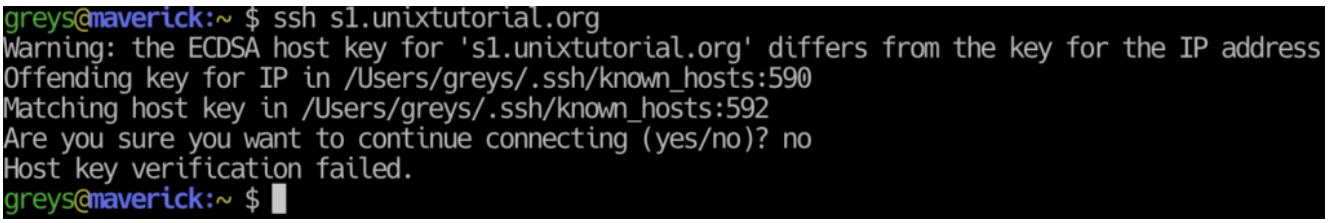
Host Key Verification Failed



Host key verification failed

When reinstalling servers with new versions of operating system or simply reprovisioning VMs under the same hostname, you eventually get this **Host Key Verification Failed** scenario. Should be easy enough to fix, once you're positive that's a valid infrastructure change.

Host Key Verification

Host key verification happens when you attempt to access remote server with SSH. Before verifying if you have a user on the remote server and whether your password or SSH key match that remote user, SSH client must do basic sanity checks on the lower level.

Specifically, SSH client checks if you attempted connecting to the remote server before. And whether anything changed since last time (it shouldn't have).

Server (host) keys must not change during a normal life cycle of a server – they are generated at server/VM build stage (when OpenSSH starts up the first time) and remain the same – it's the server's identity.

This means if your SSH client has one keyprint for a particular server, and then suddenly detects it's a different one – it's flagged as an issue: at best, you're looking at the new, legit server replacement with the same hostname. At worst, someone's trying to intercept your connection and/or pretend to be your server.

Host Key Verification Failed

Here's how I get this error on my Macbook (**s1.unixtutorial.org** doesn't really exist, it's just a hostname I show here as example):

```
greys@maverick:~ $ ssh s1.unixtutorial.org
Warning: the ECDSA host key for 's1.unixtutorial.org' differs from the key for the IP address '51.159.18.142'
Offending key for IP in /Users/greys/.ssh/known_hosts:590
Matching host key in /Users/greys/.ssh/known_hosts:592
Are you sure you want to continue connecting (yes/no)?
```

At this stage your default answer should always be "no", followed by inspection of the known_hosts file to confirm what happened and why identity appears to be different.

If you answer no, you'll get the **Host Key Verification Failed** error:

```
greys@maverick:~ $ ssh s1.unixtutorial.org
```

```
Warning: the ECDSA host key for 's1.unixtutorial.org' differs
from the key for the IP address '51.159.18.142'
Offending key for IP in /Users/greys/.ssh/known_hosts:590
Matching host key in /Users/greys/.ssh/known_hosts:592
Are you sure you want to continue connecting (yes/no)? no
 Host key verification failed.
```

How To Solve Host Key Verification Errors

The output above actually tells you what to do: inspect file known_hosts and look at the lines 590 and 592 specifically. One of them is likely to be obsolete, and if you remove it the issue will go away.

Specifically, if you (like me) just reinstalled the dedicated server or VM with a new OS but kept the original hostname, then the issue is expected (new server definitely generated a new host key), so the solution is indeed to remove old key from the known_hosts file and re-attempt the connection.

First, I edited the /Users/greys/.ssh/known_hosts file and removed the line 590, which looked something like this. We simply need to find the line with given number, or look for the hostname we just tried to ssh into (**s1.unixtutorial.org** in my case):

```
s1.unixtutorial.org,51.159.xx.yy    ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlkzdHAyNTYAAAAxyzAgBPbBCXCL5w8
```

We can try reconnecting now, answer **yes** and connect to the server:

```
greys@maverick:~ $ ssh s1.unixtutorial.org
The authenticity of host 's1.unixtutorial.org (51.159.xx.yy)' can't be established.
ECDSA key fingerprint is SHA256:tviW39xN2M+4eZOUGi8UFvBZoHKaLaijBA581Nrhjac.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 's1.unixtutorial.org,51.159.xx.yy' (ECDSA) to the list of known hosts.
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Fri Feb  7 21:18:35 2020 from unixtutorial.org
[greys@s1 ~]$
```

As you can see, the output now makes a lot more sense: our SSH client can’t establish authenticity of the remote server **s1.unixtutorial.org** – this is because we removed any mention of that server from our known_hosts file in previous step. Answering yes adds info about **s1.unixtutorial.org**, so any later SSH sessions will work just fine:

```
greys@maverick:~ $ ssh s1.unixtutorial.org
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Sat Feb  8 18:31:39 2020 from 93.107.36.193
[greys@s1 ~]$
```

Copying Host Keys to New Server

I should note that in some cases your setup or organisation would require the same host keys to be kept even with server

reinstall. In this case, you'll need to use last know backup of old server to grab SSH host keys from, to re-deploy them onto the new server – I'll show/explain this in one of the future posts.

See Also

- Passwordless SSH
- SSH reference
- SSH key fingerprints
- SSH command
- SSH port