

Title: Spring Boot Admin - Too Many open Files In System Error

Post Body:

I'm trying to run spring-boot-admin on ECS Fargate - and after a few minutes the server dies and the logs are filled with 'too many open files in system' errors.

I'm using spring-boot 2.3.1, and have tried 2.2.3 and the 2.3.0-SNAPSHOT of spring-boot-admin. The jar is running on an ubuntu 20.04 base image with openjdk-11-jdk-headless installed. The ECS service has 2gb RAM available, and I've increased Ulimits on nofile and nproc (100000)

```
Ulimits:          - Name: nofile          HardLimit: 1000000          SoftLimit: 1000000          - Name: nproc
```

Stacktrace:

```
2020-06-29 22:03:35.691 ERROR 6 --- [io-8080-exec-24] o.a.c.c.C.[.][.][dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is io.netty.channel.ChannelException: io.netty.channel.unix.Errors$NativeIoException: newSocketStream(..) failed: Too many open files in system] with root cause io.netty.channel.unix.Errors$NativeIoException: newSocketStream(..) failed: Too many open files in system
2020-06-29 22:03:36.345 ERROR 6 --- [io-8080-exec-14] o.a.c.c.C.[.][.][dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is io.netty.channel.ChannelException: io.netty.channel.unix.Errors$NativeIoException: newSocketStream(..) failed: Too many open files in system] with root cause io.netty.channel.unix.Errors$NativeIoException: newSocketStream(..) failed: Too many open files in system
2020-06-29 22:03:36.350 ERROR 6 --- [o-8080-Acceptor] org.apache.tomcat.util.net.Acceptor : Socket accept failed
java.io.IOException: Too many open files in system at java.base/sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method) ~[na:na] at java.base/sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:533) ~[na:na] at java.base/sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:285) ~[na:na] at org.apache.tomcat.util.net.NioEndpoint.serverSocketAccept(NioEndpoint.java:469) ~[tomcat-embed-core-9.0.36.jar!/:9.0.36] at org.apache.tomcat.util.net.NioEndpoint.serverSocketAccept(NioEndpoint.java:71) ~[tomcat-embed-core-9.0.36.jar!/:9.0.36] at org.apache.tomcat.util.net.Acceptor.run(Acceptor.java:95) ~[tomcat-embed-core-9.0.36.jar!/:9.0.36] at java.base/java.lang.Thread.run(Thread.java:834) ~[na:na]
```

I've got a set of 8 microservices connected with the sba-client (no security at the moment) for 3 environments (24 instances in total). Only settings in the client are:

```
spring.boot.admin.client.instance.prefer-ip=true spring.boot.admin.client.url=https://xxxxx.com spring.boot.admin.client.insta
```

I've enabled prefer IP as the majority of these instances aren't behind Eureka or a load balancer, and just process data off queues.

The server only has `spring.boot.admin.ui.public-url` set.

For the first few minutes everything works fine - but then these errors start occurring and everything falls over. Cloudwatch metrics say the CPU shoots to 100%, then target-group healthchecks on sba fail and ECS restarts the task. This currently takes about 30 minutes.

Raising the ulimits from defaults has increased the time before the app falls over, but it still falls over eventually - as if it's leaking sockets / connections.

I've not had any experience running webflux / netty apps - is there something I'm missing? Do I need to set a higher ulimit?

Accepted Answer:

I was having the same problem, found out that there's an issue logged in spring boot about this: [Many File Open Issue : Spring Boot 2.3.0 -> Spring Boot 2.3.1 #21934](#)

Until a new version is out, bumping reactor-netty to 0.9.9.RELEASE should fix it, did for me!

Highest Rated Answer:

I faced same issue, unfortunately updating reactor-netty to 0.9.9.RELEASE did NOT fix the issue, but downgrading to 0.9.7.RELEASE.