



# Solved - Spring Cloud Config Server invalid SSH key on a MacOS Mojave

PUBLISHED ON 27 MAY 2019

LAST UPDATED ON 19 APR 2021

```
Run: ConfigServer x
Console Endpoints
2019-05-27 11:05:59.435 INFO 19051 --- [main] trationDelegatesBeanPostProcessorChecker : Bean 'org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfig
:: Spring Boot :: (v2.1.5.RELEASE)
2019-05-27 11:05:59.592 INFO 19051 --- [main] com.skryvets.demo.ConfigServer : No active profile set, falling back to default profiles: default
2019-05-27 11:06:00.382 INFO 19051 --- [main] o.s.cloud.context.scope.GenericScope : BeanFactory id=38d6ce87-f4c0-344c-990d-b25f64d78bca
2019-05-27 11:06:00.406 INFO 19051 --- [main] trationDelegatesBeanPostProcessorChecker : Bean 'org.springframework.cloud.autoconfigure.ConfigurationPropertiesRebinderAutoConfig
2019-05-27 11:06:00.562 INFO 19051 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2019-05-27 11:06:00.581 INFO 19051 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2019-05-27 11:06:00.581 INFO 19051 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.19]
2019-05-27 11:06:00.662 INFO 19051 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2019-05-27 11:06:00.662 INFO 19051 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1058 ms
2019-05-27 11:06:01.309 INFO 19051 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2019-05-27 11:06:01.555 INFO 19051 --- [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 2 endpoint(s) beneath base path '/actuator'
2019-05-27 11:06:01.644 INFO 19051 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2019-05-27 11:06:01.647 INFO 19051 --- [main] com.skryvets.demo.ConfigServer : Started ConfigServer in 2.836 seconds (JVM running for 3.288)
2019-05-27 11:06:01.995 INFO 19051 --- [1]-192.168.1.67] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2019-05-27 11:06:01.995 INFO 19051 --- [1]-192.168.1.67] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2019-05-27 11:06:02.001 INFO 19051 --- [1]-192.168.1.67] o.s.web.servlet.DispatcherServlet : Completed initialization in 5 ms
2019-05-27 11:06:02.187 WARN 19051 --- [2]-192.168.1.67] .c.s.e.MultipleJGitEnvironmentRepository : Error occurred cloning to base directory.

org.eclipse.jgit.api.errors.TransportException: git@github.com:skryvets/java-spring-cloud-tolls-config.git: invalid privatekey: [B@19d20305
--at org.eclipse.jgit.api.FetchCommand.call(FetchCommand.java:254) ~[org.eclipse.jgit-5.1.3.201810200350-r.jar:5.1.3.201810200350-r]
--at org.eclipse.jgit.api.CloneCommand.fetch(CloneCommand.java:306) ~[org.eclipse.jgit-5.1.3.201810200350-r.jar:5.1.3.201810200350-r]
--at org.eclipse.jgit.api.CloneCommand.call(CloneCommand.java:200) ~[org.eclipse.jgit-5.1.3.201810200350-r.jar:5.1.3.201810200350-r]
--at org.springframework.cloud.config.server.environment.JGitEnvironmentRepository.cloneToBasedir(JGitEnvironmentRepository.java:589) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.JGitEnvironmentRepository.copyRepository(JGitEnvironmentRepository.java:564) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.JGitEnvironmentRepository.createGitClient(JGitEnvironmentRepository.java:547) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.JGitEnvironmentRepository.refresh(JGitEnvironmentRepository.java:268) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.JGitEnvironmentRepository.getLocations(JGitEnvironmentRepository.java:246) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.MultipleJGitEnvironmentRepository.getLocations(MultipleJGitEnvironmentRepository.java:146) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.AbstractScmEnvironmentRepository.findOne(AbstractScmEnvironmentRepository.java:51) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.MultipleJGitEnvironmentRepository.findOne(MultipleJGitEnvironmentRepository.java:186) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.environment.CompositeEnvironmentRepository.findOne(CompositeEnvironmentRepository.java:52) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.cloud.config.server.config.ConfigServerHealthIndicator.doHealthCheck(ConfigServerHealthIndicator.java:73) [spring-cloud-config-server-2.1.1.RELEASE.jar:2.1.1.RELEASE]
--at org.springframework.boot.actuate.health.AbstractHealthIndicator.health(AbstractHealthIndicator.java:84) [spring-boot-actuator-2.1.5.RELEASE.jar:2.1.5.RELEASE]
--at org.springframework.boot.actuate.health.CompositeHealthIndicator.health(CompositeHealthIndicator.java:98) [spring-boot-actuator-2.1.5.RELEASE.jar:2.1.5.RELEASE]
--at org.springframework.boot.actuate.health.HealthEndpoint.health(HealthEndpoint.java:50) [spring-boot-actuator-2.1.5.RELEASE.jar:2.1.5.RELEASE] <4 internal calls>
```

# Introduction

More and more often it's all about configuration and some hidden "setting" which does the job...

Spring Config Server allows to fetch a configuration from the remote git repository via https:

```
spring:
  cloud:
    config:
      server:
        git:
          uri: https://github.com/my-organization/my-config-repo.git
          username: goat
          password: mystrongpa$$word
```

I'm not a security expert, but this approach has an obvious downside ("username" and "password" in code). One might argue that it's possible to inject the values via environment variables or using some other technique. Nonetheless, the main flaw is the idea of supplying "username" and "password" in the first place.

## An alternative way and the problem

An better technique is to use ssh key:

```
spring:
  cloud:
    config:
      server:
        git:
          uri: git@github.com:my-organization/my-config-repo.git
```

It's a good solution, but **doesn't work out of the box**. According to [spring docs](#) it should:

*If you do not use HTTPS and user credentials, SSH should also work out of the box when you store keys in the default directories ( `~/.ssh` ) and the URI points to an SSH location, such as `git@github.com:configuration/cloud-configuration` . It is important that an entry for the Git server be present in the `~/.ssh/known_hosts` file and that it is in `ssh-rsa` format. Other formats (such as `ecdsa-sha2-nistp256` ) are not supported. To avoid surprises, you should ensure that only one entry is present in the `known_hosts` file for the Git server and that it matches the URL you provided to the config server. If you use a hostname in the URL, you want to have exactly that (not the IP) in the `known_hosts` file. The repository is accessed by using JGit, so any documentation you find on that should be applicable. HTTPS proxy settings can be set in `~/.git/config` or (in the same way as for any other JVM process) with system properties ( `-Dhttps.proxyHost` and `-Dhttps.proxyPort` ).*

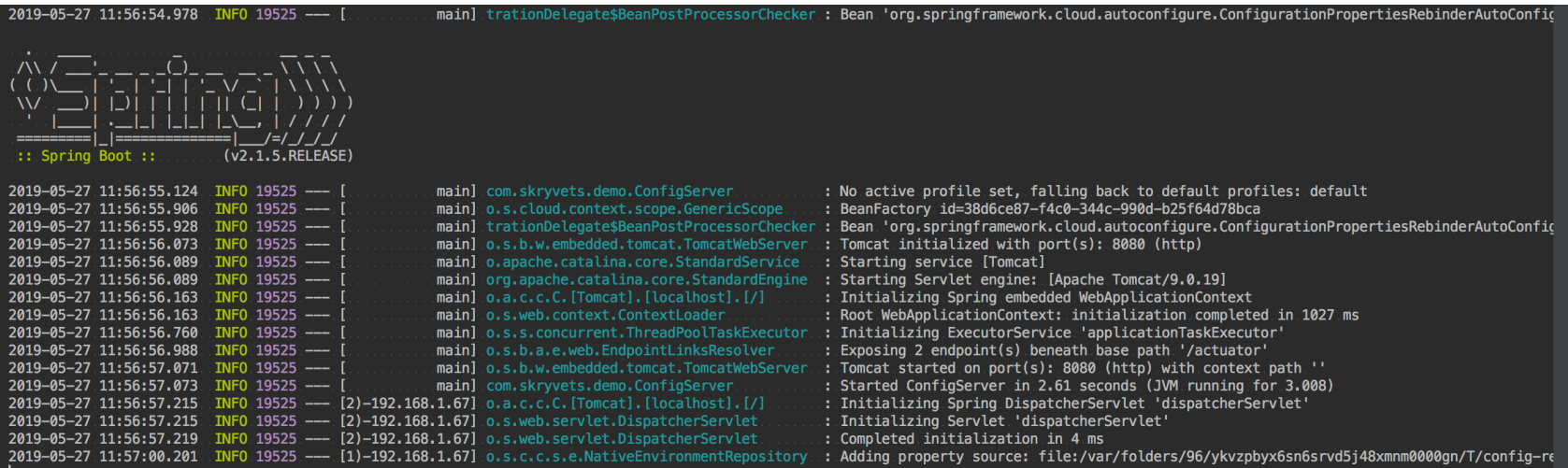
After googling around I found multiple threads on StackOverflow that helped other people. Particularly, this one [Spring Cloud Config cannot clone private bitbucket repository using ssh key](#). However, after applying proper configuration in my `~/.ssh/config` file it still didn't work (as you can see from the screenshot for this post).

## The solution

Turns out the problem is hidden in the bowels of Mac OS Mojave (and any later version). By default, It creates the key in different, new "OpenSSH" format where Spring only supports ssh-rsa (more on this [here](#)). The solution is to re-generate ssh key for the account using the following command:

```
ssh-keygen -m PEM -t rsa -b 4096 -C "your_email@example.com"
```

Once I applied newly generated key, the application was able to start successfully:



Thanks to [Wpigott](#) who pointed this solution.

- Share:
- Twitter

Facebook

Linkedin



**Kubernetes Kustomize - JsonPatches6902 ...**

2 years ago • 4 comments

While official documentation provides a great overview of the basic features such as ...

**Passed Spring Professional ...**

2 years ago • 2 comments

This article is aimed to share my Spring Certified Professional exam ...

**Enhance exception handling when ...**

3 years ago • 3 comments

Customize default "Whitelabel Error Page" to return JSON. Configure your application ...

**Exposing TCP and UDP Services via ingress on ...**

2 years ago • 4 comments

A quick workaround to expose TCP and UDP Services via ingress on Minikube

What do you think?

2 Responses

Upvote

Funny

Love

Surprised

Angry

Sad

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

- Raman Khrapavitski • 2 years ago

Thank you for this solution, it helps.

1 ^ | ▾ • Reply • Share ›
- Book Read • a year ago

Thanks.Who's here from javabrain's spring boot microservice course :)

^ | ▾ • Reply • Share ›
- vivek • 2 years ago

"To avoid surprises, you should ensure that only one entry is present in the

file for the Git server and that it matches the URL you provided to the config server." I am confused by this.I

have this in known\_hosts

\$ cat known\_hosts

bitbucket.org,18.205.93.1 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAubiN81eDcafrgMeLzaFPsw2kNvEcqTKl/VqLat

/xxxxxxxxxxxx<siomemore stuff="">==

So what should known\_hosts have ?

When I start the config client spring boot , I immediately see in config server console:

org.eclipse.jgit.api.errors.TransportException: git@bitbucket.org:<myuserid>c/config-repo.git: Auth fail

at org.eclipse.jgit.api.FetchC...(FetchCommand.java:254) ~[org.eclips

^ | ▾ • Reply • Share ›

Sergey Kryvets Mod ➡ vivek • 2 years ago • edited

That's the quote from the official spring docs.

My understanding that

only one entry is present in the file for the Git server

- it matches the URL you provided to the config server

Url you have in your known\_hosts (like on your example [bitbucket.org](https://bitbucket.org)) matches the url in your application properties or yml file:

spring:

cloud:

config:

server:

git:

uri: [bitbucket.org](https://bitbucket.org)/path-to-your-repo

Now, if that correct, and you still don't have it working (and as you showed in your example you have ssh-rsa which is supported encryption format) are you sure that you can access that repository from your local machine directly (without using spring server)? Try out `ssh -T bitbucket.org` command as it's described in more details here -

<https://help.github.com/en/...>

^ | v • Reply • Share ›

---

 [Subscribe](#)  [Add Disqus to your site](#)[Add Disqus](#)[Add](#)  [Do Not Sell My Data](#)

[Github](#) [Linkedin](#) [Twitter](#) [StackOverflow](#)

2017 - 2021 Sergey Kryvets. All rights reserved.