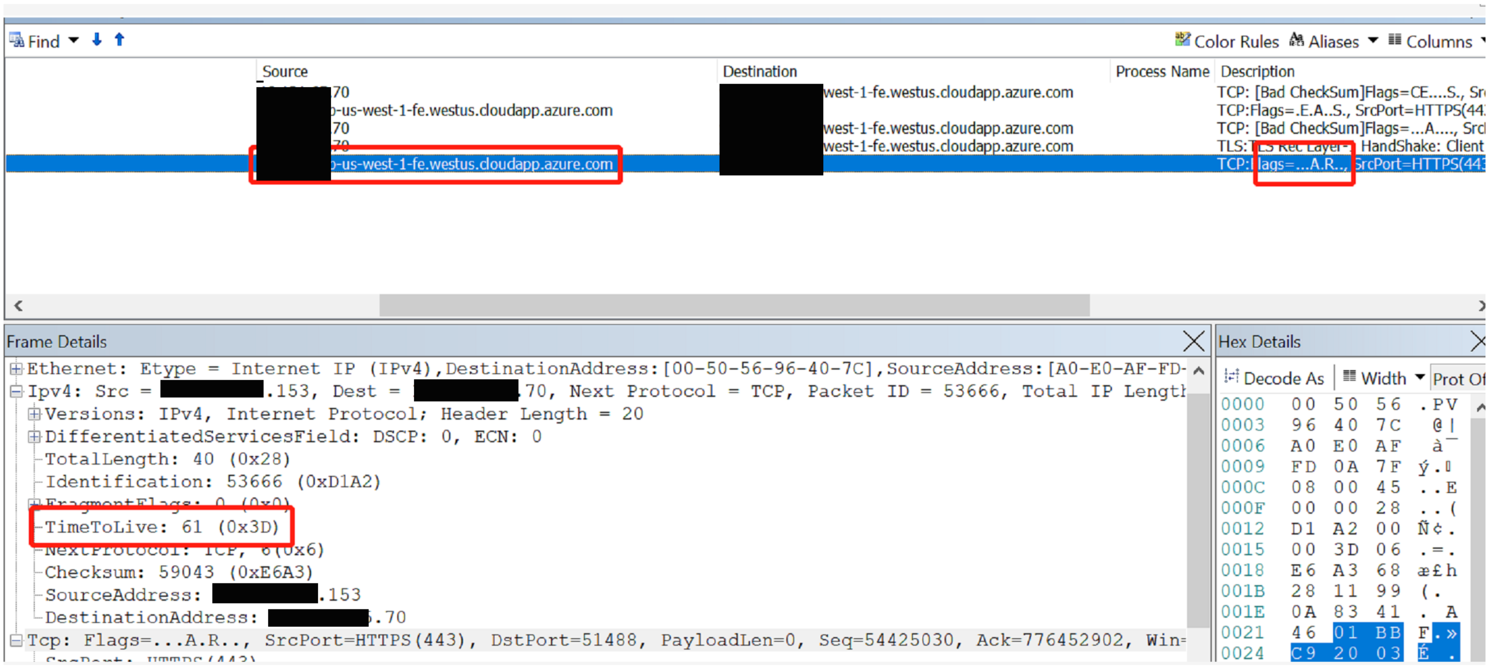
rame Summary - TCP.Flags.Reset == 1

Frame Number	Time Offset	Time Date Local Adjusted	Source	De
345	22.1880026	8:16:08 PM 5/13/2020	[redacted]	10.
508	34.3110582	8:16:20 PM 5/13/2020	[redacted]	10.
510	34.3110958	8:16:20 PM 5/13/2020	[redacted]	10.
621	47.3764689	8:16:33 PM 5/13/2020	[redacted]	10.
856	81.1242026	8:17:07 PM 5/13/2020	[redacted]	10.
949	94.2308144	8:17:20 PM 5/13/2020	[redacted]	10.
951	94.2308704	8:17:20 PM 5/13/2020	[redacted]	10.
1028	107.9993771	8:17:34 PM 5/13/2020	[redacted]	10.
1401	154.0921144	8:18:20 PM 5/13/2020	[redacted]	10.
1449	154.3671923	8:18:20 PM 5/13/2020	[redacted]	10.
1451	154.3676982	8:18:20 PM 5/13/2020	[redacted]	10.
1570	154.9688274	8:18:21 PM 5/13/2020	[redacted].westus.cloudapp.azure.com	10.
1575	155.0720415	8:18:21 PM 5/13/2020	[redacted]~~.westus.cloudapp.azure.com~~	10.
1580	155.1539054	8:18:21 PM 5/13/2020	[redacted].westus.cloudapp.azure.com	10.
1595	155.3137837	8:18:21 PM 5/13/2020	[redacted]~~.westus.cloudapp.azure.com~~	10.
1599	155.3170442	8:18:21 PM 5/13/2020	[redacted].westus.cloudapp.azure.com	10.
1605	155.3974860	8:18:21 PM 5/13/2020	[redacted].westus.cloudapp.azure.com	10.

Based the the netmon trace collected, we can see the TTL total is 64, according to https://packetpushers.net/ip-time-to-live-and-hop-limit-basics/ with info below, it most likely to be the Linux System reset that package and caused the disconnection.

Default TTL and Hop Limit Values

```
Default TTL and Hop Limit values vary between different operating systems, here are the defaults for a few:
•       Linux kernel 2.4 (circa 2001): 255 for TCP, UDP and ICMP
•       Linux kernel 4.10 (2015): 64 for TCP, UDP and ICMP
•       Windows XP (2001): 128 for TCP, UDP and ICMP
•       Windows 10 (2015): 128 for TCP, UDP and ICMP
•       Windows Server 2008: 128 for TCP, UDP and ICMP
•       Windows Server 2019 (2018): 128 for TCP, UDP and ICMP
•       MacOS (2001): 64 for TCP, UDP and ICMP
```



```
Source                                        Destination                                    Process Name  Description
[redacted]70                                  [redacted]west-1-fe.westus.cloudapp.azure.com                TCP: [Bad CheckSum]Flags=CE....S., Sr
[redacted]o-us-west-1-fe.westus.cloudapp.azure.com                                                         TCP:Flags=.E.A..S., SrcPort=HTTPS(44
[redacted]70                                  west-1-fe.westus.cloudapp.azure.com                          TCP: [Bad CheckSum]Flags=...A...., Src
[redacted]70                                  west-1-fe.westus.cloudapp.azure.com                          TLS:TLS Rec Layer-1 HandShake: Client
[redacted]o-us-west-1-fe.westus.cloudapp.azure.com                                                         TCP:Flags=...A.R.., SrcPort=HTTPS(44
```

Frame Details

```
⊞Ethernet: Etype = Internet IP (IPv4),DestinationAddress:[00-50-56-96-40-7C],SourceAddress:[A0-E0-AF-FD-
⊟Ipv4: Src = [redacted].153, Dest = [redacted].70, Next Protocol = TCP, Packet ID = 53666, Total IP Length
  ⊞Versions: IPv4, Internet Protocol; Header Length = 20
  ⊞DifferentiatedServicesField: DSCP: 0, ECN: 0
  -TotalLength: 40 (0x28)
  -Identification: 53666 (0xD1A2)
  ⊞FragmentFlags: 0 (0x0)
  -TimeToLive: 61 (0x3D)
  -NextProtocol: TCP, 6(0x6)
  -Checksum: 59043 (0xE6A3)
  -SourceAddress: [redacted].153
  -DestinationAddress: [redacted].70
⊟Tcp: Flags=...A.R.., SrcPort=HTTPS(443), DstPort=51488, PayloadLen=0, Seq=54425030, Ack=776452902, Win=
```

Hex Details

Decode As | Width | Prot Of

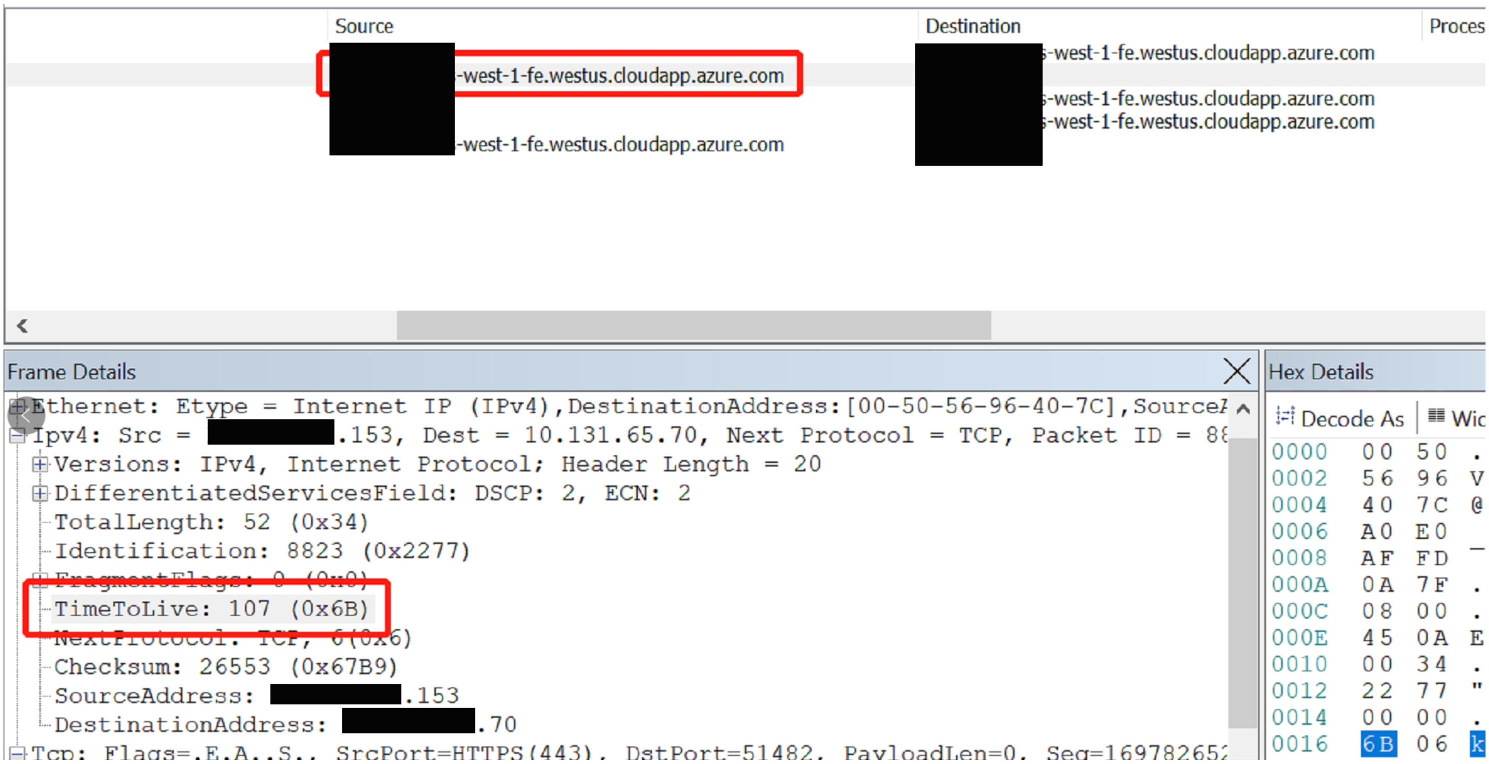
```
0000  00 50 56  .PV
0003  96 40 7C   @|
0006  A0 E0 AF   à¯
0009  FD 0A 7F  ý.
000C  08 00 45  ..E
000F  00 00 28  ..(
0012  D1 A2 00  Ñ¢.
0015  00 3D 06  .=.
0018  E6 A3 68  æ£h
001B  28 11 99  (.
001E  0A 83 41  . A
0021  46 01 BB  F.»
0024  C9 20 03  É .
```

However, why do you see 61 instead of 64? When the network package to reach to destination, it need to go through different hops such as routers/network devices, 64 would minus the number of routers/network devices until reach to final destination.

In this case, we can see Reset may sent from Linux System with TTL 64 the third hops from Self-IR.

Network package from Linux System A with TTL 64 -> B TTL 64 Minus 1 = 63 -> C TTL 63 Minus 1 = 62 -> TTL 62 Minus 1 = 61 Self-IR

In good situation, you can see TTL is 128 which means it is Windows System Running our Front-End, 128 – 107 = 21 hops, it explained to us that there are 21 hops for the package sent from FE to self-IR during TCP three handshake.



```
Source                                              Destination                                      Proces
[redacted]-west-1-fe.westus.cloudapp.azure.com      [redacted]-west-1-fe.westus.cloudapp.azure.com
                                                              -west-1-fe.westus.cloudapp.azure.com
          -west-1-fe.westus.cloudapp.azure.com                -west-1-fe.westus.cloudapp.azure.com
```

Frame Details

```
⊞Ethernet: Etype = Internet IP (IPv4),DestinationAddress:[00-50-56-96-40-7C],SourceA
⊟Ipv4: Src = [redacted].153, Dest = 10.131.65.70, Next Protocol = TCP, Packet ID = 88
  ⊞Versions: IPv4, Internet Protocol; Header Length = 20
  ⊞DifferentiatedServicesField: DSCP: 2, ECN: 2
  -TotalLength: 52 (0x34)
  -Identification: 8823 (0x2277)
  ⊞FragmentFlags: 0 (0x0)
  -TimeToLive: 107 (0x6B)
  -NextProtocol: TCP, 6(0x6)
  -Checksum: 26553 (0x67B9)
  -SourceAddress: [redacted].153
  -DestinationAddress: [redacted].70
⊟Tcp: Flags=.E.A..S., SrcPort=HTTPS(443), DstPort=51482, PayloadLen=0, Seq=16978265
```

Hex Details

Decode As | Wic

```
0000  00 50  .
0002  56 96  V
0004  40 7C  @
0006  A0 E0
0008  AF FD  ¯
000A  0A 7F  .
000C  08 00  .
000E  45 0A  E
0010  00 34  .
0012  22 77  "
0014  00 00  .
0016  6B 06  k
```

Therefore, customer need to engage your network team to see what the third hops is from self-IR, is it the firewall as linux System? If yes, then check any logs on why that device reset the package after TCP 3 handshake. However, if customer is not sure where to do investigation, they can try to get the netmon trace from Self-IR and Firewall together during the problematic time to figure out which device may reset this package and caused the disconnection.

Reference

https://icm.ad.msft.net/imp/v3/incidents/details/187018305/home
https://supportability.visualstudio.com/AzureDataFactory/_wiki/wikis/AzureDataFactory/286726/Cannot-connect-to-cloud-service

Example: 2206150030000986

How good have you found this content?

🙂 🙁