

전자정부표준프레임워크 기반의 Open PaaS 개발

BOSH 설치 가이드

|  |  |  |
| --- | --- | --- |
| 작성자 | 문철호 (인) | 2015.8.6 |
| 검토자 | 김태형 (인) | 2015.8.7 |
| 승인자 | 김기현 (인) | 2015.8.10 |

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

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-1) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 0.1 | 2015.8.6 | 최초작성 | 문철호 | 김기현 |
| 1.0 | 2016.1.6 | Bosh-int 설치 내용 추가 | 김태형 | 김기현 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 문서 개요 2](#_Toc439869829)

[*1.1.* *목적* 2](#_Toc439869830)

[*1.2.* *범위* 2](#_Toc439869831)

[*1.3.* *참고자료* 2](#_Toc439869832)

[2. Basic BOSH Workflow 2](#_Toc439869833)

[3. Install BOSH CLI 2](#_Toc439869834)

[*3.1.* *설치전 준비사항* 2](#_Toc439869835)

[*3.2.* *Ruby 설치* 2](#_Toc439869836)

[*3.3.* *BOSH INIT 설치* 2](#_Toc439869837)

[*3.4.* *BOSH CLI 설치* 2](#_Toc439869838)

[4. Install MicroBOSH 2](#_Toc439869839)

[*4.1.* *설치 절차* 2](#_Toc439869840)

[*4.2.* *OpenStack 환경설정* 2](#_Toc439869841)

[*4.3.* *배포 Manifest 작성* 2](#_Toc439869842)

[*4.4.* *BOSH Stemcell 다운로드* 2](#_Toc439869843)

[*4.5.* *Release 다운로드* 2](#_Toc439869844)

[*4.6.* *MicroBOSH 배포* 2](#_Toc439869845)

[5. Install BOSH 2](#_Toc439869846)

[*5.1.* *설치 절차* 2](#_Toc439869847)

[*5.2.* *BOSH Stemell 업로드* 2](#_Toc439869848)

[*5.3.* *BOSH Release 업로드* 2](#_Toc439869849)

[*5.4.* *배포 Manifest 작성* 2](#_Toc439869850)

[*5.5.* *BOSH 배포* 2](#_Toc439869851)

Executive Summary

본 문서는 BOSH에 대한 설치 가이드이다. BOSH를 설치하고 기본 동작하는 부분을 설명하였다.

본 문서는 다음과 같은 내용들을 포함한다.

* BOSH CLI 설치
* MicroBOSH 설치
* BOSH 설치

# 문서 개요

## ***목적***

클라우드 환경에 서비스 시스템을 배포할 수 있는 BOSH는 릴리즈 엔지니어링, 개발, 소프트웨어 라이프사이클 관리를 통합한 오픈소스 프로젝트이다. 본 문서는 로컬 환경에 BOSH 설치 환경 구성 및 MicroBOSH와 BOSH설치하여 기능 확인하는데 목적이 있다.

## ***범위***

본 가이드에서는 Linux 환경(Ubuntu 14.04)을 기준으로 설치환경 구성 및 BOSH CLI 설치하고, 이를 이용하여 MicroBOSH 및 BOSH를 설치하는 것을 기준으로 작성한다.

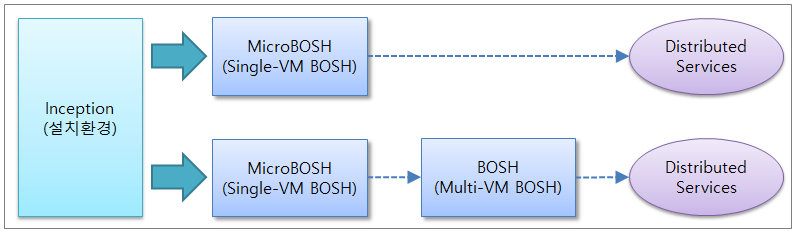
## ***참고자료***

본 문서는 Cloud Foundry의 BOSH Document를 참고로 작성하였다.

BOSH Docoument: <http://bosh.io>

# Basic BOSH Workflow

본 문서의 목적에서도 언급했듯이 BOSH는 클라우드 환경에 서비스를 배포 관리하는 소프트웨어로 BOSH자체도 클라우드에 배포되어야 하는 서비스로 설치 환경이 구성되어 있는 서버로부터 MicroBOSH만을 설치하여 운영하는 방법과 MiroBOSH 로부터 배포된 BOSH를 이용하는 방법 2가지 유형이 있으며 자신의 로컬 설치 환경에 따라 선택할 수 있다.



MicroBOSH (Single-VM BOSH)는 소프트웨어 배포 및 관리를 하나의 VM에서 수행하는 소프트웨어로 MicroBOSH를 배포하기 위해서는 Stemcell 이미지와 배포 Manifest파일이 필요하고 별도의 소프트웨어 Release는 필요로하지 않는다. (설치에 필요한 Release들은 Stemcell내부에 탑재되어 있다. )

MicroBOSH를 설치한 후에는 선택에 따라 MicroBOSH를 이용하여 BOSH (Multi-VM BOSH)를 배포하거나 또는 다른 서비스들을 배포할 수도 있다.

# Install BOSH CLI

## ***설치전 준비사항***

BOSH는 Ruby언어로 구현되어 있으므로 Ruby개발 환경 설치 후 BOSH CLI를 설치해야 한다. BOSH CLI설치 가능한 Ruby버전은 1.9.3 이상이며, 개발 환경에 이미 설치가 되어 있다면 4.1절 Ruby설치는 건너뛰면 된다.

## ***Ruby 설치***

BOSH CLI가 지원하는 Ruby버전 중 2.1.6버전을 설치 대상으로 기술하였고, 상세 설치 절차는 다음과 같다.

1. Bosh 설치에 필요한 Dependency들을 설치한다.

|  |  |  |  |
| --- | --- | --- | --- |
| # 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. Git 설치한다.

|  |
| --- |
| # git 설치  $ sudo apt-get -y install git |

1. Ruby 설치는 rbenv, rvm, source 컴파일 후 인스톨하는 3가지 방법이 있다. 본 문서에서는 rbenv를 이용하여 설치하는 방법으로 기술한다.

|  |
| --- |
| $ cd  $ git clone https://github.com/sstephenson/rbenv.git .rbenv  $ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc  $ echo 'eval "$(rbenv init -)"' >> ~/.bashrc  $ exec $SHELL  $ git clone https://github.com/sstephenson/ruby-build.git ~/.rbenv/plugins/ruby-build  $ echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc  $ exec $SHELL  $ git clone https://github.com/sstephenson/rbenv-gem-rehash.git ~/.rbenv/plugins/rbenv-gem-rehash  $ rbenv install 2.1.6  $ rbenv global 2.1.6 |

상기의 절차로 설치 후 아래의 명령어로 ruby 설치 버전을 확인한다.

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

## ***BOSH INIT 설치***

Bosh-init은 MicroBOSH를 설치에 사용하는 도구이다. Bosh-init을 설치하는 절차는 다음과 같다.

1. 다음의 웹사이트에서 설치환경에 맞는 bosh-init 바이너리 파일을 다운로드한다.

|  |
| --- |
| **https://bosh.io/docs/install-bosh-init.html** |

1. 다운로드 받은 bosh-init에 실행 권한을 부여하고 설치한다.

|  |
| --- |
| # ~/Download 디렉토리에 파일을 다운로드 받았다고 가정  $ chmod +x ~/Downloads/bosh-init-\*  $ sudo mv ~/Downloads/bosh-init-\* /usr/local/bin/bosh-init |

1. Bosh-init 설치 확인

|  |
| --- |
| $ bosh-init -v |

## ***BOSH CLI 설치***

BOSH CLI를 설치하는 절차는 다음과 같다.

1. bosh\_cli를 설치한다.

|  |
| --- |
| $ gem install bosh\_cli --no-ri --no-rdoc |

1. bosh 또는 bosh help명령어를 실행하면 사용가능한 CLI목록이 출력된다.

|  |
| --- |
| $ bosh or bosh help |

# Install MicroBOSH

본 장에서는 단일 VM에서 동작하는 MicroBOSH를 클라우드 오픈소스 플랫폼인 OpenStack에서 설치하는 절차를 기술한다.

## ***설치 절차***

MicroBOSH를 배포하기 위해서는 아래와 같이 몇가지 준비 사항들이 사전에 준비되어 되어있어야 한다.

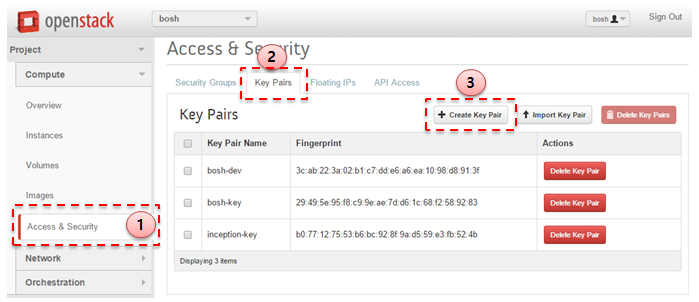
* OpenStack 환경설정
* 배포 Manifest 파일 작성
* Stemcell 다운로드
* MicroBOSH 배포

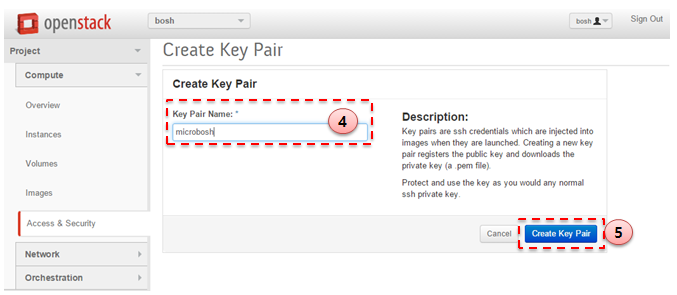
## ***OpenStack 환경설정***

Bosh-init을 통해 MicroBOSH를 배포하기 위한 OpenStack 사전 준비사항 및 필요한 사항을 아래의 가이드를 참조하여 설정한다.

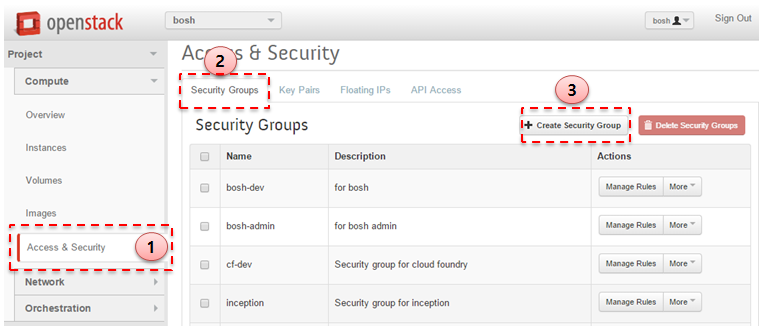
* 사전 준비 및 확인 사항
  + 지원 가능한 OpenStack Release 버전: Havana, IceHouse(권장), Juno
  + OpenStack 서비스 사용 가능 확인
    - Identity: 인증 및 OpenStack 서비스들의 Endpoint 조회
    - Compute: VM 생성과 IP 할당, 볼륨 생성 및 Attach
    - Image: 이미지 서비스를 이용한 Stemcell 저장
  + OpenStack 네트워크
    - 외부 또는 내부 네트워크의 Subnet
    - Floating IP
  + OpenStack 프로젝트 생성
* Key Pair 생성

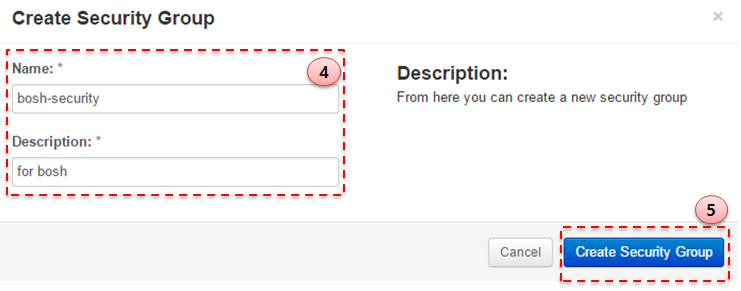
1. 오픈스택 horizon의 왼쪽 메뉴에서 Compute 🡪 Access & Security 선택

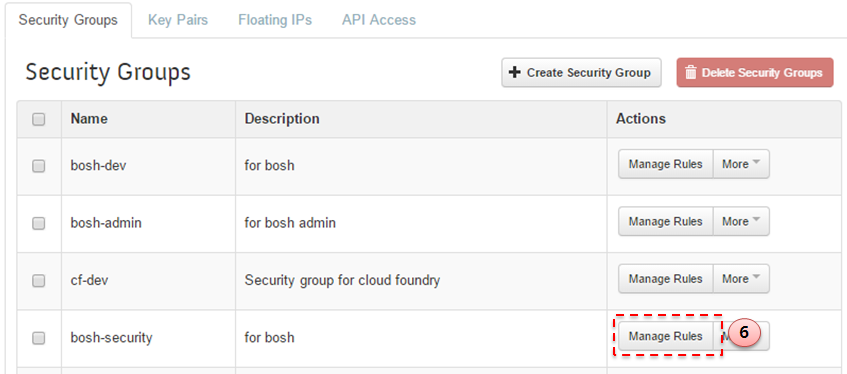
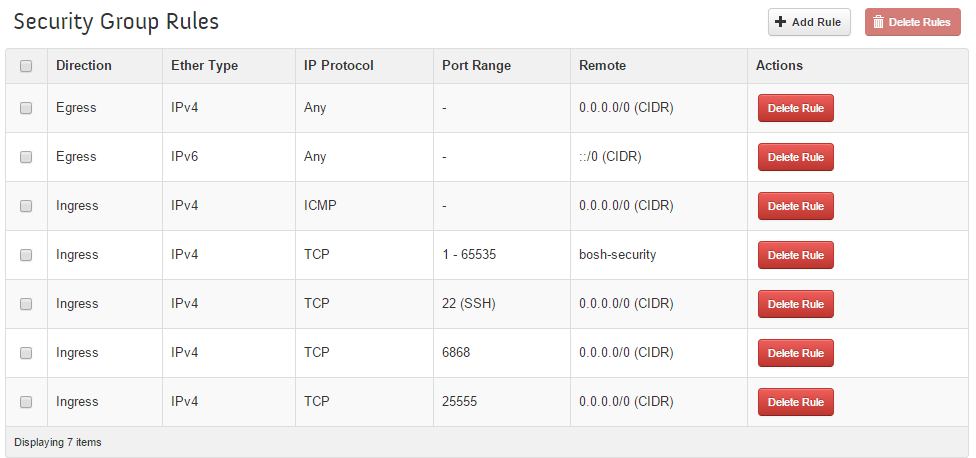


1. “Key Pairs” Tab 선택
2. 화면 상단의 “Create Key Pair” 선택
3. Key pair 이름 입력 후 “Create Key Pair” 선택 및 저장
4. Key를 생성하면 로컬에 다운로드 한다. (예제에서는 microbosh.pem 파일을 다운로드)
5. 다운로드한 Key를 로컬의 키 디렉토리에 이동시키고 권한을 400으로 변경한다.

|  |
| --- |
| $ mv ~/Download/microbosh.pem ~/.ssh/microbosh.pem  $ chmod 400 ~/.ssh/microbosh.pem |

* Security Group 생성 및 구성
  1. 오픈스택 horizon의 왼쪽 메뉴에서 Compute 🡪 Access & Security 선택
  2. Security Groups Tab 선택
  3. “Create Security Group” 선택
  4. 생성할 Security Group의 Name, Description 입력 후 “Create Security Group” 선택 (예제에서는 bosh-security 그룹을 생성)



* 1. Security Group 목록에서 생성된 Security Group의 “Manage Rules” 선택
  2. 다음 표와 같이 Bosh 관련Security Rule을 추가한다.

(예제에서는 모든 원격지에서 ssh접근을 허용하게 설정되어 있다. 설치 환경에 맞게 접근 할 수 있는 원격지를 제한 할 수 있다.)

## ***배포 Manifest 작성***

MicroBOSH 배포 Manifest는 배포에 필요한 컴포넌트 및 속성 정보를 YAML 파일 형식으로 정의한다.

1. Manifest파일을 저장할 Local 배포 디렉토리를 생성한다.
2. 생성한 디렉토리에서 배포 Manifest YAML 파일을 생성하고, 아래의 MicroBOSH 배포 Manifest Template을 복사하여 붙여넣고, 붉은 글씨 부분을 자신의 OpenStack 환경에 맞게 편집 후 저장한다.

|  |
| --- |
| ---  name: bosh # 배포명  releases:  - name: bosh # bosh 릴리즈  url: file:///***<로컬 경로>/<릴리즈 파일명>*** # bosh 릴리즈 파일이 있는 경로 및 파일명  - name: bosh-openstack-cpi # bosh-openstack-cpi 릴리즈  url: file:///***<로컬 경로>/<릴리즈 파일명>*** # bosh-openstack-cpi 릴리즈 파일이 있는 경로 및 파일명  resource\_pools:  - name: vms  network: private  stemcell:  url: file:///***<로컬 경로>/<stemcell 파일명>*** # stemcell 파일이 있는 경로 및 파일명  cloud\_properties:  instance\_type: m1.xlarge # VM 타입 (cpu, ram, disk)  disk\_pools:  - name: disks  disk\_size: 20\_000  networks:  - name: private  type: manual  subnets: # MicroBOSH를 설치할 오픈스택 내부망 정보  - range: ***PRIVATE-CIDR*** # 네트워크 대역  gateway: ***PRIVATE-GATEWAY-IP*** # 내부망 게이트웨이  dns: [***DNS-IP***] # DNS 목록  cloud\_properties: {net\_id: ***NETWORK-UUID***} # 오픈스택 내부망 아이디  - name: public  type: vip  jobs:  - name: bosh  instances: 1 # 설치 VM 수  templates: # 설치할 템플릿 목록 및 템플릿이 참조할 릴리즈명  - {name: nats, release: bosh}  - {name: redis, release: bosh}  - {name: postgres, release: bosh}  - {name: blobstore, release: bosh}  - {name: director, release: bosh}  - {name: health\_monitor, release: bosh}  - {name: registry, release: bosh}  - {name: openstack\_cpi, release: bosh-openstack-cpi}  resource\_pool: vms # 참조할 리소스풀 명칭  persistent\_disk\_pool: disks # 참조할 디스크풀 명칭  networks: # job에 설정할 네트워크 정보  - name: private  static\_ips: [***PRIVATE-IP***] # VM에 설정할 내부망 아이피  default: [dns, gateway]  - name: public  static\_ips: [***FLOATING-IP***] # VM에 설정할 외부망 아이피  properties: # job에 설정할 프로퍼티 정보  nats: # nats 프로퍼티 정보  address: 127.0.0.1  user: nats  password: nats-password  redis: # redis 프로퍼티 정보  listen\_address: 127.0.0.1  address: 127.0.0.1  password: redis-password  postgres: &db # postgres db 프로퍼티 정보  listen\_address: 127.0.0.1  host: 127.0.0.1  user: postgres  password: postgres-password  database: bosh  adapter: postgres  registry: # registry 프로퍼티 정보  address: ***PRIVATE-IP*** # VM에 설정할 내부망 아이피  host: ***PRIVATE-IP*** # VM에 설정할 내부망 아이피  db: \*db  http: {user: admin, password: admin, port: 25777}  username: admin  password: admin  port: 25777  blobstore: # blobstore 프로퍼티 정보  address: ***PRIVATE-IP*** # VM에 설정할 내부망 아이피  port: 25250  provider: dav  director: {user: director, password: director-password}  agent: {user: agent, password: agent-password}  director: # 디렉터 프로터피 정보  address: 127.0.0.1  name: my-bosh # 디렉터명  db: \*db  cpi\_job: openstack\_cpi # 플러그인 cpi명  max\_threads: 3 # 디렉터의 worker thread 개수  user\_management:  provider: local  local:  users:  - {name: admin, password: admin}  - {name: hm, password: hm-password}  hm: # hm 프로퍼티  director\_account: {user: hm, password: hm-password}  resurrector\_enabled: true # job이 중지된 경우, 해당 job에 대한 재시작 여부 설정  openstack: &openstack  auth\_url: ***IDENTITY-API-ENDPOINT*** # 오픈스택 API 엔드포인트  tenant: ***OPENSTACK-TENANT*** # 오픈스택 Tenant명  username: ***OPENSTACK-USERNAME*** # 오픈스택 사용자아이디  api\_key: ***OPENSTACK-PASSWORD*** # 오픈스택 사용자 비밀번호  default\_key\_name: ***microbosh*** # VM에 적용할 키명  default\_security\_groups: [***bosh***] # VM에 적용할 시큐리티 그룹명  agent: {mbus: "nats://nats:nats-password@***PRIVATE-IP***:4222"} # VM에 설정할 내부망 아이피  ntp: &ntp [***0.pool.ntp.org, 1.pool.ntp.org***] # ntp 서버 주소  cloud\_provider:  template: {name: openstack\_cpi, release: bosh-openstack-cpi}  ssh\_tunnel:  host: ***FLOATING-IP*** # VM에 설정할 외부망 아이피  port: 22 # VM에 설정할 ssh 포트  user: vcap # VM에 설정할 사용자 아이디  private\_key: ***~/.ssh/microbosh.pem*** # VM과 연결할 수 있는 개인키 정보  mbus: "https://mbus:mbus-password@***FLOATING-IP***:6868" # VM에 설정할 외부망 아이피  properties:  openstack: \*openstack  agent: {mbus: "https://mbus:mbus-password@0.0.0.0:6868"}  blobstore: {provider: local, path: /var/vcap/micro\_bosh/data/cache}  ntp: \*ntp |

## ***BOSH Stemcell 다운로드***

다음과 같이 BOSH CLI를 이용해서 BOSH Stemcell 이미지를 다운로드 받는다.

* BOSH CLI가 설치된 터미널 상에서 “bosh public stemcell” 명령어를 이용해서 BOSH Stemcell 목록을 확인한다.

|  |
| --- |
| **$ bosh public stemcells --all | grep openstack | grep ubuntu | sort -r** |

* BOSH CLI 명령어 “bosh download public stemcell [stemcell 파일명]”를 이용해서 다운로드 받는다.

|  |
| --- |
| **$ bosh download public stemcell bosh-stemcell-3147-openstack-kvm-ubuntu-trusty-go\_agent.tgz** |

* 본 가이드에서는 v3147을 다운로드 하였다.

## ***Release 다운로드***

다음의 웹사이트에서 설치할 릴리즈를 다운로드 한다.

* Bosh 릴리즈

|  |
| --- |
| <https://bosh.io/releases/github.com/cloudfoundry/bosh?all=1> |

* 본 가이드에서는 v233을 다운로드 하였다.
* Bosh-openstack-cpi 릴리즈

|  |
| --- |
| <https://bosh.io/releases/github.com/cloudfoundry-incubator/bosh-openstack-cpi-release?all=1> |

* 본 가이드에서는 v16을 다운로드 하였다.

## ***MicroBOSH 배포***

Bosh-init을 이용해 MicroBOSH를 배포한다.

1. 터미널 상에 bosh-init의 deploy를 실행한다.

|  |
| --- |
| $ bosh-init deploy ***<작성한 manifest>*** |

※ Deploy를 실행하면 manifest파일 있는 디렉토리에 <manifest명>-state.json 파일이 생긴다. MicroBOSH를 삭제할 경우 필요한 파일이므로 삭제하지 않는다.

1. 배포가 성공하면 다음과 같이 Target 설정하면서 기본 계정인 admin/admin으로 로그인한다.

|  |
| --- |
| $ bosh target ***<FLOATING-IP>***  **$ bosh target 172.16.xxx.xxx**  Target set to `my-bosh'  Your username: **admin**  Enter password:  Logged in as `admin' |

1. MicroBOSH의 상태정보를 확인한다.

|  |
| --- |
| **$ bosh status**  Config  /home/ubuntu/.bosh\_config **# BOSH CLI 설정파일 위치**  Director **# 현재 설정된 Director 정보**  Name my-bosh **# Director 이름**  URL **https://MicroBOSH-IP:25555** **# Director URL**  Version 1.3001.0 (00000000) **# Director 버전**  User admin **# Director 로그인 계정**  **UUID dfb961e2-2b0a-4fce-b201-dff4ea3137bc # Director UUID**  CPI openstack **# Director의 CPI**  dns enabled (domain\_name: microbosh) **# Director의 DNS**  compiled\_package\_cache disabled **# Package 캐싱 여부 설정**  snapshots disabled **# Snapshot 기능 설정 여부**  Deployment  not set |

1. deploy를 삭제할 경우

|  |
| --- |
| $ bosh-init delete ***<작성한 manifest>*** |

1. bosh-init 배포에 대한 상세한 로그를 출력하고 싶은 경우, 다음과 같이 설정하고 bosh-init을 실행한다.

|  |
| --- |
| # bosh-init 로그출력 레벨 설정  # 설정 가능한 로그 레벨은 debug, info, warn, error, none로 none이 기본 설정되어 있다.  $ export BOSH\_INIT\_LOG\_LEVEL=***<로그 레벨>***  # 로그를 파일로 출력할 경우  $ export BOSH\_INIT\_LOG\_PATH=***<로그를 출력할 경로>/<로그 파일명>*** |

# Install BOSH

본 장에서는 이전 장까지 설치한 MicroBOSH 이용하여 BOSH를 설치하는 절차를 기술한다.

## ***설치 절차***

BOSH 를 설치하는 절차는 다음과 같다.

* BOSH Stemcell 업로드
* BOSH Release 업로드
* 배포 Manifest 작성
* BOSH 배포

## ***BOSH Stemell 업로드***

Stemcell 업로드는 MicroBOSH가 BOSH VM인스턴스를 생성하기 위한 기반 이미지를 등록하는 절차로 4.4 절에서 다운로드 받은 Stemcell을 이용하여 아래와 같이 MicroBOSH Director에 업로드한다.

|  |
| --- |
| $ **bosh upload stemcell *<stemcell 파일 경로>/<stemcell 파일명>*** |

Stemcell 업로드가 성공하면 나면 bosh stemcells 명령어을 실행하여 stemcell을 확인한다.

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

## ***BOSH Release 업로드***

BOSH Release는 BOSH가 필요로 하는 소프트웨어 패키지들이 정의되어 있는 Release로 아래와 같이 CLI를 이용하여 MicroBOSH에 업로드한다.

|  |
| --- |
| $ **bosh upload release *<release 파일 경로>/<release 파일명>*** |

* 4.5 절에서 다운받은 ‘bosh’ 릴리즈를 업로드 한다.

release 업로드가 성공하면 나면 bosh releases 명령어을 실행하여 release를 확인한다.

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

## ***배포 Manifest 작성***

MicroBOSH 배포 Manifest파일 작성과 동일하게 로컬에 YAML파일을 생성하고, 아래의 BOSH 배포 Manifest을 복사하여 붙여넣고, 붉은 글씨 부분을 자신의 OpenStack 환경에 맞게 편집 후 저장한다.

|  |
| --- |
| ---  name: bosh # bosh 배포명  director\_uuid: ***6f7e18b2-52b3-429f-ba5a-7a7d2397a893*** # MicroBOSH uuid  release: # bosh 릴리즈 정보  name: bosh # 릴리즈명  version: ***233*** # 릴리즈 버전  compilation:  cloud\_properties:  instance\_type: m1.medium # 설치 패키지를 컴파일 할 VM의 타입 정보  network: default # 설치 패키지를 컴파일 할 VM에 할당할 네트워크 정보  reuse\_compilation\_vms: true  workers: 3 # 설치 패키지를 컴파일 할 VM 개수  update:  canaries: 1 # job설치를 시험하기 위한 생성 VM 수  canary\_watch\_time: 3000-120000 # 시험용 VM의 생성 응답대기 시간 (밀리초)  max\_in\_flight: 4 # 동일 Job에 대한 VM을 동시에 생성할 최대 수  update\_watch\_time: 3000-120000 # 시험용 VM의 업데이트 응답대기 시간 (밀리초)  networks:  - name: elastic # 네트워크 이름 설정  type: vip # 네트워크 타입 설정 (타입: manual, dynamic, vip)  cloud\_properties: {}  - name: default # 네트워크 이름 설정  type: manual # 네트워크 타입 설정 (타입: manual, dynamic, vip)  subnets: # 내부망 네트워크 서브넷 정보 설정  - range: ***192.168.20.0/24*** # 내부망 네트워크 대역 설정  dns: [***8.8.8.8***] # 내부망 네트워크 DNS 설정  gateway: ***192.168.20.1*** # 내부망 네트워크 게이트웨이 설정  static: # 내부망에 생성할 VM에 할당할 아이피 대역 설정  - ***192.168.20.6 - 192.168.20.12***  cloud\_properties:  net\_id: ***d8032a1c-xxxx-xxxx-xxxx-6d7282352381*** # 내부망에 대한 오픈스택 네트워크 아이디  resource\_pools: # VM에 할당할 리소스풀  - name: small # small 타입 리소스  network: default # small 타입 리소스에 할당할 네트워크 이름  size: 6 # small 타입의 job(=VM) 개수  stemcell: # small 타입의 job에 적용할 stemcell 정보  name: ***bosh-openstack-kvm-ubuntu-trusty-go\_agent*** # stemcell 이름  version: ***3147*** # stemcell 버전  cloud\_properties:  instance\_type: m1.small # 리소스에 적용할 오픈스택 flavor 타입  - name: medium # medium 타입 리소스  network: default  size: 1  stemcell:  name: ***bosh-openstack-kvm-ubuntu-trusty-go\_agent***  version: ***3140***  env:  bosh:  password: ***$6$JA/VRhS7guR2t$kruB3wpqcgyi7Ql2IZIfVC7WP.ZTxjRJ9hbyJV4Xn/wr.mcaOR0MwqAulcKY5gxkFQ0/DJul7UELCKsQ3LrZ71*** # medium 타입의 VM에 적용할 sha-512로 암호화된 비밀번호  cloud\_properties:  instance\_type: m1.medium  jobs:  - instances: 1 # 생성할 VM 수  name: nats # job(=VM) 이름  networks: # job에 할당할 네트워크 설정 정보  - default: # 기본 설정 네트워크 정보  - dns # 참조 대상의 DNS (네트워크 이름이 default의 DNS를 참조)  - gateway # 참조 대상의 게이트웨이 (네트워크 이름이 default의 게이트웨이를 참조)  name: default # 참조 대상의 네트워크 이름  static\_ips:  - ***192.168.20.6*** # job에 할당할 아이피  resource\_pool: small # job에 할당할 리소스 타입의 이름  template: nats # job 생성시에 참조할 템플릿 이름  - instances: 1  name: redis  networks:  - default:  - dns  - gateway  name: default  static\_ips:  - ***192.168.20.7***  resource\_pool: small  template: redis  - instances: 1  name: postgres  networks:  - default:  - dns  - gateway  name: default  static\_ips:  - ***192.168.20.8***  persistent\_disk: 16384 # VM에 persistent 디스크를 할당 할 경우, 디스크 사이즈 설정  resource\_pool: small  template: postgres  - instances: 1  name: blobstore  networks:  - default:  - dns  - gateway  name: default  static\_ips:  - ***192.168.20.9***  persistent\_disk: 51200  resource\_pool: small  template: blobstore  - instances: 1  name: director  networks:  - default:  - dns  - gateway  name: default  static\_ips:  - ***192.168.20.10***  - name: elastic  static\_ips:  - ***172.16.100.11*** # 디렉터의 경우 관리를 위해 외부에서 접속할 수 있는 아이피를 할당  persistent\_disk: 16384  resource\_pool: medium  templates:  - name: powerdns  release: bosh  - name: director  release: bosh  - instances: 1  name: registry  networks:  - default:  - dns  - gateway  name: default  static\_ips:  - ***192.168.20.11***  resource\_pool: small  template: registry  - instances: 1  name: health\_monitor  networks:  - default:  - dns  - gateway  name: default  static\_ips:  - ***192.168.20.12***  resource\_pool: small  template: health\_monitor  properties:  blobstore:  address: ***192.168.20.9*** # blobstore에 할당한 아이피(job에서 설정한 아이피와 동일)  agent: # agent 설정 정보  password: agent # agent 비밀번호  user: agent # agent 사용자 아이디  director: # 디렉터 설정 정보  password: director # 디렉터 비밀번호  user: director # 디렉터 사용자 아이디  director:  address: ***192.168.20.10*** # 디렉터에 할당한 아이피(job에서 설정한 아이피와 동일)  db:  database: bosh # 데이터베이스 이름  host: ***192.168.20.8*** # postgres에 할당한 아이피(job에서 설정한 아이피와 동일)  password: postgres # 데이터베이스 접속 비밀번호  user: postgres # 데이터베이스 사용자 아이디  name: mDirector # 디렉터 이름  dns:  address: ***192.168.20.10*** # DNS에 할당한 아이피(job에서 설정한 아이피와 동일)  db:  database: bosh  host: ***192.168.20.8*** # postgres에 할당한 아이피(job에서 설정한 아이피와 동일)  password: postgres  user: postgres  recursor: ***172.16.100.10*** # MicroBOSH에 할당한 아이피  hm:  director\_account:  password: admin  user: admin  http:  password: hm  user: hm  resurrector\_enabled: true # job이 중지된 경우, 해당 job에 대한 재시작 여부 설정  nats:  address: ***192.168.20.6*** # nats에 할당한 아이피(job에서 설정한 아이피와 동일)  password: nats  port: 4222  user: nats  postgres:  database: bosh  host: ***192.168.20.8*** # postgres에 할당한 아이피(job에서 설정한 아이피와 동일)  password: postgres  user: postgres  redis:  address: ***192.168.20.7*** # redis 에 할당한 아이피(job에서 설정한 아이피와 동일)  password: redis  registry:  address: ***192.168.20.11*** # registry에 할당한 아이피(job에서 설정한 아이피와 동일)  db:  database: bosh  host: ***192.168.20.8*** # postgres에 할당한 아이피(job에서 설정한 아이피와 동일)  password: postgres  user: postgres  http:  password: registry  user: registry  ntp: [***1.kr.pool.ntp.org, 0.asia.pool.ntp.org, 1.asia.pool.ntp.org***] # NTP 서버 설정  openstack:  auth\_url: ***IDENTITY-API-ENDPOINT*** # OpenStack Identity API End Point 설정  tenant: ***TENANT*** # OpenStack Tenant 이름 설정  username: ***USERNAME*** # OpenStack 계정 이름 설정  api\_key: ***PASSWORD*** # OpenStack 계정 비밀번호 설정  default\_key\_name: ***KEYPAIR-NAME*** # Key-Pair 이름 설정  private\_key: ***KEYPAIR-PATH*** # Key-Pair 경로 설정  default\_security\_groups: [***SECURITY\_GROUP\_NAME***] # Security Group 설정 |

※ VM의 vcap 아이디의 비밀번호는 다음의 절차를 통해 변경할 수 있다.

|  |
| --- |
| 1. 암호화된 비밀번호 생성도구 설치  **$ sudo apt-get install whois**  2. vcap 비밀번호 생성  **$ mkpasswd -m sha-512**  Password: ***vcap@password***  $6$4A2jmWvsaYv2O37d$WtT5ul2L4IeRcYcGljeYpd9NHuA3dbdUWcHuDInLG9542g0sU.j4/fj.0uVfEQuutTxw3FW4DANgIqn1OGVTq1  3. 생성한 암호화된 비밀번호를 manifest의 env.bosh.password에 적용 |

## ***BOSH 배포***

다음의 절차를 이용해서 BOSH를 배포한다.

1. 터미널에서 작성한 BOSH 배포 Manifest파일을 CLI에 설정한다.

|  |
| --- |
| $ bosh deployment ***<작성한 manifest>*** |

1. 아래와 같이 bosh deploy명령어를 이용하여 배포를 실행한다.

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

1. Target 설정하면서 기본 계정인 admin/admin으로 로그인한다.

|  |
| --- |
| $ bosh target ***<BOSH-FLOATING-IP>*** |

1. 배포한 BOSH의 상태정보를 확인한다.

|  |
| --- |
| $ bosh status |

1. BOSH를 삭제할 경우

|  |
| --- |
| $ bosh target *<****MicroBOSH-FLOATING-IP>***  $ bosh delete deployment ***<BOSH-DEPLOYMENT-NAME>*** |

1. BOSH 배포에서 오류가 발생한 경우, 아래의 명령어를 이용하여 로그를 확인한다.

|  |
| --- |
| $ bosh task ***<BOSH 배포 task 번호>*** --debug > ***<로그 출력 경로>/<로그 파일명>*** |

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