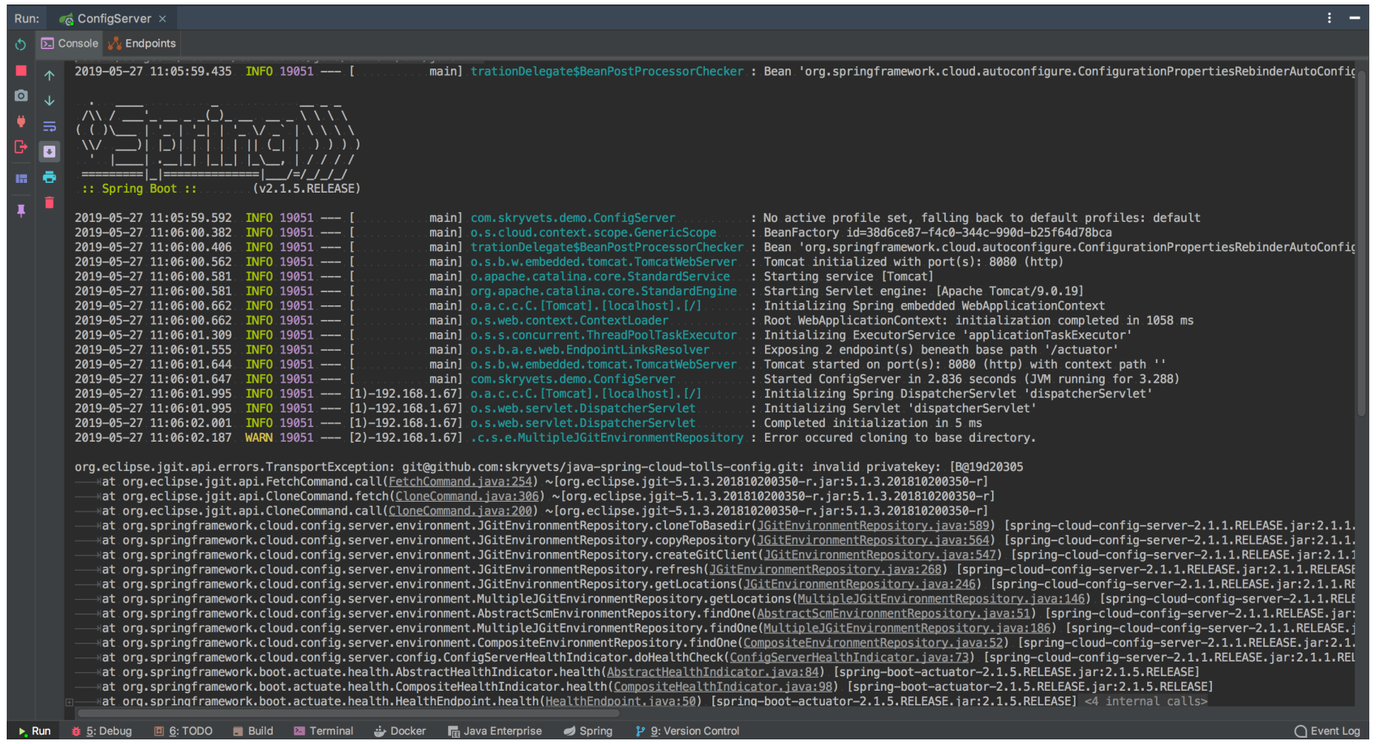
Solved - Spring Cloud Config Server invalid SSH key on a MacOS Mojave

PUBLISHED ON **27 MAY 2019**

LAST UPDATED ON **19 APR 2021**



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`).*

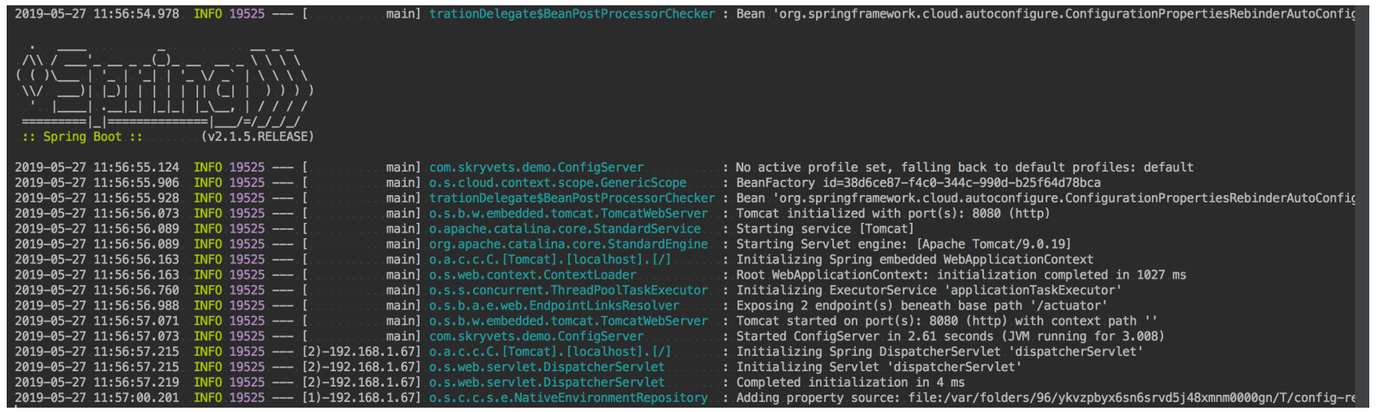
After googling around I found multiple threads on StackOverflow that helped other people. Particularly, this one Spring Cloud Config cannot clone private bitbucket repository using ssh key. However, after applying proper configuration in my `~/.ssh/config` file it still didn't work (as you can see from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

ALSO ON **SKRYVETS.COM**

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

4 Comments skryvets.com **Disqus' Privacy Policy** **Login**

Recommend **Tweet** **Share** **Sort by Best**

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 javabrains 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
/xxxxxxxxxxx<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) matches the url in your application properties or yml file:
spring:
cloud:
config:
server:
git:
uri: 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/....