

전자정부표준프레임워크 기반의 Open PaaS 개발

Bosh Lite 설치 가이드

|  |  |  |
| --- | --- | --- |
| 작성자 | 김태형 (인) | 2015.7.29 |
| 검토자 | 문철호 (인) | 2015.8.07 |
| 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 검토회의 결과 반영 (OpenPaaS Container 설치 및 배포 추가) | 김태형 | 김기현 |
| 0.9 | 2015.8.27 | 설치패키지 구성 및 명칭변경 | 김태형 | 김기현 |
| 1.0 | 2016.1.06 | OpenPaaS Controller 및 OpenPaaS Container 설치 및 배포 수정 | 김태형 | 김기현 |
| 1.0 | 2016.2.03 | 오탈자 수정 | 김준형 | 김기현 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 문서 개요 5](#_Toc440544876)

[*1.1.* *목적* 5](#_Toc440544877)

[*1.2.* *범위* 5](#_Toc440544878)

[*1.3.* *참고자료* 5](#_Toc440544879)

[2. BOSH Lite 설치 6](#_Toc440544880)

[*2.1.* *설치전 준비사항* 6](#_Toc440544881)

[*2.2.* *OpenPaaS 설치패키지 다운로드* 6](#_Toc440544882)

[*2.3.* *RUBY 설치* 7](#_Toc440544883)

[*2.4.* *가상 환경 구성* 10](#_Toc440544884)

[*2.5.* *BOSH CLI 설치* 11](#_Toc440544885)

[*2.6.* *BOSH Lite 설치* 12](#_Toc440544886)

[*2.7.* *로컬 가상 머신의 IP 변경* 18](#_Toc440544887)

[*2.8.* *BOSH Lite 설치 Troubleshooting* 19](#_Toc440544888)

[3. OpenPaaS Container 설치, 배포 및 설정 22](#_Toc440544889)

[*3.1.* *Release 업로드* 22](#_Toc440544890)

[*3.2.* *BOSH Stemcell 업로드* 22](#_Toc440544891)

[*3.3.* *OpenPaaS Controller 배포* 23](#_Toc440544892)

[*3.4.* *OpenPaaS Container 배포* 26](#_Toc440544893)

[4. OpenPaaS Container 설정 28](#_Toc440544894)

[*4.1.* *Container 배포 확인* 28](#_Toc440544895)

[*4.2.* *OpenPaaS Container설정 Troubleshooting* 29](#_Toc440544896)

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
* OpenPaaS Container 설치, 배포 및 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 OpenPaaS를 릴리즈하는 것까지 기술하였다.

## ***참고자료***

본 문서는 Cloud Foundry의 BOSH Document를 참고로 작성하였다.

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

OpenPaaS Container 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.

* 환경구성에 필요한 OpenPaaS 설치패키지
  + OpenPaaS-Env
  + OpenPaaS-Dev-Tools
* 메모리: 6GB의 메모리를 가진 가상 머신을 구동할 수 있는 여유 메모리
* 디스크: 최소 80GB 이상의 여유 디스크
* 위의 조건을 충족하는 Linux 또는 OSX 환경 (BOSH는 Window 환경을 지원하지 않는다. 참조: https://github.com/cloudfoundry-community/bosh-lite-demo)
* 가상 머신에서의 Bosh-lite의 설치는 지원하지 않는다.

## ***OpenPaaS 설치패키지 다운로드***

다음의 OpenPaaS 설치패키지를 다운받는다.

|  |  |  |  |
| --- | --- | --- | --- |
| **구성 항목** | | | **설명** |
| **구분** | **디렉토리** | **파일명** |
| warden | OpenPaaS-Env | bosh-init-0.0.81-linux-amd64 | BOSH init 실행 파일 |
| bosh\_gem\_package.tar | BOSH CLI 설치 패키지 |
| ruby-2.1.6.tar.gz | Ruby 2.1.6 버전 |
| OpenPaaS-Env/bosh\_env\_packages | 디렉토리 전체 | BOSH 실행 환경 설치 패키지 |
| OpenPaaS-IaaS-Controller | bosh-233.tgz | BOSH v233 릴리즈 |
| OpenPaaS-Controller | cf-release-226.tgz | Cf v226 릴리즈 |
| OpenPaaS-Container | diego-release-0.1442.0.tgz | Diego v0.1442.0 릴리즈 |
| garden-linux-release-0.329.0.tgz | Garden-linux v0.329.0 릴리즈 |
| etcd-release-20.tgz | Etcd v20 릴리즈 |
| OpenPaaS-Stemcells | bosh-stemcell-3147-warden-boshlite-ubuntu-trusty-go\_agent.tgz | boshlite v3147  스템셀 이미지 |
| OpenPaaS-Deployment | openpaas-cf-lite-1.0.yml | cf 배포 Manifest Sample 파일 |
| openpaas-diego-lite-1.0.yml | diego 배포 Manifest Samle 파일 |

## ***RUBY 설치***

Ruby 설치 절차는 다음과 같다.

1. Bosh Dependency를 설치한다.

* **온라인인 경우(권장)**

|  |  |  |  |
| --- | --- | --- | --- |
| * **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** | |

* **오프라인인 경우**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # 설치환경이 ubuntu (trusty 버전)인 경우 다음의 스크립트를 실행한다.  **$ cd *<다운로드 디렉토리>/OpenPaaS-Env/bosh\_env\_packages***  **$ chmod +x ./bosh-packages-install.sh**  **$ ./bosh-packages-install.sh**   * **BOSH 설치 패키지의 구성은 다음과 같다.**  |  |  |  | | --- | --- | --- | | No | 패키지명 | 패키지 파일명 (Ubuntu trusty amd64) | | 1 | libcomerr2 | libcomerr2\_1.42.9-3ubuntu1.3\_amd64.deb | | 2 | libssl1.0.0 | libssl1.0.0\_1.0.1f-1ubuntu2.16\_amd64.deb | | 3 | libkrb5support0 | libkrb5support0\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 4 | libk5crypto3 | libk5crypto3\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 5 | libkrb5-3 | libkrb5-3\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 6 | libgssapi-krb5-2 | libgssapi-krb5-2\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 7 | libxml2 | libxml2\_2.9.1+dfsg1-3ubuntu4.6\_amd64.deb | | 8 | libgssrpc4 | libgssrpc4\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 9 | libkadm5clnt-mit9 | libkadm5clnt-mit9\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 10 | libkdb5-7 | libkdb5-7\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 11 | libkadm5srv-mit9 | libkadm5srv-mit9\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 12 | mysql-common | mysql-common\_5.5.46-0ubuntu0.14.04.2\_all.deb | | 13 | libmysqlclient18 | libmysqlclient18\_5.5.46-0ubuntu0.14.04.2\_amd64.deb | | 14 | libxslt1.1 | libxslt1.1\_1.1.28-2build1\_amd64.deb | | 15 | libyaml-0 | libyaml-0-2\_0.1.4-3ubuntu3.1\_amd64.deb | | 16 | openssl\_1.0.1f-1ubuntu2.16 | openssl\_1.0.1f-1ubuntu2.16\_amd64.deb | | 17 | libstdc++-4.8-dev | libstdc++-4.8-dev\_4.8.4-2ubuntu1~14.04\_amd64.deb | | 18 | g++-4.8 | g++-4.8\_4.8.4-2ubuntu1~14.04\_amd64.deb | | 19 | g++ | g++\_4%3a4.8.2-1ubuntu6\_amd64.deb | | 20 | libdpkg-perl | libdpkg-perl\_1.17.5ubuntu5.5\_all.deb | | 21 | dpkg-dev | dpkg-dev\_1.17.5ubuntu5.5\_all.deb | | 22 | build-essential | build-essential\_11.6ubuntu6\_amd64.deb | | 23 | libfakeroot | libfakeroot\_1.20-3ubuntu2\_amd64.deb | | 24 | fakeroot | fakeroot\_1.20-3ubuntu2\_amd64.deb | | 25 | comerr-dev | comerr-dev\_2.1-1.42.9-3ubuntu1.3\_amd64.deb | | 26 | krb5-multidev | krb5-multidev\_1.12+dfsg-2ubuntu5.2\_amd64.deb | | 27 | libalgorithm-diff-perl | libalgorithm-diff-perl\_1.19.02-3\_all.deb | | 28 | libalgorithm-diff-xs-perl | libalgorithm-diff-xs-perl\_0.04-2build4\_amd64.deb | | 29 | libalgorithm-merge-perl | libalgorithm-merge-perl\_0.08-2\_all.deb | | 30 | libtinfo-dev | libtinfo-dev\_5.9+20140118-1ubuntu1\_amd64.deb | | 31 | libreadline6-dev | libreadline6-dev\_6.3-4ubuntu2\_amd64.deb | | 32 | libfile-fcntllock-perl | libfile-fcntllock-perl\_0.14-2build1\_amd64.deb | | 33 | zlib1g-dev | zlib1g-dev\_1%3a1.2.8.dfsg-1ubuntu1\_amd64.deb | | 34 | libmysqlclient-dev | libmysqlclient-dev\_5.5.46-0ubuntu0.14.04.2\_amd64.deb | | 35 | libpq5 | libpq5\_9.3.10-0ubuntu0.14.04\_amd64.deb | | 36 | libssl-dev | libssl-dev\_1.0.1f-1ubuntu2.16\_amd64.deb | | 37 | libpq-dev | libpq-dev\_9.3.10-0ubuntu0.14.04\_amd64.deb | | 38 | libsqlite3-dev | libsqlite3-dev\_3.8.2-1ubuntu2.1\_amd64.deb | | 39 | libssl-doc | libssl-doc\_1.0.1f-1ubuntu2.16\_all.deb | | 40 | libxml2-dev | libxml2-dev\_2.9.1+dfsg1-3ubuntu4.6\_amd64.deb | | 41 | libxslt1-dev | libxslt1-dev\_1.1.28-2build1\_amd64.deb | | 42 | sqlite3 | sqlite3\_3.8.2-1ubuntu2.1\_amd64.deb | | 43 | Zlibc | zlibc\_0.9k-4.1\_amd64.deb | | 44 | libyaml-dev | libyaml-dev\_0.1.4-3ubuntu3.1\_amd64.deb | | 45 | libruby1.9.1 | libruby1.9.1\_1.9.3.484-2ubuntu1.2\_amd64.deb | | 46 | ruby1.9.1-dev | ruby1.9.1-dev\_1.9.3.484-2ubuntu1.2\_amd64.deb | | 47 | ruby-dev | ruby-dev\_1%3a1.9.3.4\_all.deb | | 48 | ruby | ruby\_1%3a1.9.3.4\_all.deb | | 49 | ruby1.9.1 | ruby1.9.1\_1.9.3.484-2ubuntu1.2\_amd64.deb |  * CentOS, RHEL 또는 OSX의 경우는 표에 해당하는 패키지를 설치한다. |

1. Ruby 설치

|  |
| --- |
| **$ tar -xvzf *<다운로드 디렉토리>/OpenPaaS-Env/ruby-2.1.6.tar.gz***  **$ cd ruby-2.1.6/**  **$ ./configure**  **$ make**  **$ sudo make install** |

* Ruby를 컴파일하여 설치할 경우 기본으로 /usr/local/lib에 설치한다.
* 온라인의 경우, rvm 또는 rbenv 등을 이용하여 ruby를 설치할 수 있다. 단, ruby 1.9.3버전 이상을 설치하도록 한다.

1. 설치 확인

|  |
| --- |
| **# Ruby 버전 확인 (터미널창을 새로 열어서 확인한다.)**  **$ ruby -v**  $ ruby -v  ruby 2.1.6p336 (2015-04-13 revision 50298) [x86\_64-linux]  **# Ruby 환경설정 확인**  **$ gem env**  $ gem env  RubyGems Environment:  - RUBYGEMS VERSION: 2.2.3  - RUBY VERSION: 2.1.6 (2015-04-13 patchlevel 336) [x86\_64-linux]  - **INSTALLATION DIRECTORY: /usr/local/lib/ruby/gems/2.1.0 <- gem 설치 디렉토리**  - RUBY EXECUTABLE: /usr/local/bin/ruby  - EXECUTABLE DIRECTORY: /usr/local/bin  - SPEC CACHE DIRECTORY: /home/ubuntu/.gem/specs  - RUBYGEMS PLATFORMS:  - ruby  - x86\_64-linux  - GEM PATHS:  - /usr/local/lib/ruby/gems/2.1.0  - /home/ubuntu/.gem/ruby/2.1.0  - GEM CONFIGURATION:  - :update\_sources => true  - :verbose => true  - :backtrace => false  - :bulk\_threshold => 1000  - REMOTE SOURCES:  - https://rubygems.org/  - SHELL PATH:  - /usr/local/sbin  - /usr/local/bin  - /usr/sbin  - /usr/bin  - /sbin  - /bin  - /usr/games  - /usr/local/games |

## ***가상 환경 구성***

BOSH Lite 설치를 위해 Vagrant와 Virtualbox를 설치하여 가상 환경을 구성한다.

1. Vagrant와 Virtualbox 및 Virtualbox 확장팩 설치

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # 설치환경이 ubuntu (trusty 버전)인 경우 다음의 스크립트를 실행한다.  **$ cd *<다운로드 디렉토리>/OpenPaaS-Env/bosh\_env\_packages***  **$ chmod +x ./env-packages-install.sh**  **$ ./env-packages-install.sh**   * **가상 환경 설치 패키지의 구성은 다음과 같다.**  |  |  |  | | --- | --- | --- | | No | 패키지명 | 패키지 파일명 (Ubuntu trusty amd64) | | 1 | libsdl1.2debian | libsdl1.2debian\_1.2.15-8ubuntu1\_amd64.deb | | 2 | virtualbox-5.0 | virtualbox-5.0\_5.0.12-104815~Ubuntu~trusty\_amd64.deb | | 3 | vagrant | vagrant\_1.7.4\_x86\_64.deb | | 4 | unzip | unzip\_6.0-4ubuntu2.5\_amd64.deb |  * CentOS, RHEL 또는 OSX의 경우는 표에 해당하는 패키지를 설치한다.   # 설치 스크립트 이외의 방법으로 패키지를 설치한 경우, Virtualbox 확장팩을 다음과 같이 수동으로 설치한다.  **$ sudo VBoxManage extpack install *<다운로드 디렉토리>/OpenPaaS-Env/ bosh\_env\_packages /env\_packages/Oracle\_VM\_VirtualBox\_Extension\_Pack-5.0.12-104815.vbox-extpack*** |

* VirtualBox 확장팩 설치한 Virtualbox의 버전과 동일한 버전을 설치해야 한다.
* VirtualBox 확장팩은 모든 OS에 공통으로 설치할 수 있다.

1. VirtualBox 설치 확인

|  |
| --- |
| **$** **VBoxManage --version**  $ VBoxManage --version  5.0.12r104815 |

1. Vagrant 설치 확인

|  |
| --- |
| **$ vagrant --version**  $ vagrant --version  Vagrant 1.7.4 |

## ***BOSH CLI 설치***

BOSH CLI의 설치 절차는 다음과 같다.

1. Bosh\_cli 설치

* 온라인의 경우

|  |
| --- |
| #bosh\_cli 패키지 설치  **$ sudo gem install bosh\_cli**  *※ gem 설치 디렉토리 소유자가 로그인한 사용자와 동일한 경우는 ‘sudo’를 사용하지 않는다.* |

* 오프라인의 경우

|  |
| --- |
| **#bosh\_cli 의존 패키지 압축해제**  **$ sudo tar -xvf *<다운로드 디렉토리>/OpenPaaS-Env/bosh\_gem\_package.tar* -C *<gem 설치 디렉토리>/cache***  *※ gem 설치 디렉토리는 ‘gem env’를 실행하여 확인한다. (예: /usr/local/lib/ruby/gems/2.1.0)*  **#압축해제한 gem 설치**  **$ sudo gem install --force --local --no-ri --no-rdoc *<gem 설치 디렉토리>*/cache/\*.gem**  *※ gem 설치 디렉토리 소유자가 로그인한 사용자와 동일한 경우는 ‘sudo’를 사용하지 않는다.*  **#참고: 기존에 설치한 gem 삭제**  **$ sudo gem uninstall -aIx** |

1. Bosh 설치 확인

|  |
| --- |
| **$ bosh**  $ /OpenPaaS-Env/virtualbox-5.0$ bosh  BOSH CLI helps you manage your BOSH deployments and releases.  Usage: bosh [<options>] <command> [<args>]  -c, --config FILE Override configuration file. Also can be overridden by BOSH\_CONFIG environment variable. Defaults to $HOME/.bosh\_config. Override precedence is command-line option, then environment variable, then home directory.  --parallel MAX Sets the max number of parallel downloads  --[no-]color Toggle colorized output  -v, --verbose Show additional output  -q, --quiet Suppress all output  -n, --non-interactive Don't ask for user input  -N, --no-track Return Task ID and don't track  -P, --poll INTERVAL Director task polling interval  -t, --target URL Override target  -u, --user USER Override username  -p, --password PASSWORD Override password  -d, --deployment FILE Override deployment  -h, --help here you go  add blob <local\_path> [<blob\_dir>]  Add a local file as BOSH blob  alias <name> <command>  Create an alias <name> for command <command>  <<이하 생략>> |

## ***BOSH Lite 설치***

BOSH Lite를 설치하는 절차는 다음과 같다.

1. BOSH Lite 설치

|  |
| --- |
| **# 설치 디렉토리 생성**  **$ mkdir -p ~/workspace**  # Bosh lite 설치  **$ cd ~/workspace**  **$ unzip *<다운로드 디렉토리>/OpenPaaS-Dev-Tools/bosh-lite-master.zip***  # 디렉토리명 변경  **$ mv bosh-lite-master/ bosh-lite** |

1. Vagrantfile 수정

* ***온라인의 경우, Vagrantfile의 수정이 필요없다.***
* 오프라인의 경우

|  |
| --- |
| **$ cd bosh-lite**  **$ vi Vagrantfile**  Vagrant.configure('2') do |config|  config.vm.box = 'cloudfoundry/bosh-lite'  config.vm.provider :virtualbox do |v, override|  override.vm.box\_version = '9000.39.0' # 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  **override.vm.network :private\_network, ip: '192.168.50.4', id: :local ## 추가**  **override.vm.network :public\_network ## 추가**  end  config.vm.provider :aws do |v, override|  override.vm.box\_version = '9000.39.0' # 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"  # Following minimal config is for Vagrant 1.7 since it loads this file before downloading the box.  # (Must not fail due to missing ENV variables because this file is loaded for all providers)  v.access\_key\_id = ENV['BOSH\_AWS\_ACCESS\_KEY\_ID'] || ''  v.secret\_access\_key = ENV['BOSH\_AWS\_SECRET\_ACCESS\_KEY'] || ''  v.ami = ''  end  end |

1. Vagrant에 BOSH Lite box 추가

* **온라인의 경우, Vagrant up을 실행할 때 자동으로 BOSH Lite box를 추가한다.**
* 오프라인의 경우

|  |
| --- |
| **$ vagrant box add cloudfoundry/bosh-lite** ***<다운로드 디렉토리>/OpenPaaS-Env/vagrantbox/bosh-lite.box*** |

* 추가하는 box명은 cloudfoundry/bosh-lite 이다.
* Vagrantfile의 config.vm.box에서 선언한 이름과 동일해야 한다.
* 설치패키지의 override.vm.box\_version은 ‘9000.85.0’이다.

1. BOSH Lite 설치

* 온라인의 경우

|  |
| --- |
| # 가상 머신 생성  **$ cd ~/workspace/bosh-lite**  **$ vagrant up --provider=virtualbox**  $ vagrant up  Bringing machine 'default' up with 'virtualbox' provider...  ==> default: Box 'cloudfoundry/bosh-lite' could not be found. Attempting to find and install...  default: Box Provider: virtualbox  default: Box Version: 9000.45.0  ==> default: Loading metadata for box 'cloudfoundry/bosh-lite'  default: URL: https://atlas.hashicorp.com/cloudfoundry/bosh-lite  ==> default: Adding box 'cloudfoundry/bosh-lite' (v9000.45.0) for provider: virtualbox  default: Downloading: https://atlas.hashicorp.com/cloudfoundry/boxes/bosh-lite/versions/9000.45.0/providers/virtualbox.box  ==> default: Successfully added box 'cloudfoundry/bosh-lite' (v9000.45.0) for 'virtualbox'!  ==> default: Importing base box 'cloudfoundry/bosh-lite'...  ==> default: Matching MAC address for NAT networking...  ==> default: Checking if box 'cloudfoundry/bosh-lite' is up to date...  ==> default: Setting the name of the VM: bosh-lite\_default\_1441161503381\_13241  ==> default: Clearing any previously set network interfaces...  ==> default: Preparing network interfaces based on configuration...  default: Adapter 1: nat  default: Adapter 2: hostonly  ==> default: Forwarding ports...  default: 22 => 2222 (adapter 1)  ==> default: Running 'pre-boot' VM customizations...  ==> default: Booting VM...  ==> default: Waiting for machine to boot. This may take a few minutes...  default: SSH address: 127.0.0.1:2222  default: SSH username: vagrant  default: SSH auth method: private key  default: Warning: Connection timeout. Retrying...  default:  default: Vagrant insecure key detected. Vagrant will automatically replace  default: this with a newly generated keypair for better security.  default:  default: Inserting generated public key within guest...  default: Removing insecure key from the guest if it's present...  default: Key inserted! Disconnecting and reconnecting using new SSH key...  ==> default: Machine booted and ready!  ==> default: Checking for guest additions in VM...  ==> default: Setting hostname...  ==> default: Configuring and enabling network interfaces...  ==> default: Mounting shared folders...  default: /vagrant => /home/cloud4u/workspace/bosh-lite |

* 오프라인 또는 WI-FI로 연결한 경우

|  |
| --- |
| # 네트워크 설정 확인  **$ ifconfig**  #가상 머신 생성  **$ cd ~/workspace/bosh-lite**  **$ vagrant up --provider=virtualbox**  $ vagrant upBringing machine 'default' up with 'virtualbox' provider...==> default: Importing base box 'cloudfoundry/bosh-lite'...==> default: Matching MAC address for NAT networking...==> default: Setting the name of the VM: bosh-lite\_default\_1438745131006\_83536==> default: Clearing any previously set network interfaces...==> default: Available bridged network interfaces:***# 설치한 환경에 따라 아래의 network interface의 예제는 다르게 나타난다. 1) eth1 # 오프라인인 경우 선택2) wlan0 # WI-FI인 경우 선택***3) virbr0==> default: When choosing an interface, it is usually the one that is==> default: being used to connect to the internet. default: Which interface should the network bridge to? ***1 # 설치 환경을 오프라인으로 가정한 경우***  ==> default: Preparing network interfaces based on configuration... default: Adapter 1: nat default: Adapter 2: hostonly default: Adapter 3: bridged==> default: Forwarding ports... default: 22 => 2222 (adapter 1)==> default: Booting VM...==> default: Waiting for machine to boot. This may take a few minutes... default: SSH address: 127.0.0.1:2222 default: SSH username: vagrant default: SSH auth method: private key default: Warning: Connection timeout. Retrying...==> default: Machine booted and ready!==> default: Checking for guest additions in VM... default: The guest additions on this VM do not match the installed version of default: VirtualBox! In most cases this is fine, but in rare cases it can default: prevent things such as shared folders from working properly. If you see default: shared folder errors, please make sure the guest additions within the default: virtual machine match the version of VirtualBox you have installed on default: your host and reload your VM.  default: default: Guest Additions Version: 4.3.18 default: VirtualBox Version: 5.0==> default: Configuring and enabling network interfaces...==> default: Mounting shared folders... default: /vagrant => ~/workspace/bosh-lite  # 가상 머신의 ‘/vagrant’ 디렉토리 권한 확인  **$ cd ~/workspace/bosh-lite**  **$ vagrant ssh**  **$ ll /**  $ ll /  total 97  drwxr-xr-x 24 root root 4096 Jan 13 08:50 ./  drwxr-xr-x 24 root root 4096 Jan 13 08:50 ../  drwxr-xr-x 2 root root 4096 Dec 25 11:51 bin/  drwxr-xr-x 4 root root 1024 Dec 25 11:50 boot/  drwxr-xr-x 2 root root 4096 Dec 25 11:57 .bundle/  drwxr-xr-x 16 root root 9160 Jan 13 08:51 dev/  drwxr-xr-x 85 root root 4096 Jan 13 08:51 etc/  drwxr-xr-x 4 root root 4096 Dec 25 11:50 home/  lrwxrwxrwx 1 root root 33 Dec 25 11:48 initrd.img -> boot/initrd.img-3.19.0-42-generic  drwxr-xr-x 22 root root 4096 Dec 25 11:46 lib/  drwxr-xr-x 2 root root 4096 Dec 25 11:45 lib64/  drwx------ 2 root root 16384 Dec 25 11:45 lost+found/  drwxr-xr-x 3 root root 4096 Dec 25 11:45 media/  drwxr-xr-x 2 root root 4096 Apr 10 2014 mnt/  drwxr-xr-x 4 root root 4096 Dec 25 11:50 opt/  dr-xr-xr-x 162 root root 0 Jan 13 08:50 proc/  drwx------ 2 root root 4096 Dec 25 11:51 root/  drwxr-xr-x 17 root root 620 Jan 13 08:51 run/  drwxr-xr-x 2 root root 12288 Dec 25 11:51 sbin/  drwxr-xr-x 2 root root 4096 Aug 5 05:11 srv/  dr-xr-xr-x 13 root root 0 Jan 13 08:50 sys/  drwxrwx--- 5 root vcap 4096 Jan 13 08:51 tmp/  drwxr-xr-x 10 root root 4096 Dec 25 11:45 usr/  **drwxrwxrwx 1 vagrant vagrant 4096 Jan 13 08:47 vagrant/ # 권한 확인**  drwxr-xr-x 12 root root 4096 Dec 25 11:51 var/  lrwxrwxrwx 1 root root 30 Dec 25 11:48 vmlinuz -> boot/vmlinuz-3.19.0-42-generic  # ‘/vagrant’의 권한이 ‘drwxrwxrwx’가 아닌 경우 다음의 처리를 실행  # 1. 가상 머신 재기동  **$ cd ~/workspace/bosh-lite**  **$ vagrant ssh**  **$ sudo reboot**  # 2. 가상 머신의 ‘/vagrant’ 디렉토리에 쓰기권한 설정  **$ vagrant ssh**  **$ sudo chmod 777 -R /vagrant** |

* Vagrantfile이 있는 곳에서 vagrant up을 실행해야 한다.

1. BOSH Lite 로그인

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

* Bosh lite의 기본 IP는 192.168.50.4 이다.
* login 기본 사용자와 패스워드는 admin/admin
* 붉은 글씨는 입력항목

1. Route 추가

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ sudo bin/add-route** |

1. 생성한 Bosh lite 가상 머신 확인

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ vagrant ssh**  $ vagrant ssh  Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-45-generic x86\_64)  \* Documentation: https://help.ubuntu.com/  Last login: Wed Aug 5 07:27:34 2015 from 192.168.50.1  vcap@agent-id-bosh-0:~$ |

## ***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.

|  |
| --- |
| **#가상 머신 삭제**  **$ cd ~/workspace/bosh-lite**  **$ vagrant destroy**  **위의 방법으로 가상 머신을 삭제할 수 없는 경우, Virtualbox에서 해당 가상 머신을 삭제한다.**  **#가상 머신 삭제 확인**  **$ vagrant global-status --prune**  $ ~/workspace/bosh-lite$ vagrant global-status --prune  id name provider state directory  --------------------------------------------------------------------  There are no active Vagrant environments on this computer! Or,  you haven't destroyed and recreated Vagrant environments that were  started with an older version of Vagrant. |

1. Bosh의 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 vagrant halt 또는 virtualbox에서 power off 방식으로 종료하면 가상 머신에 오류가 발생 할 수 있다. Virtualbox 등에서 가상 머신을 <Save state>모드로 종료 한다. 또는 다음을 실행하여 VM을 재생성 할 수 있다.

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

1. Vagrant up 실행시 VirtualBox에 가상 머신이 등록되지 않는 경우

|  |
| --- |
| # 가끔 virtualbox 확장팩 버전과 bosh lite box의 확장팩 버전이 일치 하지 않는 경우 발생한다.  # 설치한 가상 머신을 삭제하고 서버를 재기동 한 후, 다시 가상 머신을 설치하면 해결 된다.  **# 가상 머신 삭제**  **$ vagrant destroy**  # 서버 재시작  **$ sudo reboot**  # 가상 머신 설치  **$ cd ~/workspace/bosh-lite**  **$ vagrant up --provider=virtualbox** |

# OpenPaaS Container 설치, 배포 및 설정

본 절에서는 BOSH Lite에 OpenPaaS Container를 릴리즈하는 절차를 기술한다. Container는 OpenPaaS의 새로운 런타임 아키텍처로 이전 버전의 OpenPaaS의 DEAs와 Health Manager를 대체 한다. 자세한 내용은 다음 사이트를 참조 한다.

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

* 환경구성에 필요한 OpenPaaS 설치패키지
  + OpenPaaS-Stemcells
  + OpenPaaS-Controller
  + OpenPaaS-Container

## ***Release 업로드***

1. OpenPaaS controller를 bosh에 업로드

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ bosh upload release <다운로드 디렉토리>/OpenPaaS-Controller/cf-release-226.tgz** |

1. OpenPaaS Container Release 업로드

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ bosh upload release <다운로드 디렉토리>/OpenPaaS-Container/diego-release-0.1442.0.tgz** |

1. garden-linux 업로드

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ bosh upload release <다운로드 디렉토리>/OpenPaaS-Container/garden-linux-release-0.329.0.tgz** |

1. etcd Release 업로드

|  |
| --- |
| **$ cd ~/workspace/bosh-lite**  **$ bosh upload release <다운로드 디렉토리>/OpenPaaS-Container/etcd-release-20.tgz** |

1. release 업로드 확인

|  |
| --- |
| **$ bosh releases** |

## ***BOSH Stemcell 업로드***

1. BOSH에 Warden Stemcell 업로드

|  |
| --- |
| **$ bosh upload stemcell *<다운로드 디렉토리>/OpenPaaS-Stemcells/bosh-stemcell-3147-warden-boshlite-ubuntu-trusty-go\_agent.tgz*** |

1. Stemcell 업로드 확인

|  |
| --- |
| **$ bosh stemcells** |

## ***OpenPaaS Controller 배포***

OpenPaaS Controller를 배포한다.

1. Bosh 상태 확인

|  |
| --- |
| **$ bosh status**  $ ~/workspace/bosh-lite$ bosh status  Config  /home/cloud4u/.bosh\_config  Director  Name Bosh Lite Director  URL https://192.168.50.4:25555  Version 1.3062.0 (00000000)  User admin  UUID **cf5f29da-51c9-433f-8c1f-9ae0536d4778**  CPI cpi  dns disabled  compiled\_package\_cache enabled (provider: local)  snapshots disabled  Deployment  not set |

1. Controller.yml 파일을 OpenPaaS환경에 맞게 수정

|  |
| --- |
| **$ cp *<다운로드 디렉토리>/OpenPaaS-Deployment/openpaas-cf-lite-1.0.yml* ~/workspace**  **$ vi ~/workspace/openpaas-cf-lite-1.0.yml**  compilation:  cloud\_properties:  name: random  network: cf1  reuse\_compilation\_vms: true  workers: 6  director\_uuid:  **cf5f29da-51c9-433f-8c1f-9ae0536d4778 🡪 조회한 bosh director uuid로 변경**  jobs:  - instances: 1  name: consul\_z1  networks:  - name: cf1  static\_ips:  - 10.244.0.54  <<후략>> |

1. Openpaas-cf-lite-1.0.yml 파일을 BOSH Lite에 deployment

|  |
| --- |
| **$ bosh deployment ~/workspace/openpaas-cf-lite-1.0.yml** |

1. OpenPaaS Controller 배포

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

1. 배포한 release 확인

|  |
| --- |
| **$** **bosh vms**  $ bosh vms  Acting as user 'admin' on 'Bosh Lite Director'  Deployment `openpaas-warden'  Director task 17  Task 17 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 | 10.244.0.142 |  | 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 | 10.244.0.138 |  | 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 |  | runner\_z1/0 | running | runner\_z1 | 10.244.0.26 |  | uaa\_z1/0 | running | medium\_z1 | 10.244.0.130 |  +------------------------------------+---------+---------------+--------------+  VMs total: 12 |

## ***OpenPaaS Container 배포***

OpenPaaS Container Release를 배포한다.

1. Container.yml 파일을 OpenPaaS환경에 맞게 수정

|  |
| --- |
| **$ cp <다운로드 디렉토리>/OpenPaaS-Deployment/openpaas-diego-lite-1.0 ~/workspace**  **$ vi ~/workspace/openpaas-diego-lite-1.0**  compilation:  cloud\_properties: {}  network: diego1  reuse\_compilation\_vms: true  workers: 6  director\_uuid:  **cf5f29da-51c9-433f-8c1f-9ae0536d4778 🡪 조회한 bosh director uuid로 변경**  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  <<후략>> |

1. Openpaas-diego-lite-1.0.yml 파일을 BOSH Lite에 deployment

|  |
| --- |
| **$ bosh deployment ~/workspace/openpaas-diego-lite-1.0.yml** |

1. OpenPaaS Controller 배포

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

1. 배포한 release 확인

|  |
| --- |
| **$** **bosh vms**  $ bosh vms  Acting as user 'admin' on 'Bosh Lite Director'  Deployment `openpaas-warden'  Director task 19  Task 19 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 | 10.244.0.142 |  | 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 | 10.244.0.138 |  | 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 |  | runner\_z1/0 | running | runner\_z1 | 10.244.0.26 |  | uaa\_z1/0 | running | medium\_z1 | 10.244.0.130 |  +------------------------------------+---------+---------------+--------------+  VMs total: 12  Deployment `openpaas-warden-container'  Director task 20  Task 20 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 |

# OpenPaaS Container 설정

본 절에서는 BOSH Lite에서 OpenPaaS Container를 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>

* 환경구성에 필요한 OpenPaaS 설치패키지
  + OpenPaaS-Env

## ***Container 배포 확인***

1. Cf CLI 설치

|  |
| --- |
| **$ tar -C /usr/local/bin -xzf *<다운로드 디렉토리>/OpenPaaS-Env/cf-cli\_6.14.1\_linux\_x86-64.tgz*** |

1. Cf 플러그인 설치

|  |
| --- |
| **$ chmod +x *<다운로드 디렉토리>/OpenPaaS-Env/diego-enabler\_linux\_amd64***  **$ cf install-plugin *<다운로드 디렉토리>/OpenPaaS-Env/diego-enabler\_linux\_amd64*** |

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>*** |

## ***OpenPaaS Container설정 Troubleshooting***

1. CF 로그인에서 오류가 발생하는 경우

|  |
| --- |
| # 오프라인 상태에서 OpenPaaS Container를 타겟 지정 할 경우, 실패한다.  네트워크에 연결하여 온라인 상태로 OpenPaaS Container를 타겟 지정하거나, 별도 DNS서버를 구축하여 DNS서버와 연결한 후, OpenPaaS Container를 타겟 지정한다. |

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