

전자정부표준프레임워크 기반의 Open PaaS 개발

MySQL 서비스팩 설치 가이드

(vSphere)

|  |  |  |
| --- | --- | --- |
| 작성자 | 송창학 (인) | 2015.7.31 |
| 검토자 | 안찬영 (인) | 2015.8. |
| 승인자 | 현재욱 (인) | 2015.8. |

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

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-1) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 1.0 | 2015.7.31 | 최초작성 | 송창학 | 현재욱 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 문서 개요 4](#_Toc429407313)

[*1.1.* *목적* 4](#_Toc429407314)

[*1.2.* *범위* 4](#_Toc429407315)

[*1.3.* *시스템 구성도* 4](#_Toc429407316)

[*1.4.* *참고자료* 5](#_Toc429407317)

[2. MySQL 서비스팩 설치 6](#_Toc429407318)

[*2.1.* *설치전 준비사항* 6](#_Toc429407319)

[*2.2.* *MySQL 서비스 릴리즈 업로드* 6](#_Toc429407320)

[*2.3.* *MySQL 서비스 Deployment 파일 수정 및 배포* 13](#_Toc429407321)

[*2.4.* *MySQL 서비스 브로커 등록* 29](#_Toc429407322)

[3. MySQL 연동 Sample Web App 설명 31](#_Toc429407323)

[*3.1.* *Sample Web App 구조* 31](#_Toc429407324)

[*3.2.* *개방형 클라우드 플랫폼에서 서비스 신청* 31](#_Toc429407325)

[*3.3.* *Sample Web App에 서비스 바인드 신청 및 App 확인* 32](#_Toc429407326)

# 문서 개요

## ***목적***

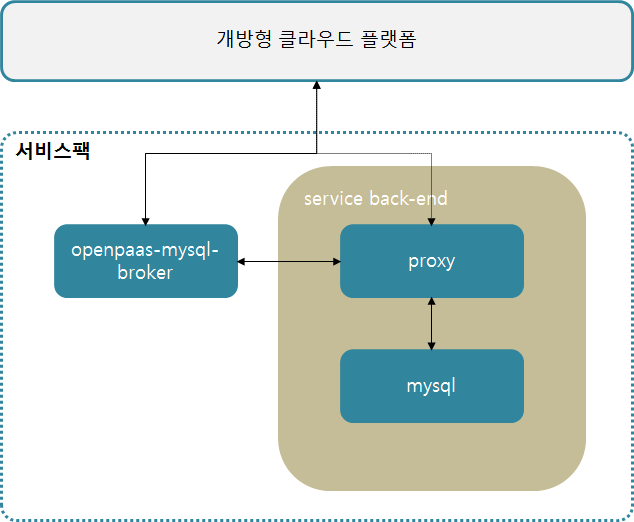
본 문서(MySQL 서비스팩 설치 가이드)는 전자정부표준프레임워크 기반의 Open PaaS에서 제공되는 서비스팩인 MySQL 서비스팩을 Bosh를 이용하여 설치 하는 방법과 Open PaaS의 SaaS 형태로 제공하는 Application 에서 MySQL 서비스를 사용하는 방법을 기술하였다.

## ***범위***

설치 범위는 MySQL 서비스팩을 검증하기 위한 기본 설치를 기준으로 작성하였다.

## ***시스템 구성도***

본 문서의 설치된 시스템 구성도입니다. MySQL Server, MySQL 서비스 브로커, Proxy로 최소사항을 구성하였다.



|  |  |
| --- | --- |
| 구분 | 스펙 |
| openpaas-mysql-broker | 2vCPU / 4GB RAM / 10GB Disk |
| proxy | 2vCPU / 4GB RAM / 10GB Disk |
| mysql | 2vCPU / 4GB RAM / 10GB Disk+100GB(영구적 Disk) |

## ***참고자료***

<http://bosh.io/docs>

<http://docs.cloudfoundry.org/>

# MySQL 서비스팩 설치

## ***설치전 준비사항***

본 설치 가이드는 Linux 환경에서 설치하는 것을 기준으로 하였다.

서비스팩 설치를 위해서는 먼저 BOSH CLI 가 설치 되어 있어야 하고 BOSH 에 로그인 및 타켓 설정이 되어 있어야 한다.

BOSH CLI 가 설치 되어 있지 않을 경우 먼저 BOSH 설치 가이드 문서를 참고 하여 BOSH CLI를 설치 해야 한다.

OpenPaaS 에서 제공하는 압축된 릴리즈 파일들을 다운받는다. (OpenPaaS-Deployment.zip, OpenPaaS-Sample-Apps.zip, OpenPaaS-Services.zip)

## ***MySQL 서비스 릴리즈 업로드***

* OpenPaaS-Services.zip 파일 압축을 풀고 폴더안에 있는 MySQL 서비스 릴리즈 openpaas-mysql-release-beta-1.0.tgz 파일을 복사한다.

업로드할 openpaas-mysql-release-beta-1.0.tgz 파일을 확인한다.

|  |
| --- |
| $ ls –all |

* 업로드 되어 있는 릴리즈 목록을 확인한다.

|  |
| --- |
| $ bosh releases    Mysql 서비스 릴리즈가 업로드 되어 있지 않은 것을 확인 |

* MySQL 서비스 릴리즈 파일을 업로드한다.

|  |
| --- |
| $ bosh upload release {서비스 릴리즈 파일 PATH}  $ bosh upload release openpaas-mysql-release-beta-1.0.tgz |

* 업로드 된 MySQL 릴리즈를 확인한다.

|  |
| --- |
| $ bosh releases    Mysql 서비스 릴리즈가 업로드 되어 있는 것을 확인 |

## ***MySQL 서비스 Deployment 파일 수정 및 배포***

BOSH Deployment manifest 는 components 요소 및 배포의 속성을 정의한 YAML[[2]](#footnote-2) 파일이다.

Deployment manifest 에는 sotfware를 설치 하기 위해서 어떤 Stemcell[[3]](#footnote-3)(OS, BOSH agent) 을 사용할것이며 Release[[4]](#footnote-4)(Software packages, Config templates, Scripts) 이름과 버전, VMs 용량, Jobs params 등을 정의가 되어 있다.

* OpenPaaS-Deployment.zip 파일 압축을 풀고 폴더안에 있는 vSphere용 MySQL Deployment 화일인 openpaas-mysql-vsphere.yml 를 복사한다.

다운로드 받은 Deployment Yml 파일을 확인한다. (openpaas-mysql-vsphere.yml)

|  |
| --- |
| $ ls –all |

* Director UUID를 확인한다.

BOSH CLI가 배포에 대한 모든 작업을 허용하기위한 현재 대상 BOSH Director의 UUID와 일치해야한다. ‘bosh status’ CLI 을 통해서 현재 BOSH Director 에 target 되어 있는 UUID를 확인 할 수 있다.

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

* Deploy시 사용할 Stemcell을 확인한다. (Stemcell 3016 버전 사용)

|  |
| --- |
| $ bosh stemcells    Stemcell 목록이 존재 하지 않을 경우 BOSH 설치 가이드 문서를 참고 하여 Stemcell 3016 버전을 업로드를 해야 한다. |

* openpaas-mysql-vsphere.yml Deployment 파일을 서버 환경에 맞게 수정한다. (빨간색으로 표시된 부분 특히 주의)

|  |
| --- |
| $ vi openpaas-mysql-vsphere.yml  # openpaas-mysql-vsphere 설정 파일 내용  name: openpaas-mysql-service # 서비스 배포이름(필수)  director\_uuid: xxxxx #bosh status 에서 확인한 Director UUID을 입력(필수)  releases:  - name: openpaas-mysql #서비스 릴리즈 이름(필수)  version: beta-1.0 #서비스 릴리즈 버전(필수): latest 시 업로드된 서비스 릴리즈 최신버전  update:  canaries: 1 # canary 인스턴스 수(필수)  canary\_watch\_time: 30000-600000 # canary 인스턴스가 수행하기 위한 대기 시간(필수)  max\_in\_flight: 1 # non-canary 인스턴스가 병렬로 update 하는 최대 개수(필수)  update\_watch\_time: 30000-600000 # non-canary 인스턴스가 수행하기 위한 대기 시간(필수)  compilation: # 컴파일시 필요한 가상머신의 속성(필수)  cloud\_properties: # 컴파일 VM을 만드는 데 필요한 IaaS의 특정 속성 (instance\_type, availability\_zone), 직접 cpu,disk,ram 사이즈를 넣어도 됨  cpu: 4  disk: 20480  ram: 4096  network: openpaas\_network # Networks block에서 선언한 network 이름(필수)  reuse\_compilation\_vms: true # 컴파일지 VM 재사용 여부(옵션)  workers: 6 # 컴파일 하는 가상머신의 최대수(필수)  jobs:  - instances: 1 # job 인스턴스 수(필수)  name: mysql # 작업 이름(필수): MySQL 서버  networks: # 네트워크 구성정보  - name: openpaas\_network # Networks block에서 선언한 network 이름(필수)  static\_ips: 10.30.40.162 # 사용할 IP addresses 정의(필수): MySQL 서버 IP  persistent\_disk: 100000 # 영구적 디스크 사이즈 정의(옵션): 100G  properties: # job에 대한 속성을 지정(필수)  admin\_password: admin # MySQL 어드민 패스워드  cluster\_ips: # 클러스터 구성시 IPs(필수)  - 10.30.40.162 # MySQL 서버 IP  network\_name: openpaas\_network # Networks block에서 선언한 network 이름  seeded\_databases: null  syslog\_aggregator: null  release: openpaas-mysql # 서비스 릴리즈 이름(필수)  resource\_pool: services-small # Resource Pools block에 정의한 resource pool 이름(필수)  template: mysql # job template 이름(필수)  - instances: 1  name: proxy # 작업 이름(필수): proxy  networks:  - name: openpaas\_network  static\_ips: 10.30.40.163 # 사용할 IP addresses 정의(필수): Proxy IP  properties:  cluster\_ips:  - 10.30.40.162 # MySQL 서버 IP  external\_host: controller.open-paas.com # CF 설치시 설정한 외부 호스트 정보(필수)  nats: # CF 설치시 설치한 nats 정보 (필수)  machines:  - 10.30.40.111 # nats 서버 IP  password: admin # nats 유저 비밀번호  port: 4222 # nats 서버 포트번호  user: nats # nats 서버 유저아이디  network\_name: openpaas\_network  proxy: # proxy 정보 (필수)  api\_password: admin # proxy api 유저 비밀번호(필수)  api\_username: api # proxy api 유저아이디  syslog\_aggregator: null  release: openpaas-mysql  resource\_pool: services-small  template: proxy # job template 이름(필수)  - instances: 1  name: openpaas-mysql-broker # 작업 이름(필수): 서비스 브로커  networks:  - name: openpaas\_network  static\_ips: 10.30.40.161 # 사용할 IP addresses 정의(필수): 서비스 브로커 IP  properties:  auth\_password: admin # 브로커 접근 아이디 비밀번호(필수)  auth\_username: admin # 브로커 접근 아이디(필수)  cc\_api\_uri: http://api.controller.open-paas.com # CF 설치시 설정한 api uri 정보(필수)  cookie\_secret: admin  # external\_host: 10.30.40.161 # 사용할 IP addresses 정의(필수): 서비스 브로커 IP  external\_host: p-mysql.controller.open-paas.com # 사용할 외부 호스트 정의(필수): 서비스 브로커  max\_user\_connections\_default: 40  mysql\_node:  admin\_password: admin  host: 10.30.40.163 # proxy IP(필수)  persistent\_disk: 100000 # 영구적 디스크 사이즈 정의(옵션): 100G  nats: # CF 설치시 설정하는 Nats 정보 (필수): 위와 동일  machines:  - 10.30.40.111  password: admin  port: 4222  user: nats  network\_name: openpaas\_network  networks:  broker\_network: openpaas\_network  services:  - dashboard\_client: # 서비스 dashboard client 정보  id: p-mysql # 서비스 dashboard 에 접속할 ID  secret: admin # 서비스 dashboard 에 접속시 사용할 secret  description: MySQL databases on demand # 서비스 설명  id: 44b26033-1f54-4087-b7bc-da9652c2a539 # 서비스 아이디(고유식별자)  metadata:  displayName: MySQL for Pivotal Cloud Foundry # 화면에서 보여줄 서비스 설명  documentationUrl: http://docs.controller.open-paas.com/p-mysql/ # 문서제공되는 Url  imageUrl: data:image/png;base64,  longDescription: Creating a service instance provisions a database. Binding  applications provisions unique credentials for each application to access  the database. # 서비스 상세 설명  providerDisplayName: OpenPaas Software # provider 이름  supportUrl: https://support.pivotal.io # 서포트 가능한 Url  name: p-mysql # 서비스 이름  plan\_updateable: true # plan 업데이트 가능여부  plans: # plan 정보  - description: Shared MySQL Server # plan 설명  id: ab08f1bc-e6fc-4b56-a767-ee0fea6e3f20 # plan 아이디  max\_storage\_mb: 100 # plan 정보: 100mb 저장공간  max\_user\_connections: 20 # plan 정보: 사용자 접속수  metadata:  bullets: # 화면에 제공할 plan 정보  - Shared MySQL server  - 100 MB storage  - 20 concurrent connections  costs: # 과금 정책 설정  - amount:  usd: 0 # US 달러: 0달러  unit: MONTH # 월별 징수  displayName: 100 MB  name: 100mb # plan 이름  - description: Shared MySQL Server  id: 11d0aa36-dcec-4021-85f5-ea4d9a5c8342  max\_storage\_mb: 1000  max\_user\_connections: 40  metadata:  bullets:  - Shared MySQL server  - 1000 MB storage  - 40 concurrent connections  costs:  - amount:  usd: 0  unit: MONTH  displayName: 1 GB  name: 1gb  tags:  - mysql # 테그 정보  syslog\_aggregator: null  release: openpaas-mysql  resource\_pool: services-small # Resource Pools block에 정의한 resource pool 이름(필수)  template: cf-mysql-broker # job template 이름(필수)  - instances: 1  lifecycle: errand # bosh deploy시 vm에 생성되어 설치 되지 않고 bosh errand 로 실행할때 설정, 주로 테스트 용도에 쓰임  name: broker-registrar # 작업 이름: 서비스 브로커 등록  networks:  - name: openpaas\_network  properties:  broker: # 서비스 브로커 설정 정보  host: 10.30.40.161 # 서비스 브로커 IP  name: p-mysql  password: admin  username: admin  protocol: http  port: 80  cf:  admin\_password: admin  admin\_username: admin  api\_url: http://api.controller.open-paas.com  release: openpaas-mysql  resource\_pool: services-small  template: broker-registrar  - instances: 1  lifecycle: errand  name: broker-deregistrar # 작업 이름: 서비스 브로커 삭제  networks:  - name: openpaas\_network  properties:  broker:  name: p-mysql  cf:  admin\_password: admin  admin\_username: admin  api\_url: http://api.controller.open-paas.com  release: openpaas-mysql  resource\_pool: services-small  template: broker-deregistrar  - instances: 1  lifecycle: errand  name: acceptance-tests # 작업이름: 서비스팩이 정상적으로 설치 되었는 테스트  networks:  - name: openpaas\_network  properties:  broker:  host: 10.30.40.161 # 서비스 브로커 IP  cf:  admin\_password: admin  admin\_username: admin  api\_url: http://api.controller.open-paas.com  apps\_domain: controller.open-paas.com  proxy: # proxy 정보  api\_password: admin  api\_username: api  external\_host: controller.open-paas.com  service:  max\_user\_connections\_default: 40  name: p-mysql  plans:  - max\_storage\_mb: 100  max\_user\_connections: 20  plan\_name: 100mb  - max\_storage\_mb: 1000  max\_user\_connections: 40  plan\_name: 1gb  release: openpaas-mysql  resource\_pool: services-small  template: acceptance-tests  meta:  apps\_domain: controller.open-paas.com # CF 설치시 설정한 apps 도메인 정보  environment: null  external\_domain: controller.open-paas.com # CF 설치시 설정한 외부 도메인 정보  nats: # CF 설치시 설정한 nats 정보  machines:  - 10.30.40.111  password: admin  port: 4222  user: nats  syslog\_aggregator: null  networks: # 네트워크 블록에 나열된 각 서브 블록이 참조 할 수있는 작업이 네트워크 구성을 지정, 네트워크 구성은 네트워크 담당자에게 문의 하여 작성 요망  - name: openpaas\_network  subnets:  - cloud\_properties:  name: Internal # vsphere 에서 사용하는 network 이름(필수)  dns: # DNS 정보  - 10.30.20.24  - 8.8.8.8  gateway: 10.30.20.23  name: default\_unused  range: 10.30.0.0/16  #reserved: # 설치시 제외할 IP 설정  #- 10.30.0.1 - 10.30.10.254  #- 10.30.40.1 - 10.30.40.100  #- 10.30.40.201 - 10.30.254.254  static:  - 10.30.40.110 - 10.30.40.200 # 사용 가능한 IP 설정  type: manual  properties: {}  resource\_pools: # 배포시 사용하는 resource pools를 명시하며 여러 개의 resource pools 을 사용할 경우 name 은 unique 해야함(필수)  - cloud\_properties: # 컴파일 VM을 만드는 데 필요한 IaaS의 특정 속성을 설명 (instance\_type, availability\_zone), 직접 cpu, disk, 메모리 설정가능  cpu: 2  disk: 10480  ram: 4096  name: services-small # 고유한 resource pool 이름  #size: 4 # resource pool 안의 가상머신 개수, 주의) jobs 인스턴스 보다 작으면 에러가 남, size 정의하지 않으면 자동으로 가상머신 크기 설정  network: openpaas\_network  stemcell:  name: bosh-vsphere-esxi-ubuntu-trusty-go\_agent # stemcell 이름(필수)  version: "3016" # stemcell 버전(필수) |

* Deploy 할 deployment manifest 파일을 BOSH 에 지정한다.

|  |
| --- |
| $ bosh deployment {Deployment manifest 파일 PATH}  $ bosh deployment openpaas-mysql-vsphere.yml |

* MySQL 서비스팩을 배포한다.

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

* 배포된 MySQL 서비스팩을 확인한다.

|  |
| --- |
| $ bosh vms |

## ***MySQL 서비스 브로커 등록***

Mysql 서비스팩 배포가 완료 되었으면 Application에서 서비스 팩을 사용하기 위해서 먼저 MySQL 서비스 브로커를 등록해 주어야 한다.

서비스 브로커 등록시 개방형 클라우드 플랫폼에서 서비스 브로커를 등록 할 수 있는 사용자로 로그인이 되어 있어야 한다.

* 서비스 브로커 목록을 확인한다.

|  |
| --- |
| $ cf service-brokers |

* MySQL 서비스 브로커를 등록한다.

|  |
| --- |
| $ cf create-service-broker {서비스팩 이름} {서비스팩 사용자ID} {서비스팩 사용자비밀번호} http://{서비스팩 URL}   * 서비스팩 이름 : 서비스 팩 관리를 위해 개방형 클라우드 플랫폼에서 보여지는 명칭이다. 서비스 Marketplace에서는 각각의 API 서비스 명이 보여지니 여기서 명칭은 서비스팩 리스트의 명칭이다. * 서비스팩 사용자ID / 비밀번호 : 서비스팩에 접근할 수 있는 사용자 ID입니다. 서비스팩도 하나의 API 서버이기 때문에 아무나 접근을 허용할 수 없어 접근이 가능한 ID/비밀번호를 입력한다. * 서비스팩 URL : 서비스팩이 제공하는 API를 사용할 수 있는 URL을 입력한다.   $ cf create-service-broker mysql-service-broker admin admin http://p-mysql.controller.open-paas.com |

* 등록된 MySQL 서비스 브로커를 확인한다.

|  |
| --- |
| $ cf service-brokers |

* 접근 가능한 서비스 목록을 확인한다.

|  |
| --- |
| $ cf service-access    서비스 브로커 생성시 디폴트로 접근을 허용하지 않는다. |

* 특정 조직에 해당 서비스 접근 허용을 할당하고 접근 서비스 목록을 다시 확인한다. (전체 조직)

|  |
| --- |
| $ cf enable-service-access p-mysql  $ cf service-access |

# MySQL 연동 Sample Web App 설명

본 Sample Web App은 개발형 클라우드 플랫폼에 배포되며 MySQL의 서비스를 Provision과 Bind를 한 상태에서 사용이 가능하다.

## ***Sample Web App 구조***

Sample Web App은 개방형 클라우드 플랫폼에 App으로 배포가 된다. App을 배포하여 구동시 Bind 된 MySQL 서비스 연결정보로 접속하여 초기 데이터를 생성하게 된다. 배포 완료 후 정상적으로 App 이 구동되면 브라우져나 curl로 해당 App에 접속 하여 MySQL 환경정보(서비스 연결 정보)와 초기 적재된 데이터를 보여준다.

Sample Web App 구조는 다음과 같다.

|  |  |
| --- | --- |
| **이름** | **설명** |
| src | Sample 소스 디렉토리 |
| manifest | 개방형 클라우드 플랫폼에 app 배포시 필요한 설정을 저장하는 파일 |
| pom.xml | 메이븐 project 설정 파일 |
| target | 메이블 빌드시 생성되는 디렉토리(war 파일, classes 폴더 등) |

* OpenPaaS-Sample-Apps.zip 파일 압축을 풀고 Service 폴더안에 있는 MySQL Sample Web App인 hello-spring-mysql를 복사한다.

|  |
| --- |
| $ ls -all |

## ***개방형 클라우드 플랫폼에서 서비스 신청***

Sample Web App에서 MySQL 서비스를 사용하기 위해서는 서비스 신청(Provision)을 해야 한다.

\*참고: 서비스 신청시 개방형 클라우드 플랫폼에서 서비스를 신청 할 수 있는 사용자로 로그인이 되어 있어야 한다.

* 먼저 개방형 클라우드 플랫폼 Marketplace에서 서비스가 있는지 확인을 한다.

|  |
| --- |
| $ cf marketplace |

* Marketplace에서 원하는 서비스가 있으면 서비스 신청(Provision)을 한다.

|  |
| --- |
| $ cf create-service {서비스명} {서비스플랜} {내서비스명}   * 서비스명 : p-mysql로 Marketplace에서 보여지는 서비스 명칭이다. * 서비스플랜 : 서비스에 대한 정책으로 plans에 있는 정보 중 하나를 선택한다. MySQL 서비스는 100mb, 1gb를 지원한다. * 내서비스명 : 내 서비스에서 보여지는 명칭이다. 이 명칭을 기준으로 환경설정정보를 가져온다.   $ cf create-service p-mysql 100mb mysql-service-instance |

* 생성된 MySQL 서비스 인스턴스를 확인한다.

|  |
| --- |
| $ cf services |

## ***Sample Web App에 서비스 바인드 신청 및 App 확인***

서비스 신청이 완료되었으면 Sample Web App 에서는 생성된 서비스 인스턴스를 Bind 하여 App에서 MySQL 서비스를 이용한다.

\*참고: 서비스 Bind 신청시 개방형 클라우드 플랫폼에서 서비스 Bind 신청 할 수 있는 사용자로 로그인이 되어 있어야 한다.

* Sample Web App 디렉토리로 이동하여 manifest 파일을 확인한다.

|  |
| --- |
| $ cd hello-spring-mysql  $ vi manifest.yml  ---  applications:  - name: hello-tomcat-mysql #배포할 App 이름  memory: 512M # 배포시 메모리 사이즈  instances: 1 # 배포 인스턴스 수  path: target/hello-spring-mysql-1.0.0-BUILD-SNAPSHOT.war #배포하는 App 파일 PATH  참고: target/hello-spring-mysql-1.0.0-BUILD-SNAPSHOT.war 파일이 존재 하지 않을 경우 mvn 빌드를 수행 하면 파일이 생성된다. |

* --no-start 옵션으로 App을 배포한다.

--no-start: App 배포시 구동은 하지 않는다.

|  |
| --- |
| $ cf push --no-start |

* 배포된 Sample App을 확인하고 로그를 수행한다.

|  |
| --- |
| $ cf apps    $ cf logs {배포된 App명}  $ cf logs hello-tomcat-mysql |

* Sample Web App에서 생성한 서비스 인스턴스 바인드 신청을 한다.

|  |
| --- |
| $ cf bind-service hello-tomcat-mysql mysql-service-instance |

* 바인드가 적용되기 위해서 App을 재기동한다.

|  |
| --- |
| $ cf restart hello-tomcat-mysql |

* (참고) 바인드 후 App구동시 Mysql 서비스 접속 에러로 App 구동이 안될 경우 보안 그룹을 추가한다.

|  |
| --- |
| * rule.json 화일을 만들고 아래와 같이 내용을 넣는다.   $ vi rule.json  [  {  "protocol": "tcp",  "destination": "10.30.40.163",  "ports": "3306"  }  ]   * 보안 그룹을 생성한다.   $ cf create-security-group p-mysql rule.json     * 모든 App에 Mysql 서비스를 사용할수 있도록 생성한 보안 그룹을 적용한다.   $ cf bind-running-security-group p-mysql     * App을 리부팅 한다.   $ cf restart hello-tomcat-mysql |

* App이 정상적으로 MySQL 서비스를 사용하는지 확인한다.

|  |
| --- |
| * curl 로 확인   $ curl hello-tomcat-mysql.controller.open-paas.com     * 브라우져에서 확인     초기 적재 데이터  MySQL 접속 환경정보 |

1. 변경 내용: 변경이 발생되는 위치와 변경 내용을 자세히 기록(장/절과 변경 내용을 기술한다.) [↑](#footnote-ref-1)
2. YAML Ain’t Markup Language, http://www.yaml.org, http://ko.wikipedia.org/wiki/YAML [↑](#footnote-ref-2)
3. BOSH가 Stemcell로부터 복사된 VM을 제어할 수 있도록 BOSH Agent가 내장되어 있는데 이를 “Stemcell”이라 부른다. [↑](#footnote-ref-3)
4. Release는 시스템에서 설치될 구성 및 소프트웨어들을 포함한다. [↑](#footnote-ref-4)