

cf ssh – MANAGING THE APPS AND SERVICES IN DIEGO CONTAINERS

cf-ssh OVERVIEW

To troubleshoot the applications at system level, you need to go to the OS or VM and check files and logs. In containerized PCF version, cf-ssh does that for user.

COMMAND:

```
cf ssh APP_NAME [-i app-instance-index] [-c command] [-L  
[bind_address:]port:host:hostport] [--skip-host-validation] [--skip-remote-  
execution] [--request-pseudo-tty] [--force-pseudo-tty] [--disable-pseudo-tty]
```

OPTIONS

-L : Local port forward specification. This flag can be defined more than once.
--app-instance-index, -i : Application instance index
--command, -c : Command to run. This flag can be defined more than once.
--disable-pseudo-tty, -T : Disable pseudo-tty allocation
--force-pseudo-tty, -tt : Force pseudo-tty allocation
--request-pseudo-tty, -t : Request pseudo-tty allocation
--skip-host-validation, -k : Skip host key validation
--skip-remote-execution, -N : Do not execute a remote command

ENABLING APPS FOR SSH

To do a ssh into the app, we have to configure the environment for ssh, enable ssh to the space and app as well.

Note: Make sure Diego Brain VM is allowed to listen on port 2222. If not then discuss with the F5 LB team to allow port 2222 at VIP level. In cloudeast Prod, the 3.3.167.183 is the VIP configured at F5, which load balances 3.3.81.250, 3.3.81.251, 3.3.81.252, , 3.3.81.253, the router VMs. The IP address of diego brain VM is 3.3.81.166. We added the Diego Brain VM to the 3.3.167.183 VIP to make the list as 3.3.81.250, 3.3.81.251, 3.3.81.252, 3.3.81.253, 3.3.81.166. If this is not done we will get following type of error:

"Error opening SSH connection: dial tcp 3.3.167.183:2222: getsockopt: no route to host"

Or

"Error opening SSH connection: ssh: handshake failed: read tcp 3.3.86.254:49529->3.3.167.183:2222: read: connection reset by peer"

- “Allow SSH access to apps” is enabled in ER-> Settings-> Diego.
- Applications and spaces are running in Diego. DEA based apps are not allowed to do ssh.
- The command to enable ssh to the space:

```
cf allow-space-ssh SPACE_NAME
```

- The command to enable ssh to the app:

```
cf enable-ssh APP_NAME
```

We have taken example of *datamaker-qa* application in Org: *SalesAnalyticsProducts* & Space: *datamaker-qa*. The commands were:

```
[root@aoaaplp00336 openplatform]# cf allow-space-ssh datamaker-qa
ssh support is already enabled in space 'datamaker-qa'
[root@aoaaplp00336 openplatform]#
```

```
[root@aoaaplp00336 openplatform]# cf enable-ssh datamaker-qa
ssh support is already enabled for 'datamaker-qa'
[root@aoaaplp00336 openplatform]
```

Here the applications were already executed with **allow-space-ssh** & **enable-ssh**.

After enabling datamaker-qa space and datamaker-qa application, we have following output to check if the ssh is enabled for app and space:

```
[root@aoaaplp00336 openplatform]# cf ssh-enabled datamaker-qa
ssh support is enabled for 'datamaker-qa'
[root@aoaaplp00336 openplatform]# cf space-ssh-allowed datamaker-qa
ssh support is enabled in space 'datamaker-qa'
[root@aoaaplp00336 openplatform]#
```

Once this is done, we are good to do ssh into apps.

The **cf ssh** is a cool stuff that we achieved by resolving [RITMo446447](https://docs.cloudfoundry.org/devguide/deploy-apps/ssh-apps.html) . Please check it and let us know your experience. More on this: <https://docs.cloudfoundry.org/devguide/deploy-apps/ssh-apps.html>

I want to show you what we have achieved by doing simple cf ssh:

```
[root@aoaaplp00336 openplatform]# cf target -o
SalesAnalyticsProducts -s datamaker-qa
API endpoint:   https://api.cloudeast.inbcu.com (API version:
2.43.0)
User:          admin
Org:           SalesAnalyticsProducts
Space:         datamaker-qa
[root@aoaaplp00336 openplatform]# cf ssh datamaker-qa
vcap@ig74nomkh7i:~$ ll
total 36
drwx----- 6 vcap vcap 4096 May 10 13:52 ./
drwxr-xr-x 4 root root 4096 May 17 13:23 ../
drwxr-xr-x 8 vcap root 4096 May 10 13:52 app/
-rw-r--r-- 1 vcap vcap  220 Apr  9  2014 .bash_logout
-rw-r--r-- 1 vcap vcap 3637 Apr  9  2014 .bashrc
drwxr-xr-x 2 vcap vcap 4096 May 10 13:52 logs/
-rw-r--r-- 1 vcap vcap  675 Apr  9  2014 .profile
-rw-r--r-- 1 vcap vcap 1449 May 10 13:52 staging_info.yml
drwxr-xr-x 4 vcap vcap 4096 May 17 13:24 tmp/
vcap@ig74nomkh7i:~$
```

If you notice, we are in fact in the container of the app called datamaker-qa. Imagine, we have 2 containers (instances) of datamaker-qa. We have the query for that:

```
cf ssh APP_NAME -i 2
```

Alternately you can do an ssh to app from ssh like this:

```
ssh -p 2222 cf:$(cf app datamaker-qa --guid)/0@ssh.cloudeast.inbcu.com
```

When prompted for a passcode enter the output of:

```
cf ssh-code
```

Conclusion:

Cf-ssh is a great improvement from CF. Not only app, you can also ssh into your [service](#). It made *cf files* commands redundant.