

전자정부표준프레임워크 기반의 Open PaaS 개발

Bosh Lite 설치 가이드

|  |  |  |
| --- | --- | --- |
| 작성자 | 김태형 (인) | 2015.7.29 |
| 검토자 | 문철호 (인) | 2015.8.05 |
| 승인자 | 김기현 (인) | 2015.8.10 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-1) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 0.1 | 2015.7.29 | 최초작성 | 김태형 | 김기현 |
| 0.8 | 2015.8.04 | 내부 검토회 결과 반영 (Diego 설치 및 배포 추가) | 김태형 | 김기현 |
| 1.0 | 2016.01.05 | Cf-release 및 diego-release 설치 절차 수정 | 김태형 | 김기현 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 문서 개요 5](#_Toc440544852)

[*1.1.* *목적* 5](#_Toc440544853)

[*1.2.* *범위* 5](#_Toc440544854)

[*1.3.* *참고자료* 5](#_Toc440544855)

[2. BOSH Lite 설치 6](#_Toc440544856)

[*2.1.* *설치전 준비사항* 6](#_Toc440544857)

[*2.2.* *RUBY 설치* 6](#_Toc440544858)

[*2.3.* *BOSH CLI 설치* 7](#_Toc440544859)

[*2.4.* *Virtualbox 및 Vagrant설치* 7](#_Toc440544860)

[*2.5.* *BOSH Lite설치* 8](#_Toc440544861)

[*2.6.* *로컬 가상 머신의 IP 변경* 9](#_Toc440544862)

[*2.7.* *BOSH Lite 설치 Troubleshooting* 10](#_Toc440544863)

[3. Cloud Foundary Diego 설치 및 배포 11](#_Toc440544864)

[*3.1.* *spiff 설치* 11](#_Toc440544865)

[*3.2.* *cf-release 설치* 11](#_Toc440544866)

[*3.3.* *diego-release 설치* 12](#_Toc440544867)

[*3.4.* *Manifest 작성* 12](#_Toc440544868)

[*3.5.* *배포할 release 업로드* 14](#_Toc440544869)

[*3.6.* *BOSH Stemcell 업로드* 14](#_Toc440544870)

[*3.7.* *release 배포* 15](#_Toc440544871)

[*3.8.* *Cloud Foundary Diego 설치 Troubleshooting* 17](#_Toc440544872)

[4. Cloud Foundary Diego 설정 18](#_Toc440544873)

[*4.1.* *Diego 배포 확인* 18](#_Toc440544874)

Executive Summary

본 문서는 BOSH Lite에 대한 설치 가이드이다. BOSH Lite를 설치하고 기본 설정하는 부분을 설명하였다.

본 문서는 다음과 같은 내용들을 포함한다.

* BOSH Lite 설치
* Cloud Foundary Diego 설치 및 배포
* Cloud Foundary Diego 설정

# 문서 개요

## ***목적***

클라우드 환경에 서비스 시스템을 배포할 수 있는 BOSH는 릴리즈 엔지니어링, 개발, 소프트웨어 라이프사이클 관리를 통합한 오픈소스 프로젝트이다. 특히, BOSH Lite는 사용자가 BOSH를 실제 환경에 도입하기 전에 로컬 환경에서 BOSH의 주요 기능을 검증하기 위한 프로젝트이다. 본 문서는 로컬 환경에서 BOSH Lite를 설치하고, BOSH 기능을 테스트 할 수 있는 환경을 구축하는 것이 목표이다.

## ***범위***

가이드 범위는 BOSH Lite 설치하고, 설치한 BOSH Lite에 Warden CF를 릴리즈하는 것까지 기술하였다.

## ***참고자료***

본 문서는 Cloud Foundry의 BOSH Document를 참고로 작성하였다.

Bosh Lite 설치: <https://github.com/cloudfoundry/bosh-lite>

Diego 배포: <https://github.com/cloudfoundry-incubator/diego-release>

# BOSH Lite 설치

## ***설치전 준비사항***

본 설치 가이드는 Linux(Ubuntu 14.04) 환경에서 설치하는 것을 기준으로 하였다. BOSH Lite는 Ruby언어로 되었기 때문에 BOSH Lite 실행을 위한 Ruby가 설치되어 있어야 한다. 또한 BOSH Lite가 설치한 가상 머신을 구동할 가상 환경과 가상 환경을 설치하기 위한 도구로써 Virtualbox와 Vagrant를 설치한다.

* Ruby (1.9.3 이상)
* Bundler
* BOSH CLI
* VirtualBox (4.3.x 권장)
* Vagrant (1.4.3 이상)
* 메모리: 6GB의 메모리를 가진 가상 머신을 구동할 수 있는 여유 메모리
* 디스크: 최소 80GB 이상
* 위의 조건을 충족하는 Linux 또는 OSX 환경 (BOSH는 Windows 환경을 지원하지 않는다. 참조: <https://github.com/cloudfoundry-community/bosh-lite-demo>)
* 가상 머신에서의 Bosh-lite의 설치는 지원하지 않는다.

## ***RUBY 설치***

Ruby 설치 절차는 다음과 같다.

1. 의존 패키지 설치

|  |  |  |  |
| --- | --- | --- | --- |
| * **Ubuntu의 경우**  |  | | --- | | **$ sudo apt-get update**  **$ sudo apt-get install -y build-essential zlibc zlib1g-dev ruby ruby-dev openssl libxslt-dev libxml2-dev libssl-dev libreadline6 libreadline6-dev libyaml-dev libsqlite3-dev sqlite3 libxslt1-dev libpq-dev libmysqlclient-dev** |  * **CentOS의 경우**  |  | | --- | | **$ sudo yum install gcc ruby ruby-devel mysql-devel postgresql-devel postgresql-libs sqlite-devel libxslt-devel libxml2-devel yajl-ruby** |  * **OSX의 경우**  |  | | --- | | **$ xcode-select --install**  **xcode-select: note: install requested for command line developer tools** | |

1. Ruby 설치 관리자 및 Ruby 설치

|  |
| --- |
| **$ curl -L https://get.rvm.io | bash -s stable**  **$ source ~/.rvm/scripts/rvm**  **#Ruby 2.1.6 설치**  **$ rvm install 2.1.6**  **#기본 Ruby버전 설정**  **$ rvm use 2.1.6 --default** |

1. 설치 확인

|  |
| --- |
| **$ ruby -v**  $ ruby -v  ruby 2.1.6p336 (2015-04-13 revision 50298) [x86\_64-linux] |

## ***BOSH CLI 설치***

BOSH CLI의 설치 절차는 다음과 같다.

1. Bosh\_cli를 설치

|  |
| --- |
| **$ gem install bosh\_cli**  cloud4u@XPS-15-9530:~/workspace$ gem install bosh\_cli  Successfully installed bosh\_cli-1.3022.0  1 gem installed |

* gem env의 실행결과 INSTALLATION DIRECTORY의 디렉토리 소유자가 root인 경우 sudo를 사용하여 설치해야 한다.

## ***Virtualbox 및 Vagrant설치***

BOSH Lite 설치를 위해 Vagrant와 Virtualbox를 설치하여 가상 환경을 구성한다.

1. Virtualbox 설치

|  |
| --- |
| **#** Virtualbox 설치  **$ sudo apt-get install virtualbox**  # Virtualbox 확장팩 설치  **$ sudo apt-get install virtualbox-dkms** |

* BOSH Lite는 설치한 virtualbox의 버전과 의존성 관계가 있다. 만일 BOSH Lite를 통해 가상 머신을 생성할 때, 오류가 발생할 경우 다른 버전(4.3.x 이상)의 virtualbox를 설치한다. (설치 가이드에서는 virtualbox 4.3.10을 설치)

1. Vagrant 설치

|  |
| --- |
| # Vagrant 설치  **$ sudo apt-get install vagrant** |

## ***BOSH Lite설치***

BOSH Lite를 설치하는 절차는 다음과 같다.

1. 설치할 환경 생성

|  |
| --- |
| **$ mkdir -p ~/workspace**  **$ cd ~/workspace** |

1. GITHUB에서 BOSH Lite 소스 코드를 다운로드

|  |
| --- |
| # git 설치  **$ sudo apt-get install git**  # bosh lite 다운로드  **$ git clone https://github.com/cloudfoundry/bosh-lite.git** |

1. 등록된 최신의 BOSH Lite 설치

|  |
| --- |
| **$ cd bosh-lite**  **$ vagrant up --provider virtualbox** |

* Vagrantfile이 있는 곳에서 vagrant up을 실행 한다.

1. BOSH 로그인

|  |
| --- |
| **$ bosh target 192.168.50.4 lite**  Target set to `Bosh Lite Director'  Your username: admin  Enter password: \*\*\*\*\*  Logged in as `admin' |

* 기본 아이디와 패스워드는 admin/admin

1. Route 추가

|  |
| --- |
| **$ sudo bin/add-route** |

## ***로컬 가상 머신의 IP 변경***

설치한 가상 머신은 IP가 192.168.50.4로 고정되어 있다. 만일 해당 IP를 변경하고 싶다면 vagrant up을 실행하기 전에 Vagrantfile을 수정해야 한다. 수정할 곳은 붉은 표시가 된 곳의 주석을 삭제하고 바꾸고자 하는 IP주소로 변경한다. 또한 route를 추가 할 때 필요한 add-route도 수정 한다.

※Vagrantfile 수정

|  |
| --- |
| **$ vi ~/workspace/bosh-lite/Vagrantfile**  Vagrant.configure('2') do |config|  config.vm.box = 'cloudfoundry/bosh-lite'  config.vm.provider :virtualbox do |v, override|  override.vm.box\_version = '9000.38.0' # ci:replace  # To use a different IP address for the bosh-lite director, uncomment this line:  ***#*** override.vm.network :private\_network, ip: '***192.168.59.4***', id: :local  end  config.vm.provider :aws do |v, override|  override.vm.box\_version = '9000.38.0' # ci:replace  # To turn off public IP echoing, uncomment this line:  # override.vm.provision :shell, id: "public\_ip", run: "always", inline: "/bin/true"  # To turn off CF port forwarding, uncomment this line:  # override.vm.provision :shell, id: "port\_forwarding", run: "always", inline: "/bin/true"  # Needed for Vagrant 1.7 since it loads Vagrantfile before downloading the box  env = ENV.to\_hash  v.access\_key\_id = env.fetch('BOSH\_AWS\_ACCESS\_KEY\_ID', '')  v.secret\_access\_key = env.fetch('BOSH\_AWS\_SECRET\_ACCESS\_KEY', '')  v.ami = ''  end  end |

※add-route 수정

|  |
| --- |
| **$ vi ~/workspace/bosh-lite/bin/add-route**  #!/bin/bash  echo "Adding the following route entry to your local route table to enable direct warden container access. Your sudo password may be required."  echo " - net 10.244.0.0/19 via ***192.168.50.4***"  if [ `uname` = "Darwin" ]; then  sudo route delete -net 10.244.0.0/19 ***192.168.50.4*** > /dev/null 2>&1  sudo route add -net 10.244.0.0/19 ***192.168.50.4***  elif [ `uname` = "Linux" ]; then  sudo route add -net 10.244.0.0/19 gw ***192.168.50.4***  fi |

## ***BOSH Lite 설치 Troubleshooting***

1. Bosh login 또는 stemcell의 업로드가 잘 되지 않는 경우, 아래의 명령어로 bosh를 재기동 한다.

|  |
| --- |
| **$ vagrant ssh -c "sudo sv restart director"** |

1. 가상 머신을 삭제할 경우, 아래의 명령어를 사용한다.

|  |
| --- |
| **$ vagrant destroy** |

1. Blobstore를 clean할 경우, 아래의 명령어를 사용한다.

|  |
| --- |
| **$ bosh cleanup** |

1. 생성한 BOSH Lite 가상 머신에 로그인 할 경우, 아래의 명령어를 사용한다.

|  |
| --- |
| **$ vagrant ssh** |

1. 생성한 BOSH Lite 가상 머신을 재기동한 후 오류가 발생한 경우, 아래의 명령어를 사용하여 BOSH Lite 가상 머신을 재생성한다.

|  |
| --- |
| **$ bosh cck** |

# Cloud Foundary Diego 설치 및 배포

본 절에서는 BOSH Lite에 Cloud Foundry Diego를 릴리즈하는 절차를 기술한다. Diego는 Cloud Foundary의 새로운 런타임 아키텍처로 이전 버전의 Cloud Foundary의 DEAs와 Health Manager를 대체 한다. 자세한 내용은 다음 사이트를 참조 한다.

참조: <https://github.com/cloudfoundry-incubator/diego-release>

* BOSH Lite
* cf-release
* diego-release
* spiff
* cf-cli

## ***spiff 설치***

spiff는 BOSH의 배포 manifest를 생성하는데 특화된 선언형 프로그램이다. Cf-release 및 diego-release를 위한 manifest파일을 생성하기 위해 필요한 도구이다.

1. GITHUB에서 spiff 소스 코드를 다운로드

|  |
| --- |
| **$ ~/workspace**  **$ wget https://github.com/cloudfoundry-incubator/spiff/releases/download/v1.0/spiff\_linux\_amd64.zip** |

* OSX의 경우 spiff\_darwin\_amd64.zip를 다운로드 한다.

1. 다운로드한 spiff을 압축해제

|  |
| --- |
| **$ unzip spiff\_linux\_amd64.zip** |

1. 압축해제한 spiff을 PATH 환경변수가 지정된 곳으로 옮기고 사용권한을 변경

|  |
| --- |
| **$ sudo mv ~/workspace/spiff /usr/local/bin**  **$ sudo chmod 755 /usr/local/bin/spiff** |

## ***cf-release 설치***

cf-release를 설치하는 절차는 다음과 같다. 먼저 Cloud Foundry의 repository를 로컬에 다운로드한 다음, 설치하고 싶은 CF의 버전을 체크아웃 한다.

1. GITHUB에서 cf-release 소스 코드를 다운로드

|  |
| --- |
| **$ cd ~/workspace**  # cf-release 다운로드  **$ git clone https://github.com/cloudfoundry/cf-release**  **$ cd cf-release**  # 설치할 cf-release 버전으로 체크아웃  **$ git checkout *<cf-release version>***  # 해당 버전의 cf-release 서브모듈 업데이트  **$ ./scripts/update**  **$ git clean -ffd** |

* 본 가이드에서는 v226버전을 설치하였다.

## ***diego-release 설치***

diego-release를 설치하는 절차는 다음과 같다.

1. GITHUB에서 diego-release 소스 코드를 다운로드

|  |
| --- |
| **$ cd ~/workspace**  # diego 다운로드  **$ git clone https://github.com/cloudfoundry-incubator/diego-release.git**  **$ cd diego-release**  # 설치할 diego 버전으로 체크아웃  **$ git checkout *<diego-release 버전>***  # 해당 버전의 diego 서브모듈 업데이트  **$ ./scripts/update**  **$ git clean -ffd** |

* 본 가이드에서는 v0.1442.0 버전을 설치하였다.

## ***Manifest 작성***

[Yml 파일을 생성하기 위해서는 spiff 패키지가 필요하다. spiff 설치에 대해서는 3.1을 참조](#_spiff_설치)

1. cf.yml 파일 생성

|  |
| --- |
| **$ cd ~/workspace/cf-release**  **$ ./scripts/generate-bosh-lite-dev-manifest** |

1. 생성한 cf.yml 파일 확인

|  |
| --- |
| **$ vi ~/workspace/cf-release/bosh-lite/deployments/cf.yml**  #bosh가 타겟 지정한 디렉터의 UUID가 기술되어 있다.  compilation:  cloud\_properties:  name: random  network: cf1  reuse\_compilation\_vms: true  workers: 6  director\_uuid: **71d36859-4f21-446f-8a02-f18d7f1263c6**  jobs:  - instances: 1  name: consul\_z1  <<중략>> |

1. diego.yml 파일 생성

|  |
| --- |
| **$cd ~/workspace/diego-release**  **$ ./scripts/generate-bosh-lite-manifests** |

1. 생성한 diego.yml 파일 확인

|  |
| --- |
| **$ vi ~/workspace/diego-release/bosh-lite/deployments/diego.yml**  compilation:  cloud\_properties: {}  network: diego1  reuse\_compilation\_vms: true  workers: 6  director\_uuid: 71d36859-4f21-446f-8a02-f18d7f1263c6  jobs:  - instances: 1  name: database\_z1  networks:  - name: **diego1**  persistent\_disk: 1024  properties:  consul:  agent:  services:  bbs: {}  etcd: {}  metron\_agent:  zone: z1  <<후략>> |

## ***배포할 release 업로드***

1. BOSH에 cf-release 업로드

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ bosh upload release ~/workspace/cf-release/releases/*<yml 파일명>*** |

* 체크아웃한 cf-release버전과 업로드할 yml파일명이 반드시 일치해야 한다.
* 본 가이드에서는 cf-226.yml를 업로드 하였다.

1. BOSH에 diego-release 업로드

|  |
| --- |
| **$ cd ~/workspace/diego-release**  **$ bosh upload release releases/<diego yml 파일명>** |

* 본 가이드에서는 diego-0.1442.0.yml를 업로드 하였다.

1. BOSH에 garden-linux 업로드

|  |
| --- |
| **$ bosh upload release https://bosh.io/d/github.com/cloudfoundry-incubator/garden-linux-release** |

* 본 가이드에서는 v0.329.0를 업로드 하였다.

1. BOSH에 etcd 업로드

|  |
| --- |
| **$ bosh upload release https://bosh.io/d/github.com/cloudfoundry-incubator/etcd-release** |

* 본 가이드에서는 v20를 업로드 하였다.

## ***BOSH Stemcell 업로드***

1. BOSH에 Warden Stemcell 업로드

|  |
| --- |
| **$ bosh upload stemcell https://bosh.io/d/stemcells/bosh-warden-boshlite-ubuntu-trusty-go\_agent** |

## ***release 배포***

cf-warden과 cf-warden-diego를 배포한다.

1. cf.yml을 BOSH Lite에 deployment한다.

|  |
| --- |
| **$ bosh deployment ~/workspace/cf-release/bosh-lite/deployments/cf.yml** |

1. cf-release를 배포

|  |
| --- |
| **$ bosh -n deploy** |

1. 배포 확인

|  |
| --- |
| **$ bosh vms**  $ bosh vms  Acting as user 'admin' on 'Bosh Lite Director'  Deployment `cf-warden'  Director task 26  Task 26 done  +------------------------------------+---------+---------------+--------------+  | Job/index | State | Resource Pool | IPs |  +------------------------------------+---------+---------------+--------------+  | api\_z1/0 | running | large\_z1 | 10.244.0.134 |  | consul\_z1/0 | running | small\_z1 | 10.244.0.54 |  | doppler\_z1/0 | running | medium\_z1 | 10.244.0.138 |  | etcd\_z1/0 | running | medium\_z1 | 10.244.0.42 |  | ha\_proxy\_z1/0 | running | router\_z1 | 10.244.0.34 |  | hm9000\_z1/0 | running | medium\_z1 | 10.244.0.146 |  | loggregator\_trafficcontroller\_z1/0 | running | small\_z1 | 10.244.0.142 |  | nats\_z1/0 | running | medium\_z1 | 10.244.0.6 |  | postgres\_z1/0 | running | medium\_z1 | 10.244.0.30 |  | router\_z1/0 | running | router\_z1 | 10.244.0.22 |  | runner\_z1/0 | running | runner\_z1 | 10.244.0.26 |  | uaa\_z1/0 | running | medium\_z1 | 10.244.0.130 |  +------------------------------------+---------+---------------+--------------+  VMs total: 12 |

1. diego.yml을 BOSH Lite에 deployment한다.

|  |
| --- |
| **$ bosh deployment ~/workspace/diego-release/bosh-lite/deployments/diego.yml** |

1. diego release 배포

|  |
| --- |
| **$ bosh -n deploy** |

1. 배포 확인

|  |
| --- |
| **$ bosh vms**  $ bosh vms  Acting as user 'admin' on 'Bosh Lite Director'  Deployment `cf-warden'  Director task 28  Task 28 done  +------------------------------------+---------+---------------+--------------+  | Job/index | State | Resource Pool | IPs |  +------------------------------------+---------+---------------+--------------+  | api\_z1/0 | running | large\_z1 | 10.244.0.134 |  | consul\_z1/0 | running | small\_z1 | 10.244.0.54 |  | doppler\_z1/0 | running | medium\_z1 | 10.244.0.138 |  | etcd\_z1/0 | running | medium\_z1 | 10.244.0.42 |  | ha\_proxy\_z1/0 | running | router\_z1 | 10.244.0.34 |  | hm9000\_z1/0 | running | medium\_z1 | 10.244.0.146 |  | loggregator\_trafficcontroller\_z1/0 | running | small\_z1 | 10.244.0.142 |  | nats\_z1/0 | running | medium\_z1 | 10.244.0.6 |  | postgres\_z1/0 | running | medium\_z1 | 10.244.0.30 |  | router\_z1/0 | running | router\_z1 | 10.244.0.22 |  | runner\_z1/0 | running | runner\_z1 | 10.244.0.26 |  | uaa\_z1/0 | running | medium\_z1 | 10.244.0.130 |  +------------------------------------+---------+---------------+--------------+  VMs total: 12  Deployment `cf-warden-diego'  Director task 29  Task 29 done  +--------------------+---------+------------------+---------------+  | Job/index | State | Resource Pool | IPs |  +--------------------+---------+------------------+---------------+  | access\_z1/0 | running | access\_z1 | 10.244.16.6 |  | brain\_z1/0 | running | brain\_z1 | 10.244.16.134 |  | cc\_bridge\_z1/0 | running | cc\_bridge\_z1 | 10.244.16.142 |  | cell\_z1/0 | running | cell\_z1 | 10.244.16.138 |  | database\_z1/0 | running | database\_z1 | 10.244.16.130 |  | route\_emitter\_z1/0 | running | route\_emitter\_z1 | 10.244.16.146 |  +--------------------+---------+------------------+---------------+  VMs total: 6 |

## ***Cloud Foundary Diego 설치 Troubleshooting***

1. cf.yml 생성 중 오류가 발생 할 경우

|  |
| --- |
| $ ~/workspace/cf-release에서 실행하는 경우 다음과 같은 오류가 발생한다.  $ scripts/generate-bosh-lite-dev-manifest  scripts/generate-bosh-lite-dev-manifest: line 13: bosh: command not found  scripts/generate-bosh-lite-dev-manifest: line 14: bosh: command not found  Can only target Bosh Lite Director. Please use 'bosh target' before running this script.  # 위 디렉토리 이외의 장소로 이동하여 실행한다.  $ cd ~/workspace  $ ~/workspace/cf-release/scripts/generate-bosh-lite-dev-manifest |

1. Cf 로그인시 CLI 버전으로 인해 오류가 발생하는 경우

|  |
| --- |
| #cf 확인   * $ cf -v 실행 * 설치한 버전과 다른 버전이 출력되는 경우 * $ echo $PATH 실행, 출력되는 경로상에 다른 cf 파일이 있는지 확인하고 삭제한다. |

# Cloud Foundary Diego 설정

본 절에서는 BOSH Lite에서 Cloud Foundary Diego를 설정하고 애플리케이션을 배포하는 절차를 기술한다.

Diego 플러그인 설치에 대해서는 다음을 참조한다:

<https://github.com/cloudfoundry-incubator/diego-design-notes/blob/master/migrating-to-diego.md#targeting-diego>

* cf-cli (6.13 이상)
* Diego-Enabler

## ***Diego 배포 확인***

1. 아래의 사이트에서 cf CLI 다운로드

|  |
| --- |
| **https://github.com/cloudfoundry/cli/releases** |

* 설치 환경에 맞는 cf CLI (6.13 이상)를 다운로드 한다.

1. Cf CLI 설치

|  |
| --- |
| **$ tar -C /usr/local/bin -xzf *cf-linux-amd64.tgz*** |

* 위 명령어는 다운로드한 디렉토리에서 실행하였다.
* 붉은색의 파일명이 다운로드한 파일이다.

1. Cf 플러그인 설치

|  |
| --- |
| **$ cf add-plugin-repo CF-Community http://plugins.cloudfoundry.org/**  **$ cf install-plugin Diego-Enabler -r CF-Community** |

1. CF 로그인

|  |
| --- |
| # CF 로그인  **$ cf login -a api.bosh-lite.com -u admin -p admin --skip-ssl-validation**  # Docker 사용 설정  **$ cf enable-feature-flag diego\_docker** |

1. APP 배포

|  |
| --- |
| # 조직 및 스페이스 설정  **$ cf create-org *<조직>***  **$ cf target -o *<조직>***  **$ cf create-space *<스페이스>***  **$ cf target -s *<스페이스>***  # APP 업로드 (OpenPaaS가 제공하는 샘플 app으로 테스트할 경우 OpenPaaS-Sample-Apps/Etc/hello-spring를 실행)  **$ cd *<app-directory>***  **$ cf push --no-start**  # APP에 Diego 지원 설정  **$ cf enable-diego *<app-name>***  # APP 시작  **$ cf start *<app-name>*** |

1. 변경 내용: 변경이 발생되는 위치와 변경 내용을 자세히 기록(장/절과 변경 내용을 기술한다.) [↑](#footnote-ref-1)