

# Cloud Foundry installation tutorial on a MacBook Pro (through Bosh Lite)

Paulo Jerônimo, 2018-06-19 06:45:08 WEST

# Table of Contents

1. Prerequisites .....	1
1.1. References .....	3
2. Install .....	3
2.1. References .....	3
3. Stemcell upload .....	3
3.1. References .....	4
4. Cloud Foundry Deploy .....	4
4.1. References .....	5
5. Logging into the CF API .....	5
6. Deploy a sample app .....	7
6.1. References .....	7
7. View the logs .....	7
7.1. References .....	7
8. Connect a Database .....	7
8.1. References .....	7
9. Scale the app .....	8
9.1. References .....	8
10. Additional Commands .....	8
11. Optional steps (Beyond Cloud Foundry Deploy) .....	8
11.1. Zookeeper Deploy .....	8
11.1.1. References .....	9
11.2. Enabling SSH Access .....	9
11.2.1. References .....	9
12. Start/Stop .....	9
12.1. References .....	10
13. Uninstall .....	10
14. Additional References .....	11

Here I show you my steps to put [Cloud Foundry](#) running on my MacBook Pro through [Bosh Lite](#). My intention is to prepare an environment to deploy some [UAA](#) labs that I'll write in a nearby future. I'm not detailing concepts, writing extensive comments or explaining each command at this moment. I just show you the command and its output. So, **if you want details read the references for each topic.**



This is a work in progress.

## 1. Prerequisites

The prerequisites are listed in this page: <http://bosh.io/docs/quick-start/#prerequisites>.

*Some details relative to my macOS environment:*

- All my commands in this tutorial were done on a [MacBook Pro \(15-inch, 2016\)](#) with 16 GB of memory, 2.7 GHz quad-core processor.
- I use [Homebrew](#) to install packages on my mac. I have details about how I use and install Homebrew in my environment [in this page \(pt-br\)](#).
- I use Bash version 4 on my macOS:

```
$ echo $BASH_VERSION
4.4.23(1)-release
```

Details about why and how to install this version of Bash can be found [in this post](#). To resume, the following commands do the upgrade:



```
brew update && brew install bash && sudo bash -c 'echo
/usr/local/bin/bash >> /etc/shells' && chsh -s /usr/local/bin/bash
```

- I prefer to use the GNU version of [sed](#) to make my scripts compatible with Linux environments. To do this you can [see this question on Stack Overflow](#). To resume, the following commands do this change:

```
brew install gnu-sed --with-default-names
```

- The [VirtualBox](#) version on my machine is shown in the output of the following command:

```
$ VBoxManage --version
5.2.12r122591
```

- The [additional dependencies necessary to run bosh-cli](#) are installed.

Install bosh-cli using Homebrew:

```
brew install cloudfoundry/tap/bosh-cli
```

Check your bosh version:

```
$ bosh -v
version 4.0.1-a18c7230-2018-05-23T23:11:20Z

Succeeded
```

## 1.1. References

- <http://bosh.io/docs/quick-start/#prerequisites>
- <http://bosh.io/docs/cli-v2-install/>

## 2. Install



If you are trying to reinstalling the environment first execute the steps in [Uninstall](#).

Recreate (if existent) `~/bosh-env/virtualbox` and change your current dir to it:

```
rm -rf ~/bosh-env && mkdir -p ~/virtualbox && cd $_
```

Clone [bosh-deployment](#) repo:

```
git clone https://github.com/cloudfoundry/bosh-deployment.git
```

Call `create-env.sh`:

```
./bosh-deployment/virtualbox/create-env.sh
```

Output (*on my machine*): [output.1.txt](#).

Read `.envrc` variables to your current shell:

```
source .envrc
```

Configure bosh to use `vbox` environment:

```
bosh -e vbox env
```

Output (*on my machine*): [output.8.txt](#)

## 2.1. References

- <http://bosh.io/docs/quick-start/#install>

## 3. Stemcell upload

Upload a stemcell:

```
bosh -e vbox upload-stemcell https://bosh.io/d/stemcells/bosh-warden-bosh-lite-ubuntu-trusty-go_agent?v=3586.23 --sha1 11e6461b4aca491e28a90686fe9c92dda03d9dc2
```

Output (on my machine): [output.6.txt](#).



The version specified on the above command usually are updated in [bosh-deployment like in this commit](#). So, pay attention if you are having problems in future executions of this tutorial.

## 3.1. References

- <http://bosh.io/docs/quick-start/#install>
- <https://bosh.cloudfoundry.org/stemcells/>

## 4. Cloud Foundry Deploy

```
git clone https://github.com/cloudfoundry/cf-deployment
cd cf-deployment
```

When I was updating this tutorial the version of stemcell used had not yet been updated to [3586.23](#) on [cf-deployment.yml](#). So, the next command does this change:

```
sed -i 's/3586.16/3586.23/g' cf-deployment.yml
```

```
bosh -e vbox update-cloud-config iaas-support/bosh-lite/cloud-config.yml
```

Output (on my machine): [output.5.txt](#)

```
bosh -e vbox -d cf deploy cf-deployment.yml \
-o operations/bosh-lite.yml \
--vars-store deployment-vars.yml \
-v system_domain=bosh-lite.com
```

Output (on my machine): [output.4.txt](#)



As you can see in this output, this is the most time-consuming command in this tutorial. (*1 hour and 35 minutes on my machine*). This is because many downloads and compilation tasks will be executed. This command can possibly crash (*this happened a few times on my machine*). In that case, I just re-executed the command.

## 4.1. References

- <https://starkandwayne.com/blog/running-cloud-foundry-locally-on-bosh-lite-with-bosh2/>
- <https://docs.cloudfoundry.org/cf-cli/getting-started.html>

## 5. Logging into the CF API

Add the following route so you will be able to reach the CF API and also bosh ssh to the instances you deployed:

```
sudo route add -net 10.244.0.0/16 192.168.50.6
```



To execute the next commands, we need `cf-cli` installed. The references in the previously topic explain this. But, to resume, the `cf-cli` can be installed by typing this two command lines:

```
brew tap cloudfoundry/tap
brew install cf-cli
```

```
$ cf api https://api.bosh-lite.com --skip-ssl-validation
Setting api endpoint to https://api.bosh-lite.com...
OK
```

```
api endpoint:  https://api.bosh-lite.com
api version:   2.113.0
```

```
export CF_ADMIN_PASSWORD=$(bosh int ./deployment-vars.yml --path /cf_admin_password)
```

```
$ cf auth admin $CF_ADMIN_PASSWORD
API endpoint: https://api.bosh-lite.com
Authenticating...
OK
Use 'cf target' to view or set your target org and space.
```

```
$ cf create-org paulojeronimo-org
Creating org paulojeronimo-org as admin...
OK
```

```
Assigning role OrgManager to user admin in org paulojeronimo-org ...
OK
```

TIP: Use 'cf target -o "paulojeronimo-org"' to target new org

```
$ cf orgs
Getting orgs as admin...
```

```
name
paulojeronimo-org
system
```

```
$ cf target -o paulojeronimo-org
api endpoint:  https://api.bosh-lite.com
api version:   2.113.0
user:         admin
org:          paulojeronimo-org
No space targeted, use 'cf target -s SPACE'
```

```
$ cf create-space paulojeronimo-dev
Creating space paulojeronimo-dev in org paulojeronimo-org as admin...
OK
Assigning role RoleSpaceManager to user admin in org paulojeronimo-org / space
paulojeronimo-dev as admin...
OK
Assigning role RoleSpaceDeveloper to user admin in org paulojeronimo-org / space
paulojeronimo-dev as admin...
OK
```

TIP: Use 'cf target -o "paulojeronimo-org" -s "paulojeronimo-dev"' to target new space

```
$ cf target -o "paulojeronimo-org" -s "paulojeronimo-dev"
api endpoint:  https://api.bosh-lite.com
api version:   2.113.0
user:         admin
org:          paulojeronimo-org
space:        paulojeronimo-dev
```



## 6. Deploy a sample app

```
$ cd ~/bosh-env
$ git clone https://github.com/cloudfoundry-samples/spring-music
$ cd spring-music
$ ./gradlew assemble
```

```
cf push --hostname spring-music
```

Output (on my machine): [output.7.txt](#)

Open the URL of the application. In my case (*seeing the last output*) this is the URL: <http://spring-music-brash-genet.bosh-lite.com>.

### 6.1. References

- <https://pivotal.io/platform/pcf-tutorials/getting-started-with-pivotal-cloud-foundry-dev/deploy-the-sample-app>

## 7. View the logs

```
cf apps
```

Output (on my machine): [output.9.txt](#)

```
cf logs spring-music --recent
```

Output (on my machine): [output.a.txt](#)

### 7.1. References

- <https://pivotal.io/platform/pcf-tutorials/getting-started-with-pivotal-cloud-foundry-dev/view-the-logs>

## 8. Connect a Database

TODO

### 8.1. References

- <https://pivotal.io/platform/pcf-tutorials/getting-started-with-pivotal-cloud-foundry-dev/connect->

## 9. Scale the app

TODO

### 9.1. References

- <https://pivotal.io/platform/pcf-tutorials/getting-started-with-pivotal-cloud-foundry-dev/scale-the-app>

## 10. Additional Commands

```
bosh stemcells
```

Output (on my machine): [output.b.txt](#)

```
bosh releases
```

Output (on my machine): [output.c.txt](#)

```
bosh vms
```

Output (on my machine): [output.d.txt](#)

## 11. Optional steps (Beyond Cloud Foundry Deploy)

### 11.1. Zookeeper Deploy

```
bosh -e vbox -d zookeeper deploy <(wget -O-  
https://raw.githubusercontent.com/cppforlife/zookeeper-  
release/master/manifests/zookeeper.yml)
```

Output (on my machine): [output.2.txt](#)

```
bosh -e vbox -d zookeeper run-errand smoke-tests
```

Output (on my machine): [output.3.txt](#)



We can repeat this command many times if we want. Only the task number will be incremented.

### 11.1.1. References

- <https://ultimateguidetobosh.com/tutorials/bosh-lite-virtualbox/>
- <https://github.com/starkandwayne/ultimate-guide-to-bosh/>

## 11.2. Enabling SSH Access

```
$ bosh int creds.yml --path /jumpbox_ssh/private_key > jumpbox.key
$ chmod 600 jumpbox.key
$ bosh -e vbox env
Using environment '192.168.50.6' as client 'admin'
...
$ ssh jumpbox@192.168.50.6 -i jumpbox.key
```

### 11.2.1. References

- <http://bosh.io/docs/jumpbox/>
- <http://bosh.io/docs/terminology/#jumpbox>
- <https://github.com/cloudfoundry/cli-workstation/blob/master/scripts/deploy-cf/bosh-lite-cheat.md>

## 12. Start/Stop

Stop cf vm instances:

```
bosh -e vbox -d cf stop --hard
```

Output (*on my machine*): [output.e.txt](#)

Check status:

```
$ bosh vms
Using environment '192.168.50.6' as client 'admin'

Task 29. Done

Deployment 'cf'

Instance  Process State  AZ  IPs  VM CID  VM Type  Active

0 vms

Succeeded
```

Start cf vm instances:

```
bosh -e vbox -d cf start
```

Output (*on my machine*): [output.f.txt](#)



The above commands stop bosh vms only. To start/stop bosh lite, you need to follow [this steps](#).

## 12.1. References

- <https://docs.cloudfoundry.org/adminguide/start-stop-vms.html>
- <https://github.com/cloudfoundry/bosh-deployment/issues/82>

## 13. Uninstall

This will remove the created VMs from Virtualbox:

```
cd ~/bosh-env/virtualbox
./bosh-deployment/virtualbox/delete-env.sh
```

Output (*on my machine*): [uninstall.1.txt](#).

Bosh consumes a lot of disk space:

```
$ du -hsc ~/.bosh
4.7G    /Users/pj/.bosh
4.7G    total
```

```
$ du -hsc ~/.bosh_virtualbox_cpi
24G    /Users/pj/.bosh_virtualbox_cpi
24G    total
```

So, if you don't mind to download all that is already downloaded/created, you can also type this commands:

```
rm -rf ~/.bosh ~/.bosh_virtualbox_cpi
```

## 14. Additional References

- <http://basics-workshop.cloudfoundry.org/>
  - <https://github.com/cloudfoundry/summit-training-classes>
- <https://stackoverflow.com/questions/36596743/how-to-install-cloudfoundry-on-local-server>
- <https://starkandwayne.com/blog/bosh-lite-on-virtualbox-with-bosh2/>
- <https://starkandwayne.com/blog/running-cloud-foundry-locally-with-bosh-lite/>
- <https://medium.com/@ravijagannathan/install-cloud-foundry-on-bosh-lite-6d3b9a1e416a>
- <https://www.altoros.com/blog/how-to-install-cloud-foundry-locally-with-bosh-lite/>
- <https://github.com/cloudfoundry-community/bosh-lite-demo>
- <https://github.com/EngineerBetter/training-zero-to-hero>
- <http://www.xenonique.co.uk/blog/2018/01/19/how-to-deploy-bosh-lite-and-cloud-foundry-locally-on-mac-book-pro/>
- [http://aussie.linux.com.au/tutorial/bosh/bosh\\_localdev/](http://aussie.linux.com.au/tutorial/bosh/bosh_localdev/)
- <https://gist.github.com/bgandon/6b0826189b8513624e98475bc2f9a538>