

전자정부표준프레임워크 기반의 Open PaaS 개발

Bosh Lite 설치 가이드

|  |  |  |
| --- | --- | --- |
| 작성자 | 김태형 (인) | 2015.7.29 |
| 검토자 | 김기현 (인) | 2015.8.01 |
| 승인자 |  |  |

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

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-1) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 0.1 | 2015.7.29 | 최초작성 | 김태형 |  |
| 0.8 | 2015.8.04 | 내부 검토회 결과 반영 (Diego 설치 및 배포 추가) | 김태형 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 문서 개요 5](#_Toc428815789)

[*1.1.* *목적* 5](#_Toc428815790)

[*1.2.* *범위* 5](#_Toc428815791)

[*1.3.* *참고자료* 5](#_Toc428815792)

[2. BOSH Lite 설치 6](#_Toc428815793)

[*2.1.* *설치전 준비사항* 6](#_Toc428815794)

[*2.2.* *RUBY 설치* 6](#_Toc428815795)

[*2.3.* *BOSH CLI 설치* 7](#_Toc428815796)

[*2.4.* *Virtualbox 및 Vagrant설치* 7](#_Toc428815797)

[*2.5.* *BOSH Lite설치* 8](#_Toc428815798)

[*2.6.* *로컬 가상머신의 IP 변경* 8](#_Toc428815799)

[*2.7.* *BOSH Lite 설치 Troubleshooting* 10](#_Toc428815800)

[3. Cloud Foundary Diego 설치 및 배포 11](#_Toc428815801)

[*3.1.* *cf-release 설치* 11](#_Toc428815802)

[*3.2.* *cf-release 업로드* 12](#_Toc428815803)

[*3.3.* *spiff 설치* 12](#_Toc428815804)

[*3.4.* *diego-release 설치* 13](#_Toc428815805)

[*3.5.* *diego-release 업로드* 13](#_Toc428815806)

[*3.6.* *BOSH Stemcell 업로드* 15](#_Toc428815807)

[*3.7.* *director.yml 생성* 17](#_Toc428815808)

[*3.8.* *cf-release 배포* 17](#_Toc428815809)

[*3.9.* *diego-release 배포* 23](#_Toc428815810)

[*3.10.* *CF 로그인* 27](#_Toc428815811)

[*3.11.* *Cloud Foundary Diego 설치 Troubleshooting* 27](#_Toc428815812)

[4. Cloud Foundary Diego 설정 30](#_Toc428815813)

[*4.1.* *배포한 Diego 설정* 30](#_Toc428815814)

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 Lite를 실행하기 Ruby가 설치되어 있어야 한다. 또한 BOSH Lite가 설치한 가상 머신을 구동할 가상 환경과 가상 환경을 설치하기 위한 도구로써 Virtualbox와 Vagrant를 설치해야 한다.

* 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>)

## ***RUBY 설치***

Ruby 설치 절차는 다음과 같다.

1. 패키지에서 설치

|  |
| --- |
| **#의존 패키지 설치**  **$ sudo apt-get install libgdbm-dev libncurses5-dev automake libtool bison libffi-dev**  **$ curl -L https://get.rvm.io | bash -s stable**  **$ source ~/.rvm/scripts/rvm**  **#Ruby 2.2.2 설치**  **$ rvm install 2.2.2**  **#기본으로 Ruby버전 설정**  **$ rvm use 2.2.2 --default** |

1. 설치 확인

|  |
| --- |
| **$ ruby -v**  cloud4u@XPS-15-9530:~/Downloads/ruby-2.2.2$ ruby -v  ruby 2.2.2p95 (2015-04-13 revision 50295) [x86\_64-linux] |

## ***BOSH CLI 설치***

BOSH CLI의 설치 절차는 다음과 같다.

1. 설치할 환경 생성

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

1. GITHUB에서 BOSH CLI 소스 코드를 다운로드

|  |
| --- |
| **$ git clone https://github.com/cloudfoundry/bosh.git** |

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 |

※설치 중 오류가 발생할 경우 다음의 패키지를 추가로 설치해야 한다.

|  |  |
| --- | --- |
| 오류 | 추가로 설치할 패키지 |
| Ruby not installed | ruby |
| mkmf not found | ruby-dev |
| libxml not found | libxml2-dev |
| (compile.log:) make not found | build-essential |
| libsqlite not found | libsqlite3-dev |
| libxslt is missing | libxslt1-dev |
| Can’t find libpq-fe.h header | libpq-dev |
| checking for mysql\_query() … no | libmysqlclient-dev |

※참고

|  |  |
| --- | --- |
| OS | 설치 명령어 |
| Ubuntu | sudo apt-get install <설치할 패키지> |
| CentOS | Yum install <설치할 패키지> |

## ***Virtualbox 및 Vagrant설치***

BOSH Lite는 가상 환경 구축을 위해 Vagrant를 사용한다. Vagrant는 완벽한 가상 개발 환경을 구축해 주는 오픈 소스이다. 또한 BOSH Lite는 가상 환경으로 Virtualbox를 사용한다.

1. Virtualbox 설치

|  |
| --- |
| **$ sudo apt-get install virtualbox** |

* BOSH Lite는 설치한 virtualbox의 버전과 의존성 관계가 있다. 만일 BOSH Lite를 통해 가상머신을 생성할 때, 오류가 발생한다면 낮은 버전의 virtualbox를 설치할 필요가 있다. (설치가이드는 virtualbox 4.3.10을 기준으로 작성)

1. Vagrant 설치

|  |
| --- |
| **$ sudo apt-get install vagrant**  **$ sudo apt-get install virtualbox-dkms** |

## ***BOSH Lite설치***

BOSH Lite를 설치하는 절차는 다음과 같다.

1. GITHUB에서 BOSH Lite 소스 코드를 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 실행해야 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' |

* login 기본 사용자와 패스워드는 admin/admin

1. Route 추가

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

## ***로컬 가상머신의 IP 변경***

설치한 가상머신는 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 변경하면 된다. 또한 route를 추가 할 때 필요한 add-route도 수정해야 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 가상머신에 로그인 할 경우, 아래의 명령어를 사용한다.

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

1. Bosh 명령어 실행시 gem missing오류가 발생할 경우 (gem 재설치)

|  |
| --- |
| **$ cd ~/workspace/bosh**  **$ bundle install** |

# 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

## ***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**  **$ git clone https://github.com/cloudfoundry/cf-release**  **$ cd cf-release**  **$ git checkout *<cf-release version>***  **$ ./update**  cloud4u@XPS-15-9530:~/workspace/cf-release$ git checkout v180  Note: checking out 'v180'.  You are in 'detached HEAD' state. You can look around, make experimental  changes and commit them, and you can discard any commits you make in this  state without impacting any branches by performing another checkout.  If you want to create a new branch to retain commits you create, you may  do so (now or later) by using -b with the checkout command again. Example:  git checkout -b new\_branch\_name  HEAD is now at d0a8072... add blobs for release v180.  $ git submodule update --init --recursive –force |

## ***cf-release 업로드***

1. 다운로드한 release를 bosh에 업로드

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ bosh upload release ~/workspace/cf-release/releases/*<yml 파일명>***  cloud4u@XPS-15-9530:~/workspace/cf-release$ bosh upload release ~/workspace/cf-release/releases/cf-180.yml  Acting as user 'admin' on 'Bosh Lite Director'  Downloading from blobstore (id=2ee23705-597f-4c14-bbb2-8cd22b7350c0)...  …  Started creating new jobs > gorouter/22. Done (00:00:00)  Done creating new jobs (00:00:01)  Started release has been created > cf/180. Done (00:00:00)  Task 1 done  Started 2015-07-29 02:27:39 UTC  Finished 2015-07-29 02:28:00 UTC  Duration 00:00:21  Release uploaded |

* 체크아웃한 cf-release버전과 업로드할 yml파일명이 반드시 일치해야 한다.
* 본 가이드에서는 cf-212버전을 설치하였다.

## ***spiff 설치***

1. GITHUB에서 spiff 소스 코드를 다운로드

|  |
| --- |
| **$ 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/cf-release/spiff /usr/local/bin**  **$ sudo chmod 755 /usr/local/bin/spiff** |

## ***diego-release 설치***

diego-release를 설치하는 절차는 다음과 같다.

1. GITHUB에서 diego-release 소스 코드를 다운로드

|  |
| --- |
| **$ cd ~/workspace**  **$ git clone https://github.com/cloudfoundry-incubator/diego-release.git**  **$ git checkout <diego-release 버전>**  **$ ./scripts/update**  cloud4u@XPS-15-9530:~/workspace$ git clone https://github.com/cloudfoundry-incubator/diego-release.git  Cloning into 'diego-release'...  remote: Counting objects: 26800, done.  remote: Compressing objects: 100% (75/75), done.  remote: Total 26800 (delta 30), reused 0 (delta 0), pack-reused 26710  Receiving objects: 100% (26800/26800), 6.77 MiB | 1.28 MiB/s, done.  Resolving deltas: 100% (16669/16669), done.  Checking connectivity... done. |

* 본 가이드에서는 0.1304.0 버전을 설치하였다.

## ***diego-release 업로드***

1. diego-release up

|  |
| --- |
| **$ cd ~/workspace/diego-release**  **$ bosh upload release releases/<diego yml 파일명>**  cloud4u@XPS-15-9530:~/workspace/diego-release$ bosh upload release releases/diego-0.1304.0.yml  Acting as user 'admin' on 'Bosh Lite Director'  Copying packages  ----------------  route\_emitter  nsync  converger  etcd  receptor  acceptance-tests  garden-linux  golang\_1.4  smoke-tests  canary  docker\_app\_lifecycle  auctioneer  <<중략>>  Started creating new jobs > garden-linux/5c7065893556c064cfd34718ebc3da1c88f17e92. Done (00:00:01)  Started creating new jobs > smoke-tests/85b0332a1aeadd8c0dbe25b158151be0247660c9. Done (00:00:00)  Started creating new jobs > canary/0bef926471fa854e5521497eec77bd7f0774b845. Done (00:00:00)  Started creating new jobs > auctioneer/8d2e6817a62a76bd7a91bb1c0092d61d5a6dca2f. Done (00:00:00)  Started creating new jobs > file\_server/4dfaa90a6be1c2c7d62355512c0dfb711c4be382. Done (00:00:00)  Started creating new jobs > ssh\_proxy/137c69b592076a95af2408e3d832ef5e2c221cb5. Done (00:00:00)  Started creating new jobs > rep/1fc9f9a01f696db600bc7c92cae964270f2e9455. Done (00:00:00)  Started creating new jobs > tps/19b030497a141c30e082ba37c131f024ff2482b9. Done (00:00:00)  Started creating new jobs > runtime\_metrics\_server/88831f69f5557df5d2d43b7c1353846fa644ba25. Done (00:00:00)  Started creating new jobs > stager/a322ac6deb2c32405ac0db34af8f65f1dd163d3d. Done (00:00:00)  Done creating new jobs (00:00:01)  Started release has been created > diego/0.1304.0. Done (00:00:00)  Task 29 done  Started 2015-08-07 08:26:17 UTC  Finished 2015-08-07 08:26:26 UTC  Duration 00:00:09  Release uploaded |

* 본 가이드에서는 diego-0.1304.0.yml를 업로드 하였다.

## ***BOSH Stemcell 업로드***

1. Warden Stemcell 다운로드

|  |
| --- |
| **$ curl -L -J -O https://bosh.io/d/stemcells/bosh-warden-boshlite-ubuntu-trusty-go\_agent**  cloud4u@XPS-15-9530:~/workspace/bosh-lite$ curl -L -J -O https://bosh.io/d/stemcells/bosh-warden-boshlite-ubuntu-trusty-go\_agent  % Total % Received % Xferd Average Speed Time Time Time Current  Dload Upload Total Spent Left Speed  100 131 0 131 0 0 55 0 --:--:-- 0:00:02 --:--:-- 55  100 431M 100 431M 0 0 2344k 0 0:03:08 0:03:08 --:--:-- 1583k |

* ***반드시 manifests/cf-manifest에서 기술한 stemcell을 다운로드 한다.***

|  |
| --- |
| **$ vi manifests/cf-manifest.yml**  compilation:  cloud\_properties:  name: random  network: cf1  reuse\_compilation\_vms: true  workers: 6  director\_uuid: 737bd8ee-f4f9-4aec-9b82-80f3d0be26f1  <<중략>>  meta:  environment: cf-warden  stemcell:  name: ***bosh-warden-boshlite-ubuntu-trusty-go\_agent***  version: latest  name: cf-warden  <<이후 생략>> |

1. BOSH에 Warden Stemcell 업로드

|  |
| --- |
| **$ bosh upload stemcell bosh-warden-boshlite-ubuntu-trusty-go\_agent**  cloud4u@XPS-15-9530:~/workspace/bosh-lite$ bosh upload stemcell bosh-warden-boshlite-ubuntu-trusty-go\_agent  [WARNING] Loading the cli took 5.1 seconds, consider cleaning your gem environment  Acting as user 'admin' on 'Bosh Lite Director'  Verifying stemcell...  File exists and readable OK  Verifying tarball...  Read tarball OK  Manifest exists OK  Stemcell image file OK  Stemcell properties OK  Stemcell info  -------------  Name: bosh-warden-boshlite-ubuntu-trusty-go\_agent  Version: 2776  Checking if stemcell already exists...  No  Uploading stemcell...  bosh-warden-b: 100% |oooooooooooooooooooooooooooooooooooooooooooo| 431.1MB 38.3MB/s Time: 00:00:11  Director task 2  Started update stemcell  Started update stemcell > Extracting stemcell archive. Done (00:00:03)  Started update stemcell > Verifying stemcell manifest. Done (00:00:00)  Started update stemcell > Checking if this stemcell already exists. Done (00:00:00)  Started update stemcell > Uploading stemcell bosh-warden-boshlite-ubuntu-trusty-go\_agent/2776 to the cloud. Done (00:00:13)  Started update stemcell > Save stemcell bosh-warden-boshlite-ubuntu-trusty-go\_agent/2776 (aba374c8-b549-438d-713f-facfdf2bbe5b). Done (00:00:00)  Done update stemcell (00:00:16)  Task 2 done  Started 2015-07-29 05:06:02 UTC  Finished 2015-07-29 05:06:18 UTC  Duration 00:00:16  Stemcell uploaded and created. |

## ***director.yml 생성***

director.yml 생성 절차는 다음과 같다.

1. director.yml 파일 생성

|  |
| --- |
| **$ mkdir -p ~/deployments/bosh-lite**  **$ cd ~/workspace/diego-release**  **$ ./scripts/print-director-stub > ~/deployments/bosh-lite/director.yml** |

1. 생성한 director.yml 파일 확인

|  |
| --- |
| **$ vi ~/deployments/bosh-lite/director.yml**  #bosh가 타겟 지정한 디렉터의 UUID가 기술되어 있다.  director\_uuid: 71d36859-4f21-446f-8a02-f18d7f1263c6 |

## ***cf-release 배포***

yml 파일을 생성하고 cf-release를 배포한다.

1. cf.yml 파일 생성

* [Yml 파일을 생성하기 위해서는 spiff 패키지가 필요하다. spiff 설치에 대해서는 3.3을 참조](#_spiff_설치)

|  |
| --- |
| **$ cd ~/workspace/cf-release**  **$ ./generate\_deployment\_manifest warden ~/deployments/bosh-lite/director.yml ~/workspace/diego-release/stubs-for-cf-release/enable\_consul\_with\_cf.yml ~/workspace/diego-release/stubs-for-cf-release/enable\_diego\_ssh\_in\_cc.yml > ~/deployments/bosh-lite/cf.yml** |

1. 생성한 cf.yml 파일 확인

|  |
| --- |
| **$ vi ~/deployments/bosh-lite/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>>  resource\_pools:  - cloud\_properties:  name: random  env:  bosh:  password: $6$4gDD3aV0rdqlrKC$2axHCxGKIObs6tAmMTqYCspcdvQXh3JJcvWOY2WGb4SrdXtnCyNaWlrf3WEqvYR2MYizEGp3kMmbpwBC6jsHt0  name: small\_z1  network: cf1  stemcell:  name: **bosh-warden-boshlite-ubuntu-trusty-go\_agent #stemcell 버전**  version: latest  <<이하 생략>> |

1. cf.yml을 BOSH Lite에 deployment

|  |
| --- |
| **$ bosh deployment ~/deployments/bosh-lite/cf.yml**  cloud4u@XPS-15-9530:~/deployments/bosh-lite$ bosh deployment ~/deployments/bosh-lite/cf.yml  Deployment set to `/home/cloud4u/deployments/bosh-lite/cf.yml' |

1. 생성한 Yml 파일을 diego-release환경에 맞게 수정

|  |
| --- |
| **$ vi ~/deployments/bosh-lite/cf.yml**  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  networks:  - name: cf1  static\_ips:  - 10.244.0.54  persistent\_disk: 1024  properties:  consul:  agent:  mode: server  metron\_agent:  zone: z1  resource\_pool: medium\_z1  templates:  - name: consul\_agent  release: cf  - name: metron\_agent  release: cf  update:  max\_in\_flight: 1  serial: true  <<중략>>  - instances: **1 🡪 0으로 수정**  name: hm9000\_z1  networks:  - name: cf1  properties:  metron\_agent:  zone: z1  networks:  apps: cf1  resource\_pool: medium\_z1  templates:  - name: hm9000  release: cf  - name: metron\_agent  release: cf  update: {}  - instances: 0  name: hm9000\_z2  networks:  - name: cf2  properties:  metron\_agent:  zone: z2  networks:  apps: cf2  resource\_pool: medium\_z2  templates:  - name: hm9000  release: cf  - name: metron\_agent  release: cf  update: {}  - instances: **1 🡪 0으로 수정**  name: runner\_z1  networks:  - name: cf1  **static\_ips: 🡪 라인전체 삭제**  **- 10.244.0.26 🡪 라인전체 삭제**  properties:  dea\_next:  zone: z1  metron\_agent:  zone: z1  networks:  apps: cf1  resource\_pool: runner\_z1  templates:  - name: dea\_next  release: cf  - name: dea\_logging\_agent  release: cf  - name: metron\_agent  release: cf  update:  max\_in\_flight: 1  <<중략>>  - dns  default\_to\_diego\_backend: **false 🡪 true로 수정**  development\_mode: false  diego\_docker: **false 🡪 true로 수정**  directories: null  disable\_custom\_buildpacks: false  <<중략>>  thresholds:  api:  alert\_if\_above\_mb: null  restart\_if\_above\_mb: null  restart\_if\_consistently\_above\_mb: null  worker:  alert\_if\_above\_mb: null  restart\_if\_above\_mb: null  restart\_if\_consistently\_above\_mb: null  user\_buildpacks: []  users\_can\_select\_backend: **true 🡪 false로 수정**  **<<후략>>** |

1. 생성한 release를 배포

|  |
| --- |
| **$**  **bosh deploy**  cloud4u@XPS-15-9530:~/workspace$ bosh deploy  Acting as user 'admin' on deployment 'cf-warden' on 'Bosh Lite Director'  Getting deployment properties from director...  Unable to get properties list from director, trying without it...  Cannot get current deployment information from director, possibly a new deployment  Please review all changes carefully  Deploying  ---------  Are you sure you want to deploy? (type 'yes' to continue): yes  Director task 35  Started preparing deployment  Started preparing deployment > Binding deployment. Done (00:00:00)  Started preparing deployment > Binding releases. Done (00:00:00)  <<중략>>  Started preparing configuration > Binding configuration. Done (00:00:02)  Started updating job consul\_z1 > consul\_z1/0 (canary). Done (00:00:10)  Started updating job ha\_proxy\_z1 > ha\_proxy\_z1/0 (canary). Done (00:00:48)  Started updating job nats\_z1 > nats\_z1/0 (canary). Done (00:00:10)  Started updating job etcd\_z1 > etcd\_z1/0 (canary). Done (00:00:31)  Started updating job postgres\_z1 > postgres\_z1/0 (canary). Done (00:00:12)  Started updating job uaa\_z1 > uaa\_z1/0 (canary). Done (00:00:58)  Started updating job api\_z1 > api\_z1/0 (canary). Done (00:03:30)  Started updating job hm9000\_z1 > hm9000\_z1/0 (canary). Done (00:00:30)  Started updating job runner\_z1 > runner\_z1/0 (canary). Done (00:00:36)  Started updating job doppler\_z1 > doppler\_z1/0 (canary). Done (00:00:10)  Started updating job loggregator\_trafficcontroller\_z1 > loggregator\_trafficcontroller\_z1/0 (canary). Done (00:00:09)  Started updating job router\_z1 > router\_z1/0 (canary). Done (00:00:11)  Task 35 done  Started 2015-08-06 10:56:30 UTC  Finished 2015-08-06 11:05:12 UTC  Duration 00:08:42  Deployed `cf-warden' to `Bosh Lite Director' |

1. 배포한 release 확인

|  |
| --- |
| **$**  **bosh vms**  cloud4u@XPS-15-9530:~/workspace$ bosh vms  Acting as user 'admin' on 'Bosh Lite Director'  Deployment `cf-warden'  Director task 36  Task 36 done  +------------------------------------+---------+---------------+--------------+  | Job/index | State | Resource Pool | IPs |  +------------------------------------+---------+---------------+--------------+  | api\_z1/0 | running | large\_z1 | 10.244.0.134 |  | consul\_z1/0 | running | medium\_z1 | 10.244.0.54 |  | doppler\_z1/0 | running | medium\_z1 | 10.244.0.142 |  | etcd\_z1/0 | running | medium\_z1 | 10.244.0.42 |  | ha\_proxy\_z1/0 | running | router\_z1 | 10.244.0.34 |  | loggregator\_trafficcontroller\_z1/0 | running | small\_z1 | 10.244.0.146 |  | 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 |  | uaa\_z1/0 | running | medium\_z1 | 10.244.0.130 |  +------------------------------------+---------+---------------+--------------+  VMs total: 10 |

## ***diego-release 배포***

diego.yml 파일을 생성하고 diego-release를 배포한다.

1. diego.yml 파일 생성

* [Yml 파일을 생성하기 위해서는 spiff 패키지가 필요하다. spiff 설치에 대해서는 3.3을 참조](#_spiff_설치)

|  |
| --- |
| **$ cd ~/workspace/diego-release**  **$ ./scripts/generate-deployment-manifest ~/deployments/bosh-lite/director.yml manifest-generation/bosh-lite-stubs/property-overrides.yml manifest-generation/bosh-lite-stubs/instance-count-overrides.yml manifest-generation/bosh-lite-stubs/persistent-disk-overrides.yml manifest-generation/bosh-lite-stubs/iaas-settings.yml manifest-generation/bosh-lite-stubs/additional-jobs.yml ~/deployments/bosh-lite > ~/deployments/bosh-lite/diego.yml** |

1. 생성한 diego.yml 파일 확인

|  |
| --- |
| **$ vi ~/deployments/bosh-lite/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  resource\_pool: database\_z1  templates:  - name: etcd  release: diego  - name: bbs  release: diego  - name: consul\_agent  release: cf  - name: metron\_agent  release: cf  <<후략>> |

1. diego.yml을 BOSH Lite에 deployment

|  |
| --- |
| **$ bosh deployment ~/deployments/bosh-lite/diego.yml**  cloud4u@XPS-15-9530:~/workspace/diego-release$ bosh deployment ~/deployments/bosh-lite/diego.yml  Deployment set to `/home/cloud4u/deployments/bosh-lite/diego.yml' |

1. diego release 배포

|  |
| --- |
| **$ bosh deploy**  cloud4u@XPS-15-9530:~/workspace$ bosh deploy  Acting as user 'admin' on deployment 'cf-warden-diego' on 'Bosh Lite Director'  Getting deployment properties from director...  Unable to get properties list from director, trying without it...  Cannot get current deployment information from director, possibly a new deployment  Please review all changes carefully  Deploying  ---------  Are you sure you want to deploy? (type 'yes' to continue): yes  Director task 37  Started preparing deployment  Started preparing deployment > Binding deployment. Done (00:00:00)  Started preparing deployment > Binding releases. Done (00:00:00)  Started preparing deployment > Binding existing deployment. Done (00:00:00)  Started preparing deployment > Binding resource pools. Done (00:00:00)  <<중략>>  Started updating job etcd\_z1 > etcd\_z1/0 (canary). Done (00:00:13)  Started updating job brain\_z1 > brain\_z1/0 (canary). Done (00:00:14)  Started updating job cell\_z1 > cell\_z1/0 (canary)  Started updating job cc\_bridge\_z1 > cc\_bridge\_z1/0 (canary)  Started updating job route\_emitter\_z1 > route\_emitter\_z1/0 (canary)  Started updating job access\_z1 > access\_z1/0 (canary)  Done updating job route\_emitter\_z1 > route\_emitter\_z1/0 (canary) (00:00:14)  Done updating job access\_z1 > access\_z1/0 (canary) (00:00:14)  Done updating job cc\_bridge\_z1 > cc\_bridge\_z1/0 (canary) (00:00:17)  Done updating job cell\_z1 > cell\_z1/0 (canary) (00:00:33)  Task 34 done  Started 2015-08-07 09:10:24 UTC  Finished 2015-08-07 09:11:36 UTC  Duration 00:01:12  Deployed `cf-warden-diego' to `Bosh Lite Director' |

1. diego release 배포 확인

|  |
| --- |
| **$ bosh vms**  cloud4u@XPS-15-9530:~/workspace/diego-release$ bosh vms  Acting as user 'admin' on 'Bosh Lite Director'  Deployment `cf-warden'  Director task 35  Task 35 done  +------------------------------------+---------+---------------+--------------+  | Job/index | State | Resource Pool | IPs |  +------------------------------------+---------+---------------+--------------+  | api\_z1/0 | running | large\_z1 | 10.244.0.134 |  | consul\_z1/0 | running | medium\_z1 | 10.244.0.54 |  | doppler\_z1/0 | running | medium\_z1 | 10.244.0.142 |  | etcd\_z1/0 | running | medium\_z1 | 10.244.0.42 |  | ha\_proxy\_z1/0 | running | router\_z1 | 10.244.0.34 |  | loggregator\_trafficcontroller\_z1/0 | running | small\_z1 | 10.244.0.146 |  | 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 |  | uaa\_z1/0 | running | medium\_z1 | 10.244.0.130 |  +------------------------------------+---------+---------------+--------------+  VMs total: 10  Deployment `cf-warden-diego'  Director task 36  Task 36 done  +--------------------+---------+------------------+--------------+  | Job/index | State | Resource Pool | IPs |  +--------------------+---------+------------------+--------------+  | access\_z1/0 | running | access\_z1 | 10.244.16.46 |  | brain\_z1/0 | running | brain\_z1 | 10.244.16.6 |  | cc\_bridge\_z1/0 | running | cc\_bridge\_z1 | 10.244.16.14 |  | cell\_z1/0 | running | cell\_z1 | 10.244.16.10 |  | etcd\_z1/0 | running | etcd\_z1 | 10.244.16.2 |  | route\_emitter\_z1/0 | running | route\_emitter\_z1 | 10.244.16.18 |  +--------------------+---------+------------------+--------------+  VMs total: 6 |

## ***CF 로그인***

1. 아래의 사이트에서 cf CLI 다운로드

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

* 설치 환경에 맞는 CLI (6.11.3 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. 배포한 CF를 확인

|  |
| --- |
| **$**  **cf login -a api.10.244.0.34.xip.io -u admin -p admin --skip-ssl-validation**  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf login -a api.10.244.0.34.xip.io -u admin -p admin --skip-ssl-validation  API endpoint: api.10.244.0.34.xip.io  Authenticating...  OK    API endpoint: https://api.10.244.0.34.xip.io (API version: 2.29.0)  User: admin  No org or space targeted, use 'cf target -o ORG -s SPACE' |

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

1. Cf.yml 생성 중 오류가 발생 할 경우

|  |
| --- |
| #<다운로드 디렉토리>/bosh-lite-package/manifest/cf.yml 을 복사하여 사용한다.  단, 디렉터의 UUID는 반드시 설치 환경에 맞게 수정한다.  #설치 환경의 디렉터 조회   * **$ bosh status 실행**   cloud4u@XPS-15-9530:/bosh-lite-package/packages/cf-CLI$ bosh status  Config  /home/cloud4u/.bosh\_config  Director  Name Bosh Lite Director  URL https://192.168.50.4:25555  Version 1.2922.0 (00000000)  User admin  UUID **71d36859-4f21-446f-8a02-f18d7f1263c6 #디렉터의 UUID**  CPI cpi  dns disabled  compiled\_package\_cache enabled (provider: local)  snapshots disabled  Deployment  Manifest /home/cloud4u/workspace/bosh-lite/manifests/cf-manifest.yml   * **$ vi ~/deployments/bosh-lite/cf.yml 실행**   compilation:  cloud\_properties:  name: random  network: cf1  reuse\_compilation\_vms: true  workers: 6  director\_uuid: **71d36859-4f21-446f-8a02-f18d7f1263c6 #조회한 디렉터의 UUID로 변경한다.**  jobs:  - instances: 0  name: consul\_z1    <<이하 생략>>  #spiff을 설치 하지 않은 경우,   * spiff을 설치한다.   # ~/workspace/cf-release/src/loggregator/manifest-templates/cf-lamb.yml 파일이 없는 경우   * <https://github.com/cloudfoundry/loggregator> * 위 사이트에서 loggregator를 다운 받아 설치 한다. * 반드시 cf-lamb.yml의 파일 위치가 cf-release/src/loggregator/manifest-templates/cf-lamb.yml 가 되도록 설치한다. |

1. create release중 ruby 관련 오류가 발생하는 경우

|  |
| --- |
| # release를 생성하는 데 필요한 ruby 1. 9.1 이 삭제 되었거나 설치되지 않았을 때 발생 한다. (※Ubuntu 14.04에서는 ruby 1.9.1가 기본 설치 되어 있다.) |

1. cf-release를 배포하는 중 오류가 발생하는 경우

|  |
| --- |
| #<다운로드 디렉토리>/bosh-lite-package/manifest/cf.yml 을 복사하여 사용한다. |

1. Cf 로그인시 CLI 버전으로 인해 오류가 발생하는 경우

|  |
| --- |
| #cf 확인   * $ cf –version 실행 * 설치한 버전과 다른 버전이 출력되는 경우 * $ echo $PATH 실행, 출력되는 경로상에 다른 cf 파일이 있는지 확인한다. (※ruby/bin에도 cf 파일이 존재한다.) * 설치하지 않은 경우, cf CLI를 설치한다. ([4.3.5 참조](#_CF_배포_확인)) |

# Cloud Foundary Diego 설정

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

## ***배포한 Diego 설정***

1. CF 타겟 지정

|  |
| --- |
| **$ cf api --skip-ssl-validation https://api.10.244.0.34.xip.io**  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf api --skip-ssl-validation https://api.10.244.0.34.xip.io  Setting api endpoint to https://api.10.244.0.34.xip.io...  OK  API endpoint: https://api.10.244.0.34.xip.io (API version: 2.29.0)  User: admin  No org or space targeted, use 'cf target -o ORG -s SPACE' |

1. CF 로그인

|  |
| --- |
| **$ cf auth admin admin**  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf auth admin admin  API endpoint: https://api.10.244.0.34.xip.io  Authenticating...  OK  Use 'cf target' to view or set your target org and space |

1. 조직 작성

|  |
| --- |
| **$ cf create-org *diego***  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf create-org diego  Creating org diego as admin...  OK  TIP: Use 'cf target -o diego' to target new org |

* 밑줄친 곳은 변경가능하다.

1. 조직 지정

|  |
| --- |
| **$ cf target -o *diego***  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf target -o diego  API endpoint: https://api.10.244.0.34.xip.io (API version: 2.29.0)  User: admin  Org: diego  Space: No space targeted, use 'cf target -s SPACE' |

* 밑줄친 곳은 변경가능하다.

1. Space 생성

|  |
| --- |
| **$ cf create-space *diego***  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf create-space diego  Creating space diego in org diego as admin...  OK  Assigning role SpaceManager to user admin in org diego / space diego as admin...  OK  Assigning role SpaceDeveloper to user admin in org diego / space diego as admin...  OK  TIP: Use 'cf target -o diego -s diego' to target new space |

* 밑줄친 곳은 변경가능하다.

1. Space 지정

|  |
| --- |
| **$ cf target -s *diego***  cloud4u@XPS-15-9530:~/workspace/diego-release$ cf target -s diego    API endpoint: https://api.10.244.0.34.xip.io (API version: 2.29.0)  User: admin  Org: diego  Space: diego |

* 밑줄친 곳은 변경가능하다.

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