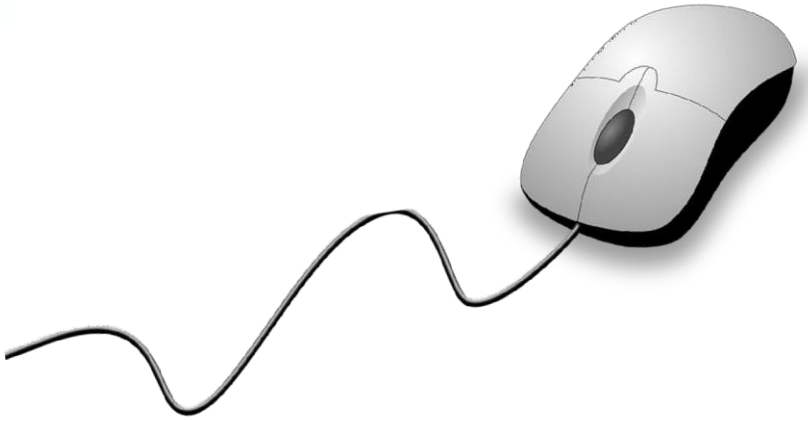
공개SW 솔루션 설치 & 활용 가이드

응용SW > 클라우드서비스



OPENSHIFT origin

제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

OPENUP

오픈소스 소프트웨어 통합지원센터

Open Source Software Support Center

CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

1. 개요

소개	• OCI 표준 컨테이너에 대한 관리 및 오케스트레이션을 위한, 엔터프라이즈급 Kubernetes 를 탑재한 컨테이너 플랫폼 혹은 PaaS(Platform as a Service) 제품 • 국내외 많은 레퍼런스를 확보하고 있으며, 표준 컨테이너 기술을 기반으로 되어 있기 때문에 높은 호환성을 제공		
주요기능	• 표준 컨테이너 관리 및 오케스트레이션 • 관리 및 서비스 노드의 HA(High Availability, 고가용성) 제공 • 장애시 자동으로 컨테이너가 재기동하는 Self-Healing 제공 • 급변하는 부하에 능동적으로 대처하기 위한 Autoscaling 제공		
대분류	• 응용SW	**소분류**	• 클라우드서비스
라이선스형태	• Apache v2	**사전설치 솔루션**	• N/A
실행 하드웨어	• Physical or virtual system, or an instance running on a public or private IaaS	**버전**	• 3.6 (2017년 9월 기준)
특징	• 개발 및 운영을 위한 다양한 툴 제공(IDE, UX 등) • 다양한 application runtimes & services 제공 • Kubernetes 기반의 컨테이너 오케스트레이션 및 관리 • 다양한 환경에 설치 가능 - Bare-Metal(물리서버), VM(Vmware, RHV, Hyper-V 등), Public/Private IaaS(AWS, Hostway, OpenStack 등)		
보안취약점	• 취약점 ID : CVE-2016-8651 • 심각도 : 3.5 LOW(V3) • 취약점 설명 : OpenShift가 이미지 요청을 처리하는 방식에 입력 인증 결함이 발견 • 대응방안 : 3.4 이상 업데이트 • 참고 경로 : https://access.redhat.com/errata/RHSA-2016:2915		
개발회사/커뮤니티	• Red Hat, Inc. / openshift.org		
공식 홈페이지	• https://www.openshift.org		

2. 기능요약

- OpenShift Origin 의 주요 기능

주요기능	지원여부
OCI 표준 컨테이너	지원
다양한 환경에 설치 - Bare-Metal(물리서버), VM(Vmware, RHV, Hyper-V 등), Public/Private IaaS(AWS, Hostway, OpenStack 등)	지원
SDN을 통한 컨테이너간 자동 네트워크 구성	지원
CI/CD 용 Git, Jenkins 컨테이너	지원
Router를 통한 자동 부하 분산	지원
급변하는 부하에 능동적으로 대처하기 위한 Autoscaling	지원
다양한 application runtimes & services 제공	지원

3. 실행환경

- OpenShift Origin v3.6 은 Fedora 21, CentOS 7.3, RHEL(Red Hat Enterprise Linux) 7.3, 혹은 RHEL(Red Hat Enterprise Linux) 7.4 에서 설치 가능합니다.
- 자세한 설치 및 실행환경에 대해서는 아래의 표를 참고하시기 바랍니다.

Masters	• Physical or virtual system, or an instance running on a public or private IaaS. • Base OS: Fedora 21, CentOS 7.3, RHEL 7.3, or RHEL 7.4 with the "Minimal" installation option and the latest packages from the Extras channel, or RHEL Atomic Host 7.3.6 or later. • 2 vCPU. • Minimum 16 GB RAM. • Minimum 40 GB hard disk space for the file system containing */var/*. (1) • Minimum 1 GB hard disk space for the file system containing */usr/local/bin/*. • Minimum 1 GB hard disk space for the file system containing the system's temporary directory. (2)
Nodes	• Physical or virtual system, or an instance running on a public or private IaaS. • Base OS: Fedora 21, CentOS 7.3, RHEL 7.3, or RHEL 7.4 with "Minimal" installation option, or RHEL Atomic Host 7.3.6 or later. • NetworkManager 1.0 or later. • 1 vCPU. • Minimum 8 GB RAM. • Minimum 15 GB hard disk space for the file system containing */var/*. (1) • Minimum 1 GB hard disk space for the file system containing */usr/local/bin/*. • Minimum 1 GB hard disk space for the file system containing the system's temporary directory. (2) • An additional minimum 15 GB unallocated space to be used for Docker's storage back end; see Configuring Docker Storage.

https://docs.openshift.org/3.6/install_config/install/prerequisites.html

4. 설치 및 실행

세부 목차

1. Prerequisites
2. 네트워크 및 Hostname 설정
3. Base Package 설치
4. Package Update
5. Hostname Update
6. RPM-based Installer 구성
7. NTP 적용
8. DNS 설치
9. Docker 설치
10. Docker Storage 구성
11. Host간 Password-less Access 구성
12. Advanced Installation 실행
13. 설치 검증

4. 설치 및 실행

4.1 Prerequisites(1/3)

- Red Hat Enterprise Linux v7.3 기반에 OpenShift Origin 3.6 을 설치 하는 방법에 대해서 설명한다.
- 3대의 Master Host(Host 라 함은 Bare-Metal 혹은 VM를 의미함)를 사용하여 3중화로 구성하여, 오케스트레이션 및 관리 영역의 가용성을 확보하며, Master S/W 설치시 etcd(Distributed Key-Value Database) 도 3중화로 동시에 구성한다.
- 단, DNS의 경우는 Master Host 내의 SkyDNS를 외부 서비스용으로 사용할 수 없기 때문에, 외부에 별도의 DNS를 새로 구성하여 사용하거나, 이미 존재하는 DNS 를 사용할 수도 있다. 본 문서에서는 Bind를 이중화하여 설치하고 OpenShift Origin 에서 이 DNS를 사용하는 방법에 대해서 설명한다.또한, 본 문서에서는 LoadBalancer를 통한 Multi Master 구성에 대한 내용 만을 설명하며, L4 를 통한 Multi Master 구성에 대한 내용은 다루지 않는다.

4. 설치 및 실행

4.1 Prerequisites(2/3)

[설치 작업 절차]

Red Hat OpenShift Enterprise v3.6 을 사용하여 PaaS 환경을 구축하는 작업 절차는 아래와 같다.

- 패키지 저장소(yum repository) 구성 : On-Line(인터넷 접속 가능) 환경시 해당
- DNS 구성(이중화 구성)
- OpenShift Origin v3.6 설치(Master Host 3중화, Node Host 2대 및 Load Balancer 구성)
- 전체 아키텍처 정상 작동 확인

상기 설치 작업을 위해서, 총 8 대의 Host 가 필요합니다. 각 Host 별 역할은 아래와 같다.

- DNS 2 EA
- S/W Load Balancer 1EA
- OpenShift Origin v3.6 Master 3EA
- OpenShift Origin v3.6 Node 2EA

4. 설치 및 실행

4.1 Prerequisites(3/3)

[설치에 필요한 OS 구성]

OpenShift Origin v3.6 은 Fedora 21, CentOS 7.3, RHEL(Red Hat Enterprise Linux) 7.3, 혹은 RHEL(Red Hat Enterprise Linux) 7.4 에서 설치 가능하다. Master 및 Node Host에 대한 자세한 System Requirements 는 아래의 URL 에서 참고하자. https://docs.openshift.org/3.6/install_config/install/prerequisites.html

주의) 상기 사양은 최소 사양일 뿐, 운영환경을 위한 사양이 아님을 주의하자.

Masters	• Physical or virtual system, or an instance running on a public or private IaaS. • Base OS: Fedora 21, CentOS 7.3, RHEL 7.3, or RHEL 7.4 with the "Minimal" installation option and the latest packages from the Extras channel, or RHEL Atomic Host 7.3.6 or later. • 2 vCPU. • Minimum 16 GB RAM. • Minimum 40 GB hard disk space for the file system containing */var/*. [1] • Minimum 1 GB hard disk space for the file system containing */usr/local/bin/*. • Minimum 1 GB hard disk space for the file system containing the system's temporary directory. [2]
Nodes	• Physical or virtual system, or an instance running on a public or private IaaS. • Base OS: Fedora 21, CentOS 7.3, RHEL 7.3, or RHEL 7.4 with "Minimal" installation option, or RHEL Atomic Host 7.3.6 or later. • NetworkManager 1.0 or later. • 1 vCPU. • Minimum 8 GB RAM. • Minimum 15 GB hard disk space for the file system containing */var/*. [1] • Minimum 1 GB hard disk space for the file system containing */usr/local/bin/*. • Minimum 1 GB hard disk space for the file system containing the system's temporary directory. [2] • An additional minimum 15 GB unallocated space to be used for Docker's storage back end; see Configuring Docker Storage.

4. 설치 및 실행

4.2 네트워크 및 Hostname 설정

- 여기서는, 아래와 같이 테스트 시스템을 구성한다.

대상장비	호스트네임	IP Address	Prefix	NIC 인터페이스
DNS #1	dns01.rhkoso36.com	192.168.122.159	24	eth0 (외부)
DNS #2	dns02.rhkoso36.com	192.168.122.160	24	eth0 (외부)
Master Host #1	master01.rhkoso36.com	192.168.122.181	24	eth0 (외부)
Master Host #2	master02.rhkoso36.com	192.168.122.182	24	eth0 (외부)
Master Host #3	master03.rhkoso36.com	192.168.122.183	24	eth0 (외부)
Node Host #1	node01.rhkoso36.com	192.168.122.184	24	eth0 (외부)
Node Host #2	node02.rhkoso36.com	192.168.122.185	24	eth0 (외부)
Node Host #3	node03.rhkoso36.com	192.168.122.186	24	eth0 (외부)
Load Balancer	lb.rhkoso36.com	192.168.122.180	24	eth0 (외부)

또한, 본 문서에서는 Master cluster의 URL은 cluster.rhkoso36.com을 사용하며, OpenShift Origin PaaS 에서 생성되는 Docker container 의 도메인 서비스를 위한 subdomain 은 *.paas.rhkoso36.com을 사용한다.

4. 설치 및 실행

4.3 Base Package 설치

- 설치전 아래 명령을 전체 Host에서 수행하여 전체 Host에 Base Package 를 설치합니다.

[모든 Host] Base Package 설치

```
# yum install wget git net-tools bind-utils iptables-services bridge-utils
bash-completion kexec-tools sos psacct
```

4. 설치 및 실행

4.4 Package Update

- 설치전 아래 명령을 전체 Host에서 수행하여 전체 Host를 update 한다.

[모든 Host] 전체 package update

```
# yum update -y
```

4. 설치 및 실행

4.5 Hostname Update

- 설치전 아래 명령을 이용하여, 전체 Host에서 수행하여 전체 Host의 hostname 를 update 한다. 여기서, <hostname>은 변경할 hostname 이다.

  ※ "4.2 네트워크 및 Hostname 설정" 의 내용을 참고

[모든 Host] hostname update

```
# hostnamectl set-hostname <hostname>
```

4. 설치 및 실행

4.6 RPM-based Installer 구성

- 설치전 아래 명령을 Master01 에서 수행하여 EPEL repository 를 설치한다.

[Master01] EPEL repository 설치

```
# yum -y install \
  https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- 아래 명령을 Master01 에서 수행하여 EPEL repository 를 disable 시킨다.

[Master01] EPEL repository disable 구성

```
# sed -i -e "s/^enabled=1/enabled=0/" /etc/yum.repos.d/epel.repo
```

- 아래 명령을 Master01 에서 수행하여 Ansible 을 설치한다.

[Master01] Ansible 설치

```
#  yum -y --enablerepo=epel install ansible pyOpenSSL
```

4. 설치 및 실행

4.7 NTP 적용

- 시간 설정 및 기본적인 튜닝을 통해 성능을 향상시킨다.(모든 서버에 동일한 시간 설정은 매우 중요합니다.)

[모든 Master Host 및 Node Host] NTP 및 Tuned 적용

```
# yum install -y ntp tuned
# ntpdate clock.redhat.com (internet 접속 가능시에만 해당)
# service ntpd start; chkconfig ntpd on
# tuned-adm profile throughput-performance (*각 시스템에 맞게 설정, VM 상에서 구성
할 경우에는 절대로 실행하면 안됨)
```

4. 설치 및 실행

4.8 DNS 설치(1/9)

- OpenShift가 설치되는 모든 Host에 대한 IP 정보 등록을 위한 DNS 서버를 설치한다.

[DNS Host 만 해당] Bind 패키지 설치

```
# yum install -y bind bind-utils
```

[DNS Host 만 해당] 설치에 필요한 변수(필요에 따라서 값을 수정하여 사용하면 됨) 등록 (DNS 설치 과정 중 재 부팅이 이루어지면 아래 변수 등록을 다시 해야 함)

```
# domain=rhkoso36.com
# keyfile=/var/named/${domain}.key
# echo ${domain}
# echo ${keyfile}
```

[DNS Host 만 해당] DNS 인증키 생성

```
# rm -vf /var/named/K${domain}*
# pushd /var/named
# dnssec-keygen -a HMAC-SHA256 -b 512 -n USER -r /dev/urandom ${domain}
# KEY="$(grep Key: K$domain*.private | cut -d ' ' -f 2)"
# echo ${KEY}
# popd
```

4. 설치 및 실행

4.8 DNS 설치(2/9)

- SELinux 및 DNS forward 설정을 한다.

[DNS Host 만 해당] SELinux Context 변경

```
# rndc-confgen -a -r /dev/urandom
# restorecon -v /etc/rndc.* /etc/named.*
# chown -v root:named /etc/rndc.key
# chmod -v 640 /etc/rndc.key
```

[DNS Host 만 해당] DNS forwarder 설정 (master 및 node 외, 갖고 있지 않은 도메인에 대해 한 단계 더 질의하는 방식) forward 안 할거면 생략 가능.

```
# vi /var/named/forwarders.conf
forwarders { 168.126.63.1; 168.126.63.2; 8.8.8.8; };

# restorecon -v /var/named/forwarders.conf
# chmod -v 755 /var/named/forwarders.conf
# rm -rvf /var/named/dynamic
# mkdir -vp /var/named/dynamic
```

4. 설치 및 실행

4.8 DNS 설치(3/9)

- dns01.rhkoso36.com 서버의 zone 파일을 수정한다.

[dns01.rhkoso36.com Primary 서버에서 작업] rhkoso36.com Zone 파일정보 수정

```
# vi /var/named/dynamic/${domain}.db

$ORIGIN .
$TTL 1 ; 1 seconds (for testing only)
rhkoso36.com IN SOA dns01.rhkoso36.com. hostmaster.rhkoso36.com. (
                  2014051404 ; serial
                  60 ; refresh (1 minute)
                  15 ; retry (15 seconds)
                  1800 ; expire (30 minutes)
                  10 ; minimum (10 seconds)
                  )

                  NS    dns01.rhkoso36.com.
                  MX    10 mail.rhkoso36.com.
$ORIGIN     rhkoso36.com.
dns01 A     192.168.122.159
dns02 A     192.168.122.160
```

4.8 DNS 설치(4/9)

- dns02.rhkoso36.com 서버의 zone 파일을 수정한다.

[dns02.rhkoso36.com Secondary 서버에서 작업] rhkoso36.com Zone 파일정보 수정

```
# vi /var/named/dynamic/${domain}.db

$ORIGIN .
$TTL 1 ; 1 seconds (for testing only)
rhkoso36.com IN SOA dns02.rhkoso36.com. hostmaster.rhkoso36.com. (
                  2014051404 ; serial
                  60 ; refresh (1 minute)
                  15 ; retry (15 seconds)
                  1800 ; expire (30 minutes)
                  10 ; minimum (10 seconds)
                  )

                  NS    dns02.rhkoso36.com.
                  MX    10 mail.rhkoso36.com.
$ORIGIN     rhkoso36.com.
dns01 A     192.168.122.159
dns02 A     192.168.122.160
```

4. 설치 및 실행

4.8 DNS 설치(5/9)

- DNS 관리에 필요한 보안키를 설정한다.

[DNS Host 만 해당] DNS 관리에 필요한 보안키(앞서 생성했었던 인증키) 설정

```
# cat <<EOF > /var/named/${domain}.key
key ${domain} {
algorithm HMAC-SHA256;
secret "${KEY}";
};
EOF

# cat /var/named/${domain}.key
key rhkoso36.com {
algorithm HMAC-SHA256;
secret "LZ3IsRS5RlSOvDxDBnE5vO/
+GaQhSQvlRi6evvbFKOuukfl7tHt+wqqP/U8aLcCvmrqbV0C45C4eV9FU3uXQ2Q==";
};

# chgrp named -R /var/named
# chown named -R /var/named/dynamic
# restorecon -rv /var/named
```

4. 설치 및 실행

4.8 DNS 설치(6/9)

- /etc/named.conf 파일의 설정을 변경한다.

[dns01.rhkoso36.com Primary 서버에서 작업] BIND 설정 파일 /etc/named.conf 설정

```
# vi /etc/named.conf

// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
      listen-on port 53 { any; };
      directory "/var/named";
      dump-file "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
      memstatistics-file "/var/named/data/named_mem_stats.txt";
      allow-query { any; };
      allow-transfer { 127.0.0.1; 192.168.122.159; }; // Primary 설정
      recursion yes;

      // forward first;
      // forward only;
      // include "forwarders.conf";

      /* Path to ISC DLV key */
      bindkeys-file "/etc/named.iscdlv.key";
};

logging {
      channel default_debug {
            file "data/named.run";
            severity dynamic;
      };
};
```

```
include "/etc/rndc.key";

controls {
      inet 127.0.0.1 port 953
      allow { 127.0.0.1; } keys { "rndc-key"; };
};

include "/etc/named.rfc1912.zones";
include "rhkoso36.com.key";

zone "rhkoso36.com" IN {
      type master;
      file "dynamic/rhkoso36.com.db";
      allow-update { key rhkoso36.com ; } ;
};
```

4. 설치 및 실행

4.8 DNS 설치(7/9)

- /etc/named.conf 파일의 설정을 변경한다.

[dns02.rhkoso36.com Secondary 서버에서 작업] BIND 설정 파일 /etc/named.conf 설정

```
# vi /etc/named.conf

// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
      listen-on port 53 { any; };
      directory "/var/named";
      dump-file "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
      memstatistics-file "/var/named/data/named_mem_stats.txt";
      allow-query { any; };
      allow-transfer { none; }; // Secondary 설정
      recursion yes;

      // forward first;
      // forward only;
      // include "forwarders.conf";

      /* Path to ISC DLV key */
      bindkeys-file "/etc/named.iscdlv.key";
};

logging {
      channel default_debug {
            file "data/named.run";
            severity dynamic;
      };
};

include "/etc/rndc.key";

controls {
      inet 127.0.0.1 port 953
      allow { 127.0.0.1; } keys { "rndc-key"; };
};

include "/etc/named.rfc1912.zones";
include "rhkoso36.com.key";

zone "rhkoso36.com" IN {
      type slave;
      masters { 192.168.122.159; };
      file "dynamic/rhkoso36.com.db";
      allow-update { key rhkoso36.com ; } ;
};
```

4. 설치 및 실행

4.8 DNS 설치(8/9)

- Bind 설정파일의 권한 설정 및 DNS 서버를 시작한다.

[DNS Host 만 해당] Bind 설정 파일 권한 설정

```
# chown -v root:named /etc/named.conf
# restorecon /etc/named.conf
```

[모든 Host 에 공통] Host Name Resolution 설정(추가)

```
# vi /etc/resolv.conf
search rhkoso36.com
nameserver 192.168.122.159
nameserver 192.168.122.160
```

[DNS Host 만 해당] DNS 서버 시작

```
# lokkit --service=dns
# chkconfig named on
# service named start

# nslookup dns01.rhkoso36.com
Server:     192.168.122.159
Address:    192.168.122.159#53

Name: dns01.rhkoso36.com
Address: 192.168.122.159
```

4. 설치 및 실행

4.8 DNS 설치(9/9)

- DNS01 장비에서 모든 Host를 등록 및 추가한다.

```
# nsupdate -k /var/named/rhkoso36.com.key
> server 127.0.0.1
> update delete master01.rhkoso36.com A
> update delete master02.rhkoso36.com A
> update delete master03.rhkoso36.com A
> update delete node01.rhkoso36.com A
> update delete node02.rhkoso36.com A
> update delete node03.rhkoso36.com A
> update delete lb.rhkoso36.com A
> update delete cluster.rhkoso36.com A
> update delete *.paas.rhkoso36.com A
> update add master01.rhkoso36.com 180 A 192.168.122.181
> update add master02.rhkoso36.com 180 A 192.168.122.182
> update add master03.rhkoso36.com 180 A 192.168.122.183
> update add node01.rhkoso36.com 180 A 192.168.122.184
> update add node02.rhkoso36.com 180 A 192.168.122.185
> update add lb.rhkoso36.com 180 A 192.168.122.180
> update add cluster.rhkoso36.com 180 A 192.168.122.180
> update add *.paas.rhkoso36.com 300 A 192.168.122.184
> update add *.paas.rhkoso36.com 300 A 192.168.122.185
> send
Ctrl-D 키를 이용하여 exit 한 후 아래의 명령으로 등록 확인

# nslookup master01.rhkoso36.com
Server:         192.168.122.159
Address:        192.168.122.159#53

Name: master01.rhkoso36.com
Address: 192.168.122.181

# nslookup master02.rhkoso36.com
Server:         192.168.122.159
Address:        192.168.122.159#53

Name: master02.rhkoso36.com
Address: 192.168.122.182
```

4. 설치 및 실행

4.9 Docker 설치

- 모든 장비에 Docker 엔진을 설치한다.

[Master Host #1 ~ #3, Node Host #1~#2] Docker 설치

Docker v1.12.6 를 설치합니다. Master 와 Node Host 에 모두 설치해야 합니다.

```
# yum install docker-1.12.6
```

[Master Host #1 ~ #3, Node Host #1~#2] /etc/sysconfig/docker 파일 수정

Docker Daemon 이 해당 subnet 상의 모든 Docker Registry 를 신뢰할 수 있도록 설정합니다. 172.30.0.0/16 은 master-config.yaml 파일상의 serviceSubnet 변수의 디폴트 값입니다. 이 값을 수정하거나, 혹은 secure 옵션을 적용할 경우에는 아래의 URL 을 참고하시기 바랍니다.

https://docs.openshift.org/3.6/install_config/install/host_preparation.html#installing-docker

```
# vi /etc/sysconfig/docker (OPTIONS 항목을 아래와 같이 수정)
OPTIONS='--selinux-enabled --insecure-registry 172.30.0.0/16'
```

4. 설치 및 실행

4.10 Docker Storage 구성

- Docker container 와 image는 모두 Docker Storage에 저장된다. 각 Host 에 Docker Storage를 위한 저장소를 구성하는 데는 세가지 방식 중 하나를 선택할 수 있다.

  Option A) Block device 를 추가하여 사용

  Option B) 이미 존재하는, 사용하지 않는 Volume Group

  Option C) root file system 이 위치하고 있는, Volume Group에 남아 있는 공간

여기서는 Option A의 방식을 예로 설명합니다. 다른 방법을 사용하길 원한다면, 아래의 URL을 참고하자.

https://docs.openshift.org/3.6/install_config/install/host_preparation.html#configuring-dockerstorage

[Master Host #1 ~ #3, Node Host #1~#2] Docker Storage Setup 파일 수정

/etc/sysconfig/docker-storage-setup 파일을 아래와 같이 수정합니다. 아래에서 DEVS 변수의 값은 block device 의 이름이며, 설치하는 환경에 맞게끔 수정하면 됩니다.

```
# cat <<EOF > /etc/sysconfig/docker-storage-setup
DEVS=/dev/vdc
VG=docker-vg
EOF
```

[Master Host #1 ~ #3, Node Host #1~#2] docker-storage-setup 실행

```
# docker-storage-setup

# lvs (* docker-pool LV 생성 여부 확인)
```

[Master Host #1 ~ #3, Node Host #1~#2] Re-initiaize Docker

```
# systemctl stop docker
# rm -rf /var/lib/docker/*
# systemctl restart docker
```

4. 설치 및 실행

4.11 Host간 Password-less Access 구성(1/2)

- Quick 혹은 advanced installation 을 사용하면, 하나의 user로 모든 host에 password 입력없이 접속이 가능해야한다. 주로, root 유저를 사용하며, non-root 유저도 사용가능하다. 단, non-root 유저의 경우에는 password 없이 sudo를 사용할 수 있는 권한이 필요하다. 본 문서는 root 유저를 기준으로 설명한다.

```
# ssh-keygen

Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase): [Enter](단순히 enter 만 입력)
Enter same passphrase again: [Enter](단순히 enter 만 입력)
Your identification has been saved in /root/.ssh/rsync_id_rsa.
Your public key has been saved in /root/.ssh/rsync_id_rsa.pub.
The key fingerprint is:
a5:14:ad:89:f7:6d:27:b2:ec:78:db:73:5b:71:2b:43 root@master01.rhkoso36.com
The key's randomart image is:
+--[ RSA 2048]----+
|         ..      |
|          ..     |
|       ..o.      |
|       ..+o      |
|        .S. . E ..|
|           o = . +|
|          . + = ..|
|          .+.. +. |
|         .oo..o.. |
+-----------------+
```

4. 설치 및 실행

4.11 Host간 Password-less Access 구성(2/2)

- 아래의 명령으로 모든 Host에 SSH key를 복사합니다.

[Master Host #1] 나머지 모든 Host 에 SSH key 복사

```
# for host in master01.rhkoso36.com \ (주의 : master01 에도 복사해야 함)
  master02.rhkoso36.com \
  master03.rhkoso36.com \
  node01.rhkoso36.com \
  node02.rhkoso36.com \
  lb.rhkoso36.com; \
  do ssh-copy-id -i ~/.ssh/id_rsa.pub $host; \
  done
```

4. 설치 및 실행

4.12 Advanced Installation 실행(1/5)

- 본격적으로 OpenShift Origin v3.6 을 설치하는 단계입니다. Quick Installation 혹은 Advanced installation 을 사용할 수 있지만, Quick Installation은 보통 Test용으로 설치할 경우에만 사용하며 운영환 경을 위한 구성시에는 Advanced Installation을 사용하는 것을 권장한다. 두 가지 방법 모두 Ansible 을 사용한다.

  본 문서에는 Advanced Installation 방법에 대해서만 설명한다. Quick Installation에 대해서 궁금하다면 아래의 URL 에서 확인할 수 있다. https://docs.openshift.org/3.6/install_config/install/quick_install.html

  또한, Advanced Installation에 대한 자세한 내용도 아래의 URL 에서 확인할 수 있다.
  https://docs.openshift.org/3.6/install_config/install/advanced_install.html

  Advanced Installation 은 /etc/ansible/hosts 파일에 구성하고자 하는 내용을 입력하여 설치를 수행한다. 상기 URL에서 다양한 방법으로 Master 및 Node 를 구성하는 여러 가지 예를 확인할 수 있다.

본 문서에서는 Master 3EA, Node 2EA 및 LoadBalancer 1EA 를 사용하여 구성하는 방법을 설명한다.

4. 설치 및 실행

4.12 Advanced Installation 실행(2/5)

- /etc/ansible/hosts 파일을 수정한다.

[Master Host #1] /etc/ansible/hosts 파일 수정

```
# vi /etc/ansible/hosts (아래는 해당 file의 내용)

# Create an OSEv3 group that contains the master, nodes, etcd, and lb
groups.
# The lb group lets Ansible configure HAProxy as the load balancing
solution.
# Comment lb out if your load balancer is pre-configured.
[OSEv3:children]
masters
nodes
etcd
lb

# Set variables common for all OSEv3 hosts
[OSEv3:vars]
ansible_ssh_user=root
openshift_deployment_type=origin

# Uncomment the following to enable htpasswd authentication; defaults to
# DenyAllPasswordIdentityProvider.
openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login':
'true', 'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider',
'filename': '/etc/origin/htpasswd'}]
#LDAP auth
#openshift_master_identity_providers=[{'name': 'my_ldap_provider',
'challenge': 'true', 'login': 'true', 'kind':
'LDAPPasswordIdentityProvider', 'attributes': {'id': ['dn'], 'email':
['mail'], 'name': ['cn'], 'preferredUsername': ['uid']}, 'bindDN': '',
'bindPassword': '', 'ca': '', 'insecure': 'false', 'url':
'ldap://ldap.example.com:389/ou=users,dc=example,dc=com?uid'}]
```

4. 설치 및 실행

4.12 Advanced Installation 실행(3/5)

- /etc/ansible/hosts 파일을 수정한다.(계속)

```
# Native high availbility cluster method with optional load balancer.
# If no lb group is defined installer assumes that a load balancer has
# been preconfigured. For installation the value of
# openshift_master_cluster_hostname must resolve to the load balancer
# or to one or all of the masters defined in the inventory if no load
# balancer is present.
openshift_master_cluster_method=native
openshift_master_cluster_hostname=cluster.rhkoso36.com
openshift_master_cluster_public_hostname=cluster.rhkoso36.com
openshift_master_default_subdomain=paas.rhkoso36.com

# override the default controller lease ttl
osm_controller_lease_ttl=30

# Configure custom named certificates
# NOTE: openshift_master_named_certificates is cached on masters and is an
# additive fact, meaning that each run with a different set of certificates
# will add the newly provided certificates to the cached
# set of certificates.
# If you would like openshift_master_named_certificates
# to be overwritten with
# the provided value,
# specify openshift_master_overwrite_named_certificates.
#openshift_master_overwrite_named_certificates: true
#
# Provide local certificate paths which will be deployed to masters
#openshift_master_named_certificates=[{"certfile": "/path/to/custom1.crt",
"keyfile": "/path/to/custom1.key"}]
```

4. 설치 및 실행

4.12 Advanced Installation 실행(4/5)

- /etc/ansible/hosts 파일을 수정한다.(계속)

```
#
# Detected names may be overridden by specifying the "names" key
#openshift_master_named_certificates=[{"certfile": "/path/to/custom1.crt",
"keyfile": "/path/to/custom1.key", "names": ["public-master-host.com"]}]

# host group for masters
[masters]
master01.rhkoso36.com
master02.rhkoso36.com
master03.rhkoso36.com

# host group for etcd
[etcd]
master01.rhkoso36.com
master02.rhkoso36.com
master03.rhkoso36.com

# Specify load balancer host
[lb]
lb.rhkoso36.com

# host group for nodes, includes region info
[nodes]
master01.rhkoso36.com openshift_node_labels="{'region': 'infra', 'zone':
'default'}"
master02.rhkoso36.com openshift_node_labels="{'region': 'infra', 'zone':
'default'}"
master03.rhkoso36.com openshift_node_labels="{'region': 'infra', 'zone':
'default'}"
node01.rhkoso36.com openshift_node_labels="{'region': 'primary', 'zone':
'east'}"
node02.rhkoso36.com openshift_node_labels="{'region': 'primary', 'zone':
'west'}"
```

4. 설치 및 실행

4.12 Advanced Installation 실행(5/5)

- Advanced Installer를 실행한다.

[Master Host #1] ansible-playbook 실행

```
# ansible-playbook [-i /path/to/inventory] \
  ~/openshift-ansible/playbooks/byo/config.yml
```

4. 설치 및 실행

4.13 설치 검증(1/3)

- 설치가 완료된 후, 아래의 명령을 수행하여 각 Host의 상태를 확인한다.

[Master Host #1] OpenShift Cluster 상태 확인

```
[root@master01 ~]# oc get nodes
NAME                     LABELS
STATUS                      AGE
master01.rhkoso36.com
kubernetes.io/hostname=192.168.122.181,region=infra,zone=default
Ready,SchedulingDisabled    1d
master02.rhkoso36.com
kubernetes.io/hostname=192.168.122.182,region=infra,zone=default
Ready,SchedulingDisabled    1d
master03.rhkoso36.com
kubernetes.io/hostname=192.168.122.183,region=infra,zone=default
Ready,SchedulingDisabled    1d
node01.rhkoso36.com
kubernetes.io/hostname=192.168.122.184,region=primary,zone=east     Ready
1d
node02.rhkoso36.com
kubernetes.io/hostname=192.168.122.185,region=primary,zone=west     Ready
1d
```

4. 설치 및 실행

4.13 설치 검증(2/3)

- 설치가 완료된 후, 아래의 명령을 수행하여 각 Host의 상태를 확인한다.

[Master Host #1] etcd cluster member list 확인

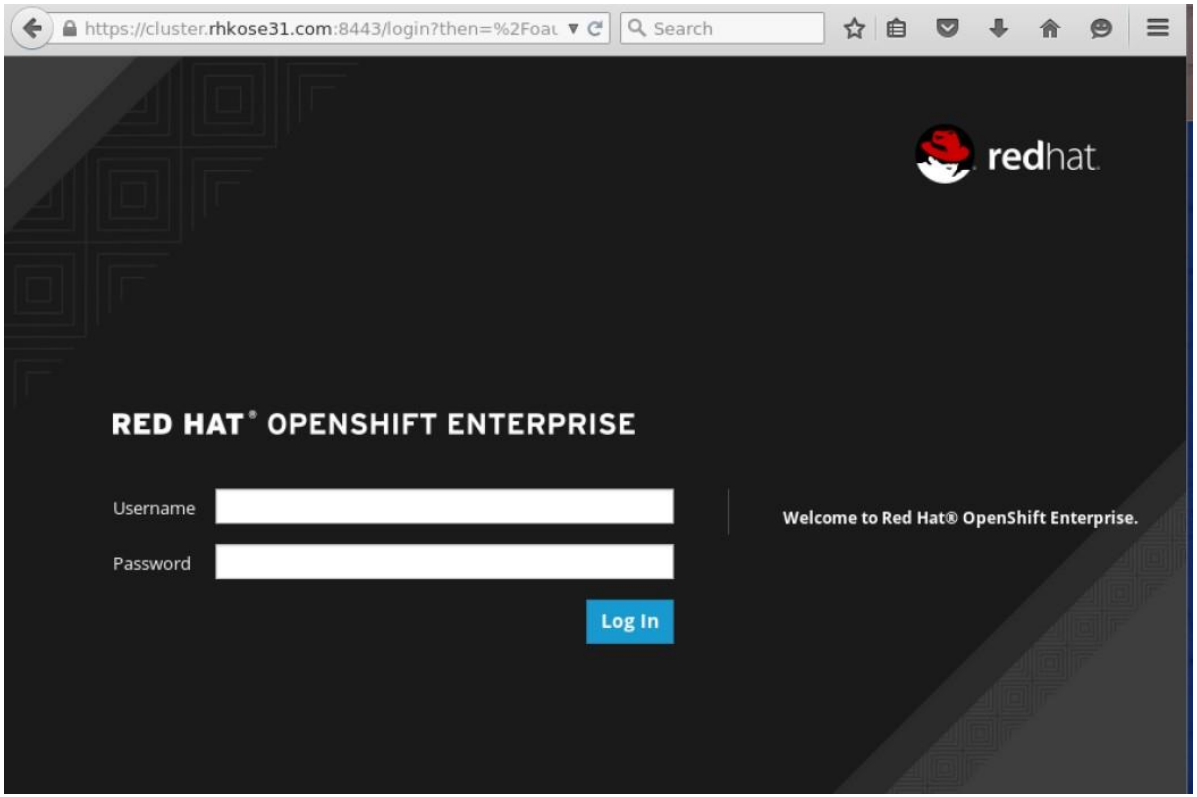
```
[root@master01 ~]# etcdctl -C \
https://master01.rhkoso36.com:2379,\
https://master02.rhkoso36.com:2379,\
https://master03.rhkoso36.com:2379 \
--ca-file=/etc/origin/master/master.etcd-ca.crt \
--cert-file=/etc/origin/master/master.etcd-client.crt \
--key-file=/etc/origin/master/master.etcd-client.key member list

1ed65af958c9b241: name=master01.rhkoso36.com
peerURLs=https://192.168.122.181:2380
clientURLs=https://192.168.122.181:2379
7dae59b2efd0ab2c: name=master02.rhkoso36.com
peerURLs=https://192.168.122.182:2380
clientURLs=https://192.168.122.182:2379
ebf2c93199c6f8a3: name=master03.rhkoso36.com
peerURLs=https://192.168.122.183:2380
clientURLs=https://192.168.122.183:2379
```

4. 설치 및 실행

4.13 설치 검증(3/3)

- 브라우저를 통해서 아래의 URL로 접속하면 아래의 화면을 볼 수 있습니다.
  https://cluster.rhkoso36.com:8443



5. 기능소개

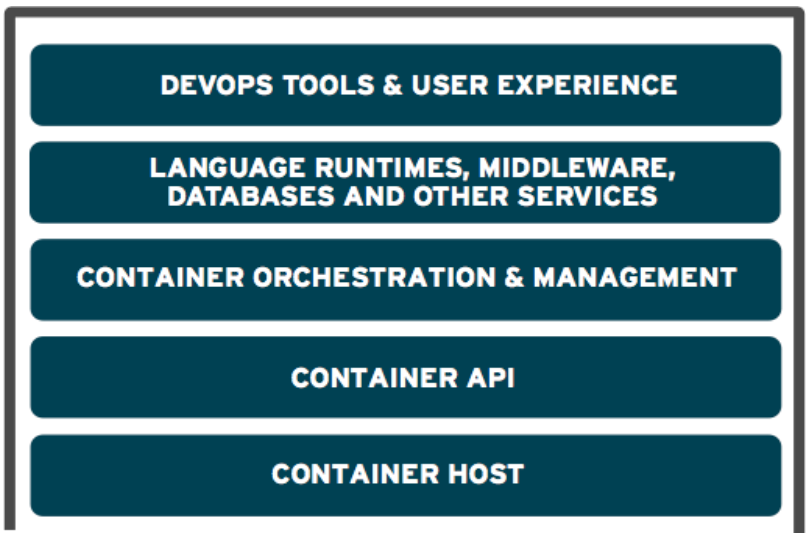
세부 목차

1. OpenShift Origin 기본 특징
2. Application 런타임 제공
3. JBoss MW Application 런타임 제공
4. Autoscaling 기능
5. WAS Cluster 기능
6. CI/CD 기능
7. Scheduling/Self-Healing
8. 자동 부하 분산
9. Service Discovery

5. 기능소개

5.1 OpenShift Origin 기본특징

OPENSHIFT origin



★ 개발 및 운영을 위한 다양한 툴 제공(IDE, UX 등)

★ Autoscaling 기능 제공

★ 다양한 application runtimes & services 제공

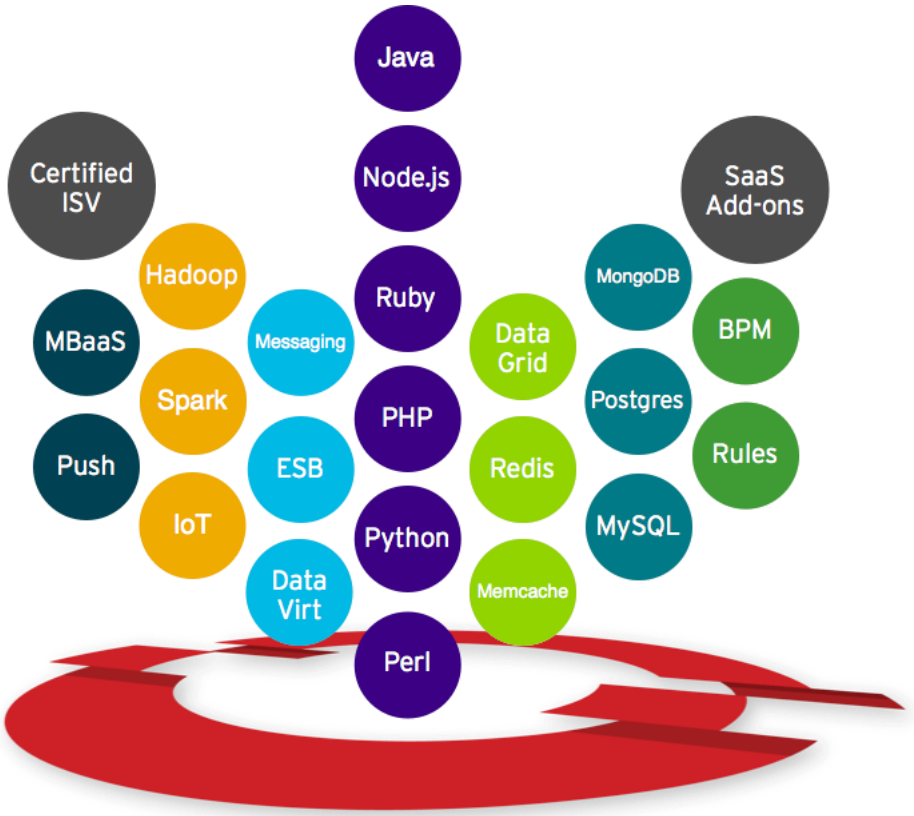
★ Docker 컨테이너 오케스트레이션 및 관리

★ OCI 표준 컨테이너 기반 – Docker

★ 컨테이너 최적화 OS(RHEL) 기반 혹은 오픈소스 OS(Fedora, CentOS) 기반

5. 기능소개

5.2 Application 런타임 제공



다양한 Application Runtime 제공

- From Red Hat
- From ISV Partners
- From the Community

5. 기능소개

5.3 JBoss MW Application 런타임 제공

Application Container Services

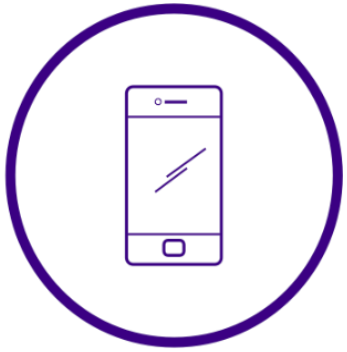
- JBoss Enterprise Application Platform
- JBoss Web Server / Tomcat
- JBoss Developer Studio

Business Process Services

- Business Process Management *
- Business Rules Management System *

Integration Services

- Fuse
- A-MQ
- Data Virtualization



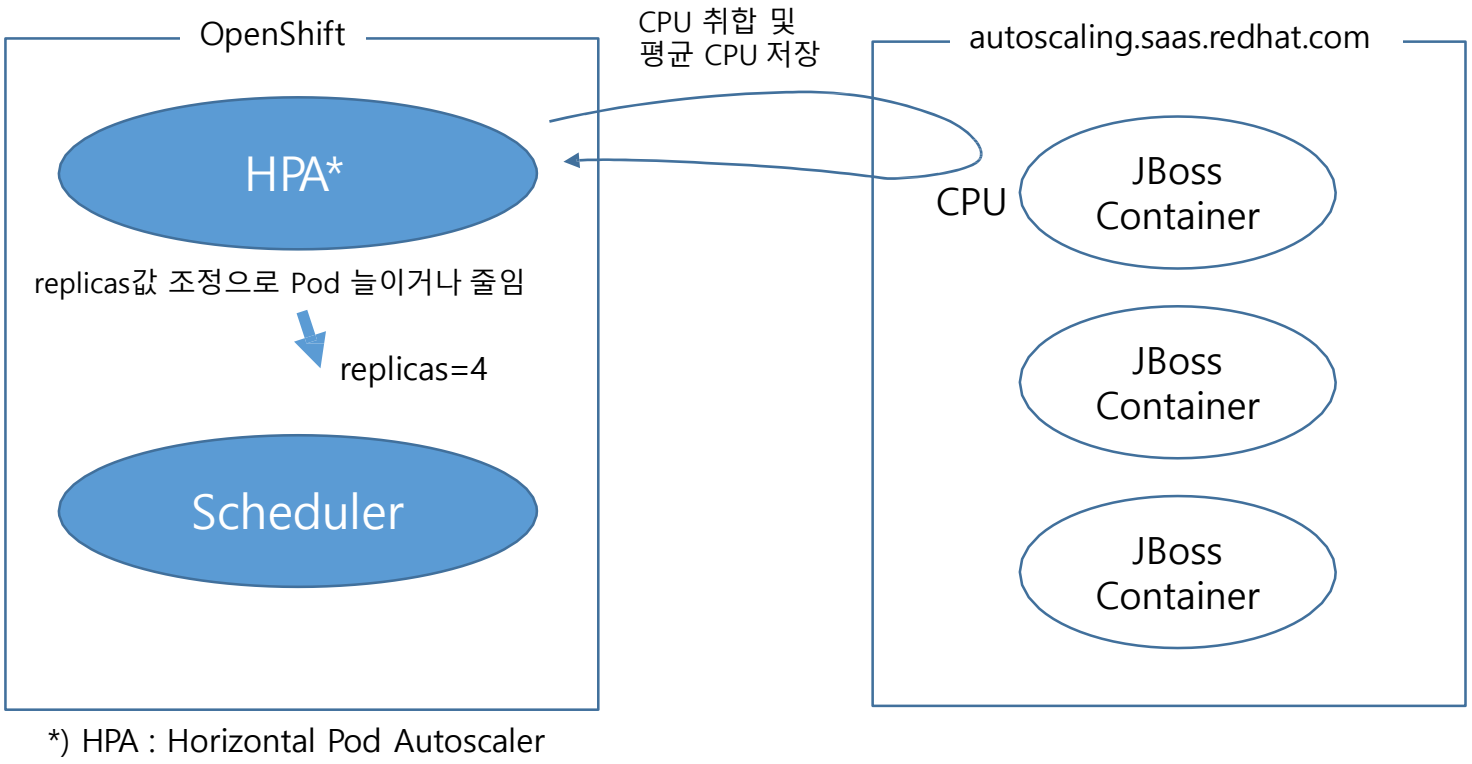
Mobile Services

- Red Hat Mobile / FeedHenry *

5. 기능소개

5.4 Autoscaling 기능(1/2)

HPA는 모니터링 대상 POD들을 모니터링하여 CPU값들을 가져와서 평균 CPU의 1분 추이를 확인하여, POD 개수를 결정한다.(알고리즘에서 상세 설명)
개수가 결정되면 HPA가 POD의 Target 개수를 설정하게 된다.



*) HPA : Horizontal Pod Autoscaler

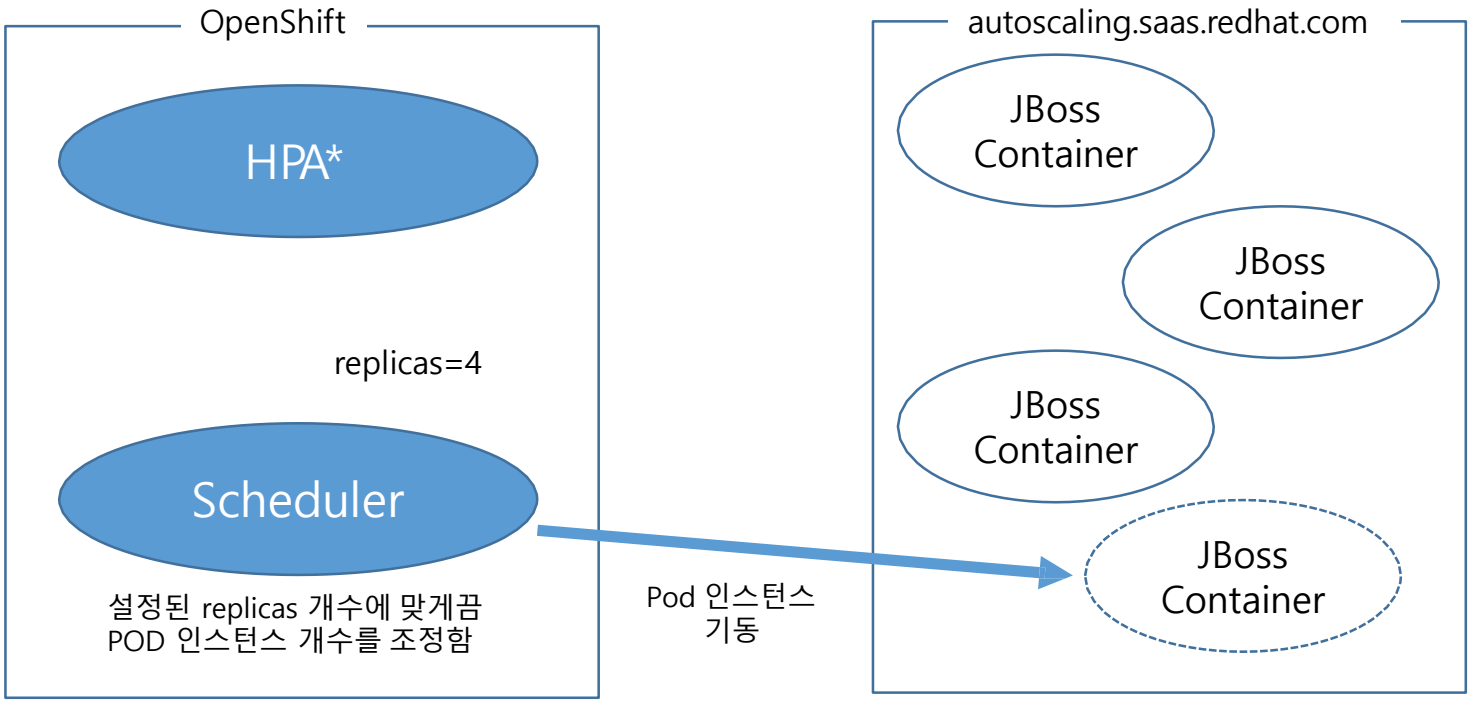
YouTube 동영상 링크 : https://youtu.be/HjYNNiW-f7A

5. 기능소개

5.4 Autoscaling 기능(2/2)

HPA는 POD의 replicas 개수만 설정하며, replicas 개수가 변경되면 OpenShift Scheduler가 현재 replica 개수와 새로 설정된 replica 개수를 비교하여 POD를 늘이거나 줄인다.

HPA에 의해서 POD Target 개수를 자동으로 결정하고 이를 반영하고, Scheduler가 Target 개수만큼 POD를 늘이거나 줄이는 것을 OpenShift 의 Auto Scaling(Scale-in, Scale-out) 이라 정의한다.

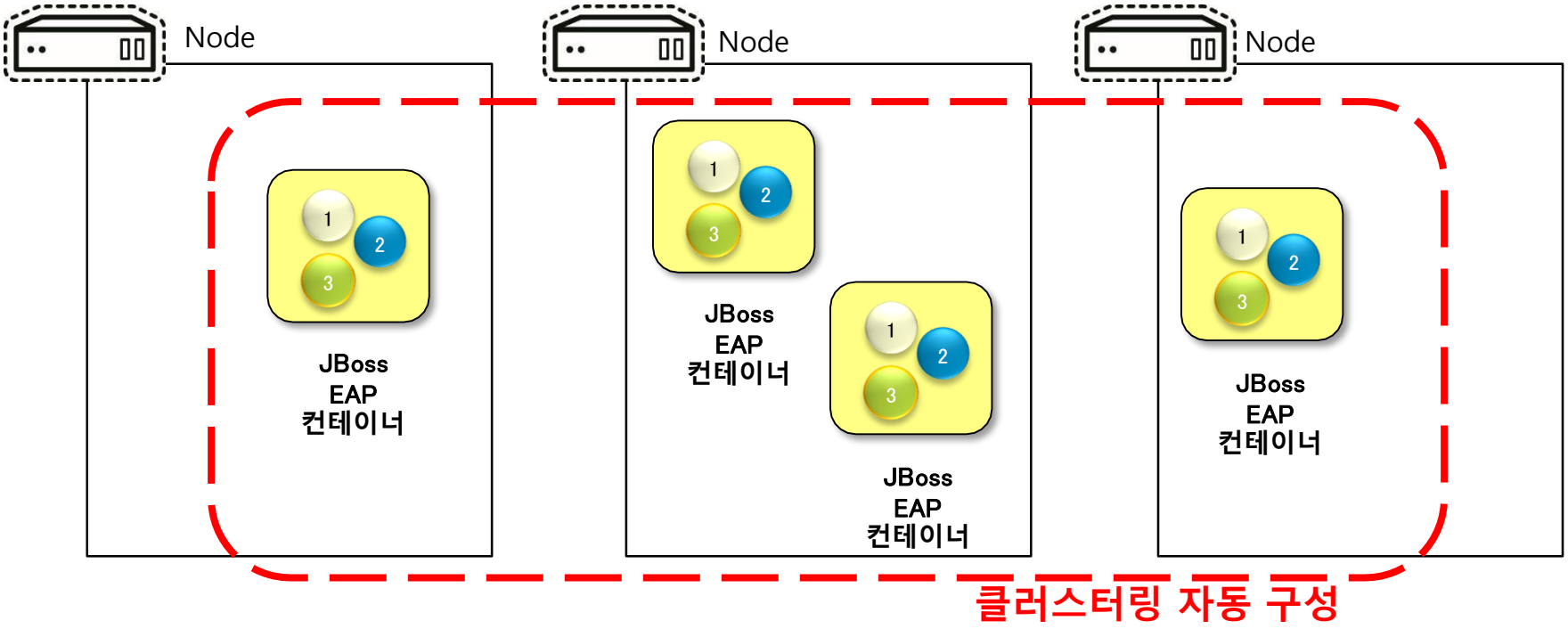


YouTube 동영상 링크 : https://youtu.be/HjYNNiW-f7A

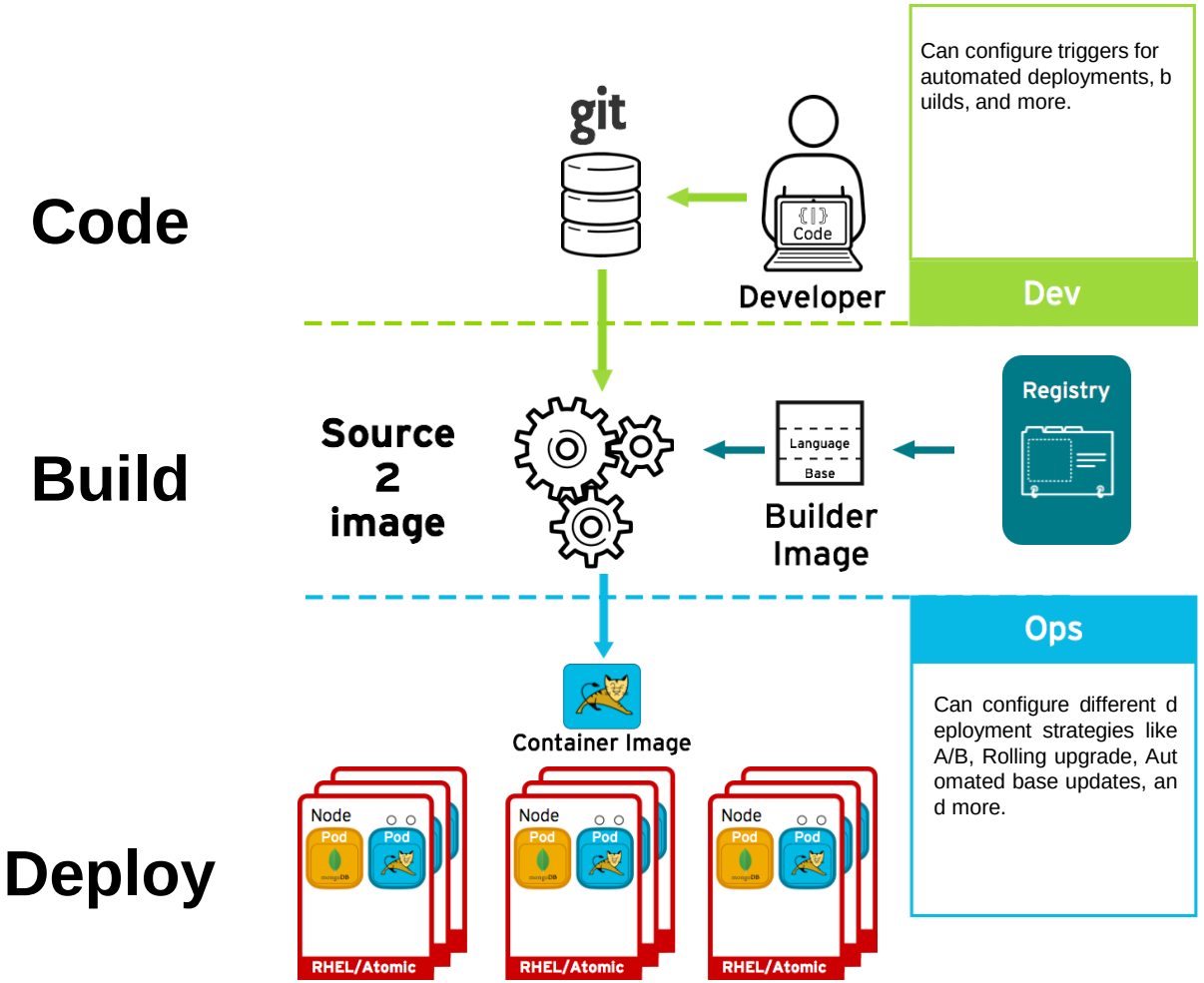
5.5 WAS Cluster 기능

JBoss EAP 컨테이너는 restart 시에 사설 IP가 자동으로 할당되기 때문에, IP가 변경될 수 있다. IP가 변경되면 WAS Clustering 기능을 사용하기가 어렵다. 특히, PaaS 환경상에서 WAS 자체의 클러스터링 기능을 사용하는 것은 매우 어려운 기술중 하나다.

OpenShift는 JBoss EAP WAS의 경우에, WAS의 기동 위치에 상관없이, 하나의 도메인 서비스(하나의 application 서비스)를 제공하는 JBoss EAP 컨테이너는, 간단한 옵션 설정만으로 Session Clustering(혹은 WAS Clustering)이 자동으로 구성된다.



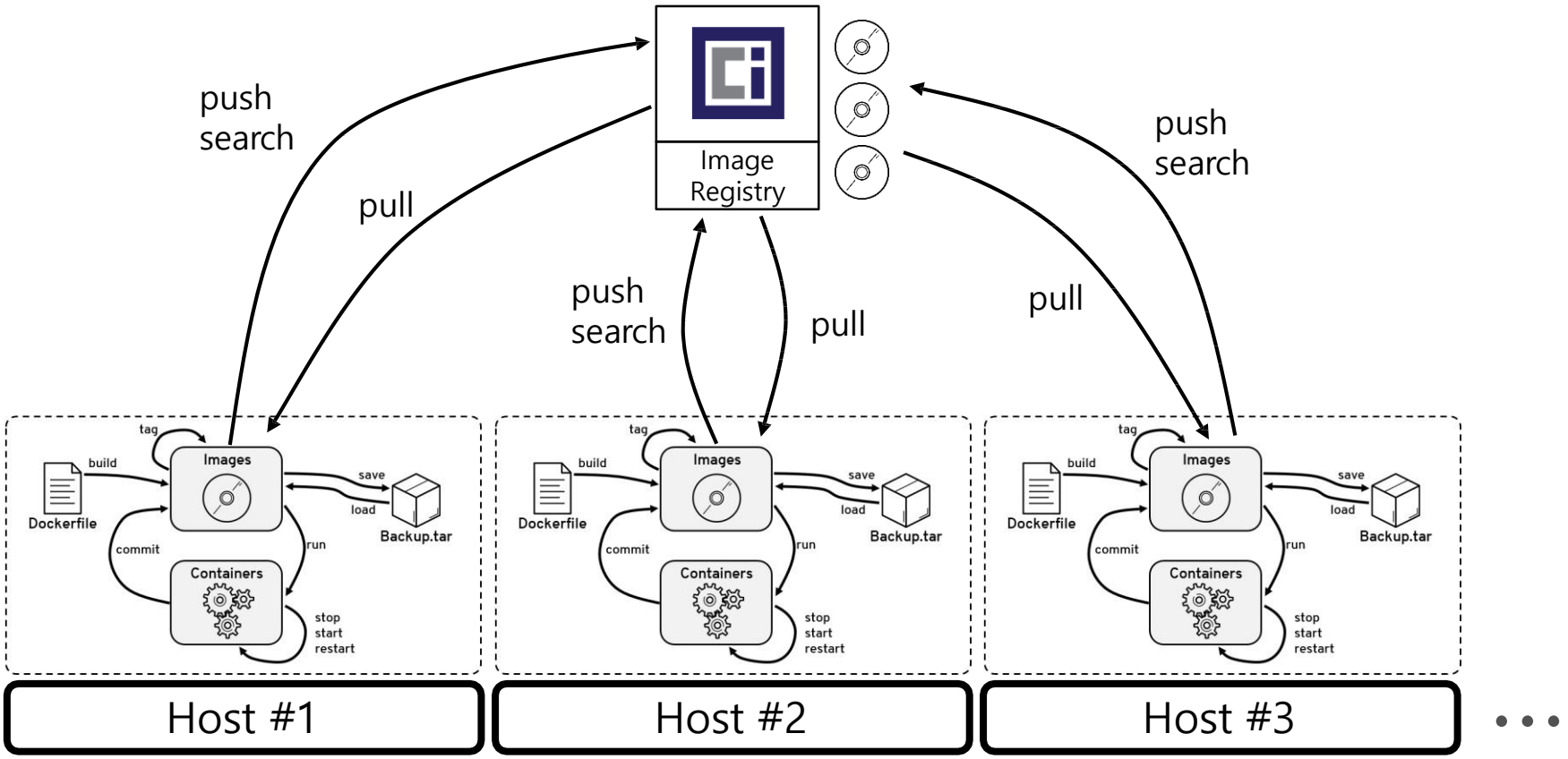
Git, Jenkins 컨테이너 이미지를 제공하여, 개발, 빌드, 배포, 테스트 및 운영환경 배포를 위한 CI/CD 프로세스 구성 및 커스터마이징이 가능하다.



5. 기능소개

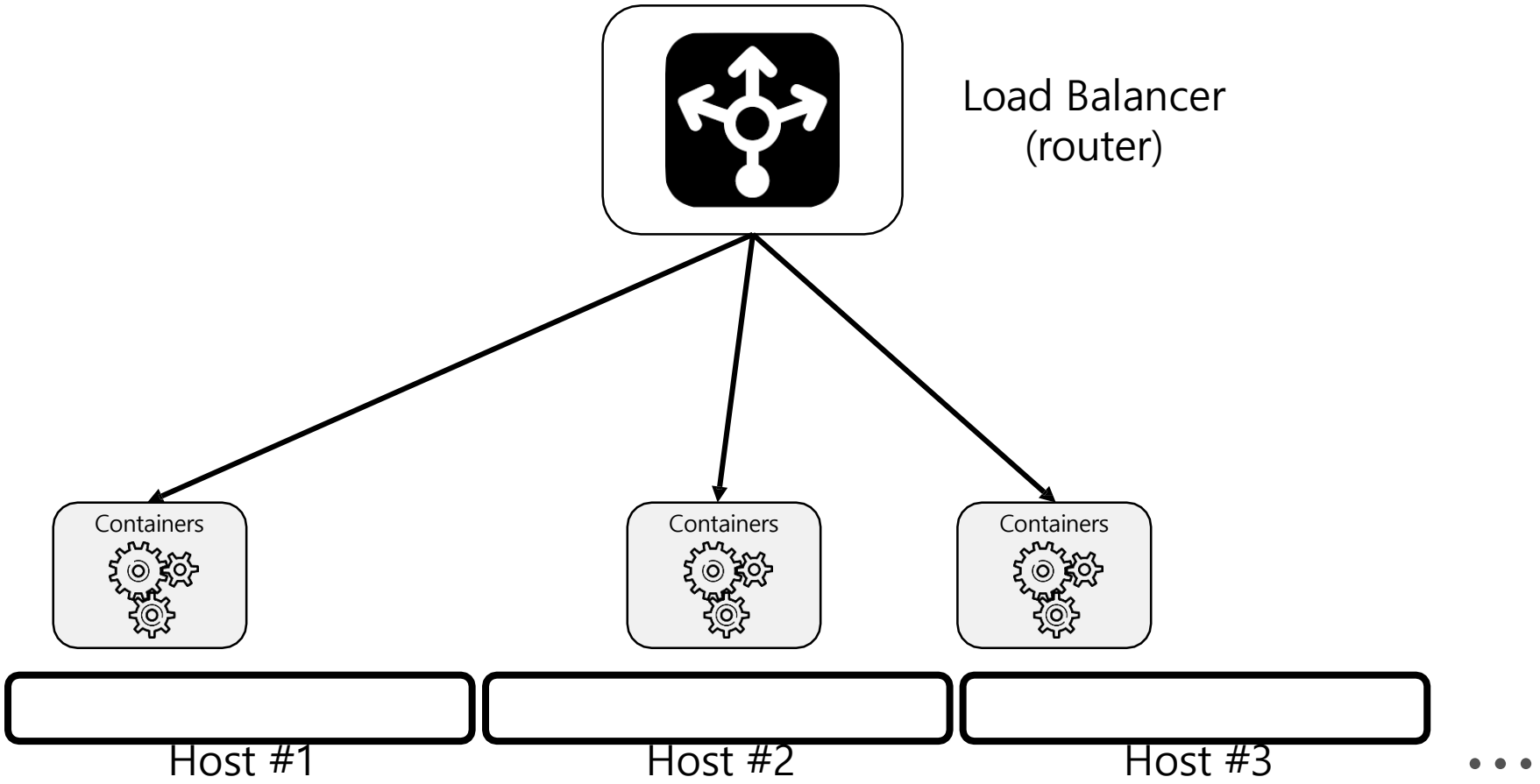
5.7 Scheduling/Self-Healing

멀티 Node(호스트)상의 멀티 컨테이너의 배치를 최적화해주는 Scheduling 기능과 배치된 컨테이너의 개수를 항상 유지시켜줌으로써 컨테이너가 Shutdown되었을 때 자동으로 재기동 시켜주는 Self-Healing 기능을 제공한다.



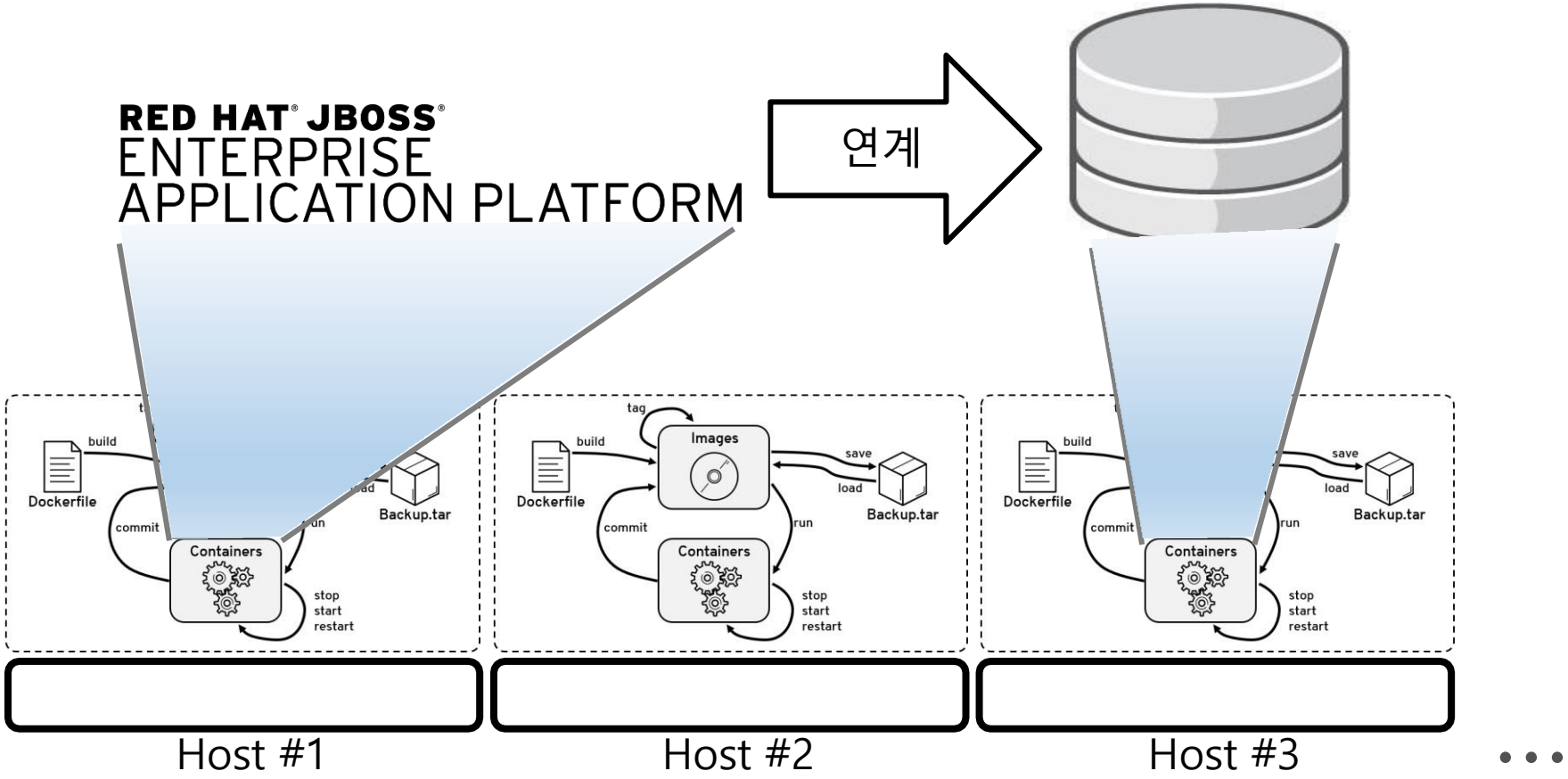
5.8 자동 부하 분산

Router 통해 들어오는 사용자의 요청을 자동으로 컨테이너 사이에 분산시켜서 부하를 분산시킴으로써 시스템의 안정적이 운영이 가능하다.



5.9 Service Discovery

OpenShift 상에 구동되어 있는 컨테이너 사이에 호출이 필요한 경우, 이들간에 서로 쉽게 찾을 수 있는 Service Discovery 기능을 제공한다.



6. 활용예제

세부 목차

1. 테스트용 ID/PW 등록
2. 로그인
3. Project 생성
4. 컨테이너 생성

6. 활용예제

6.1 테스트용 ID/PW 등록

- 모든 Master 장비에 접속하여 아래의 명령을 실행하여 Httpd-tools를 설치한다.

```
# yum install httpd-tools
```

- 모든 Master 장비에 접속하여 아래의 명령을 실행하여, ID/PW 를 신규로 추가한다.

```
$ htpasswd -c </path/to/users.htpasswd> <user_name>
```

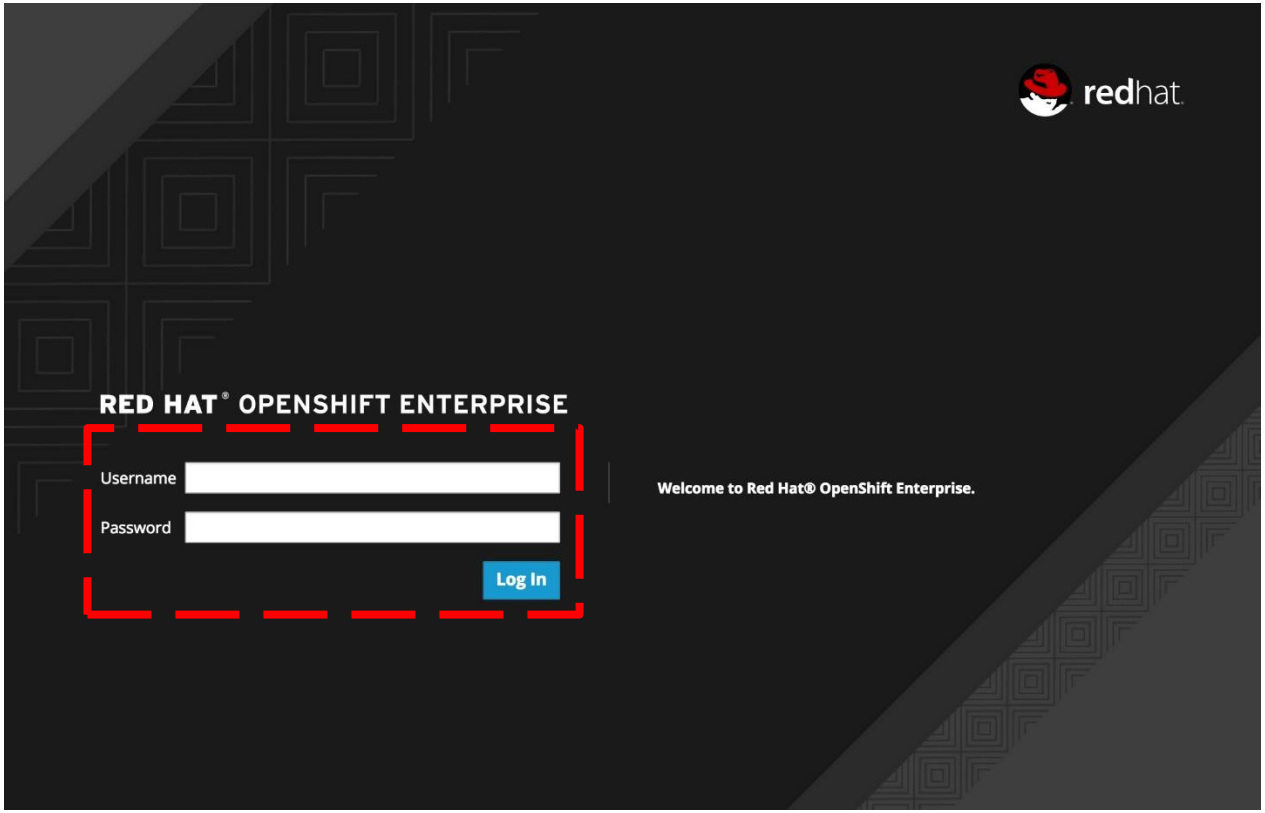
- 아래는 그 예제다.

```
htpasswd -c users.htpasswd user1
New password:
Re-type new password:
Adding password for user user1
```

6. 활용예제

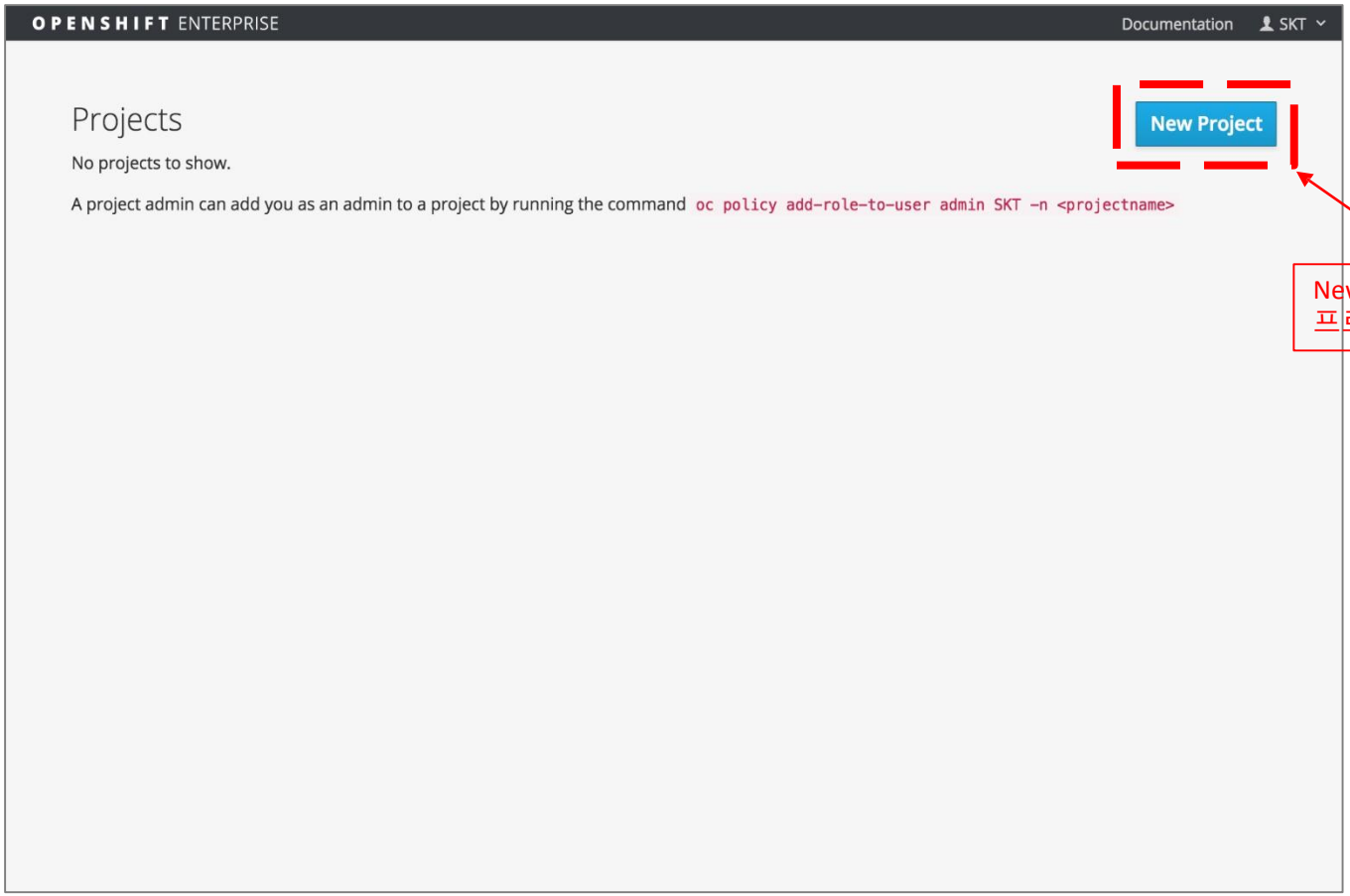
6.2 로그인

- 웹브라우저를 사용하여 아래의 URL에 접속하고, 새로 만든 ID/PW를 사용하여 로그인 합니다.
  https://cluster.rhkoso36.com:8443



6. 활용예제

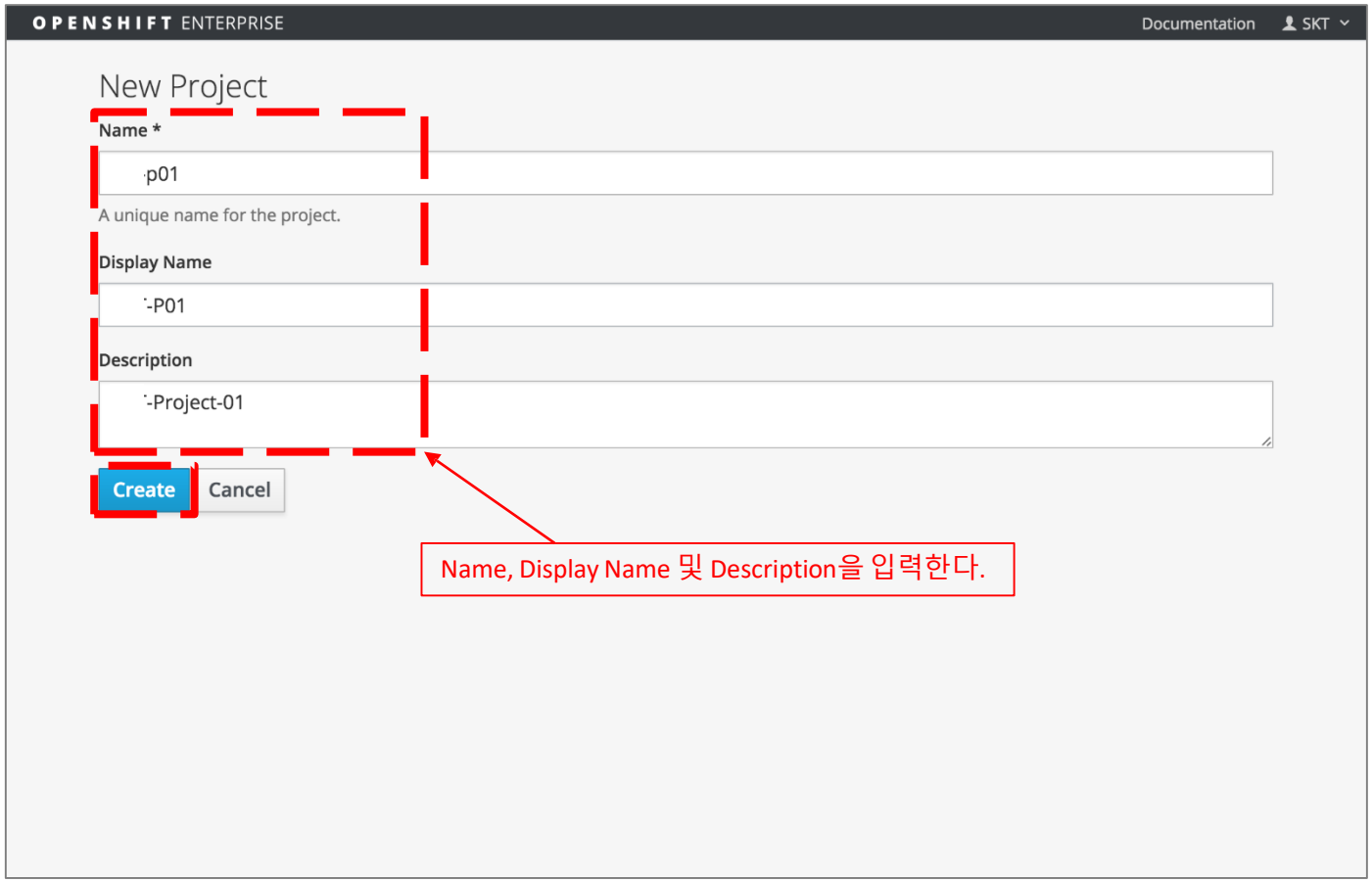
6.3 Project 생성(1/3)



New Project 버튼을 클릭하여 프로젝트를 새로 생성한다.

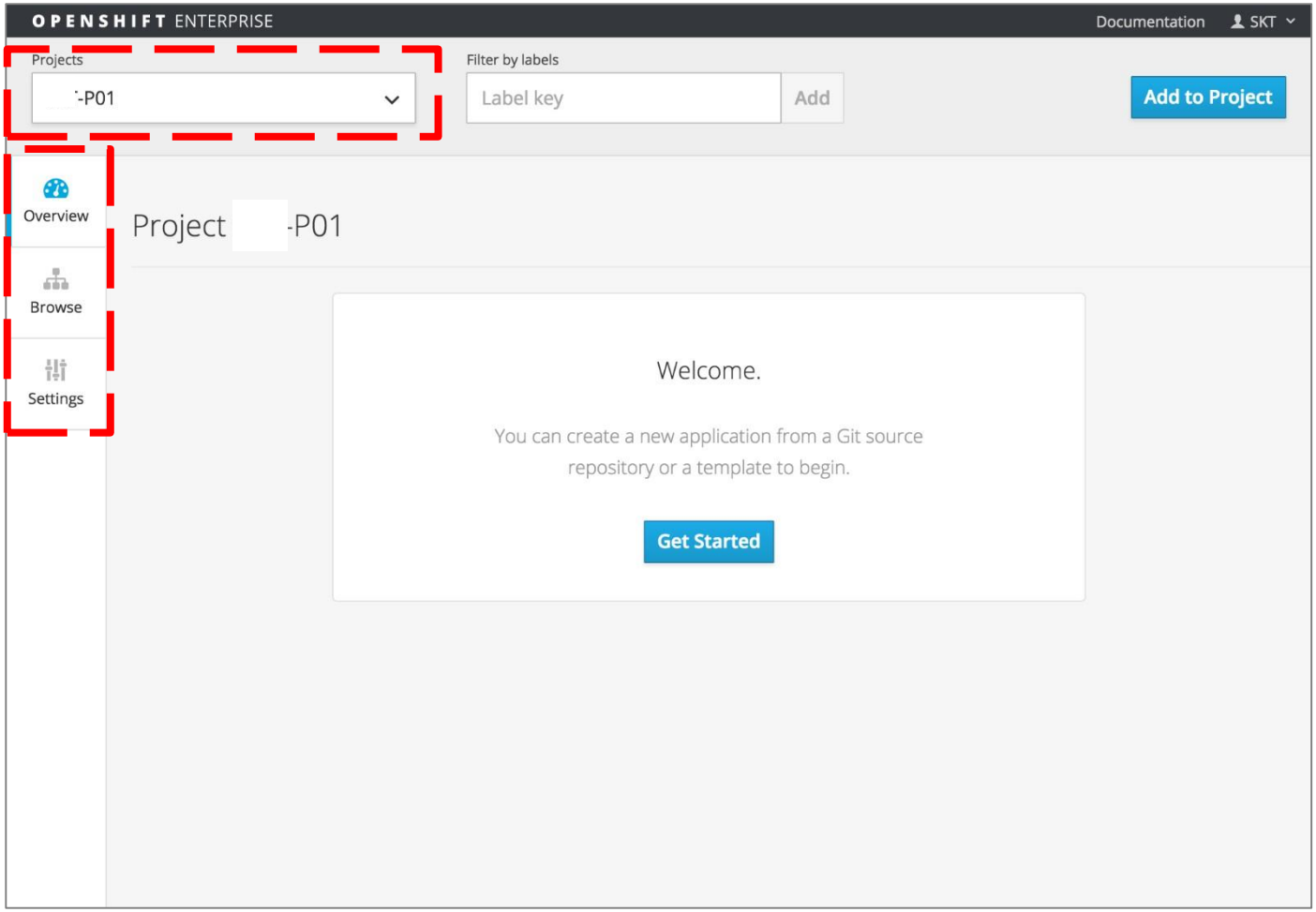
6. 활용예제

6.3 Project 생성(2/3)



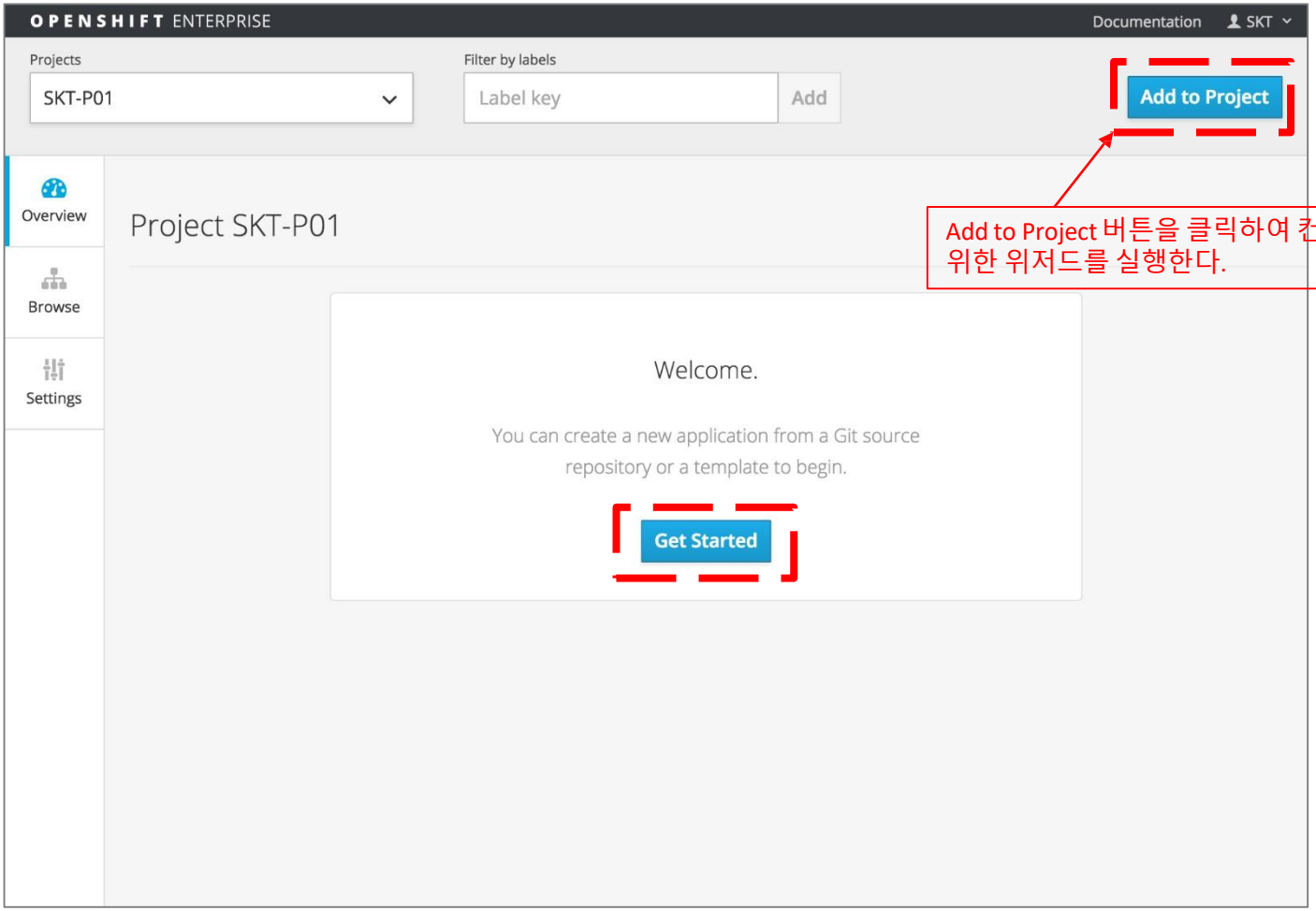
6. 활용예제

6.3 Project 생성(3/3)



6. 활용예제

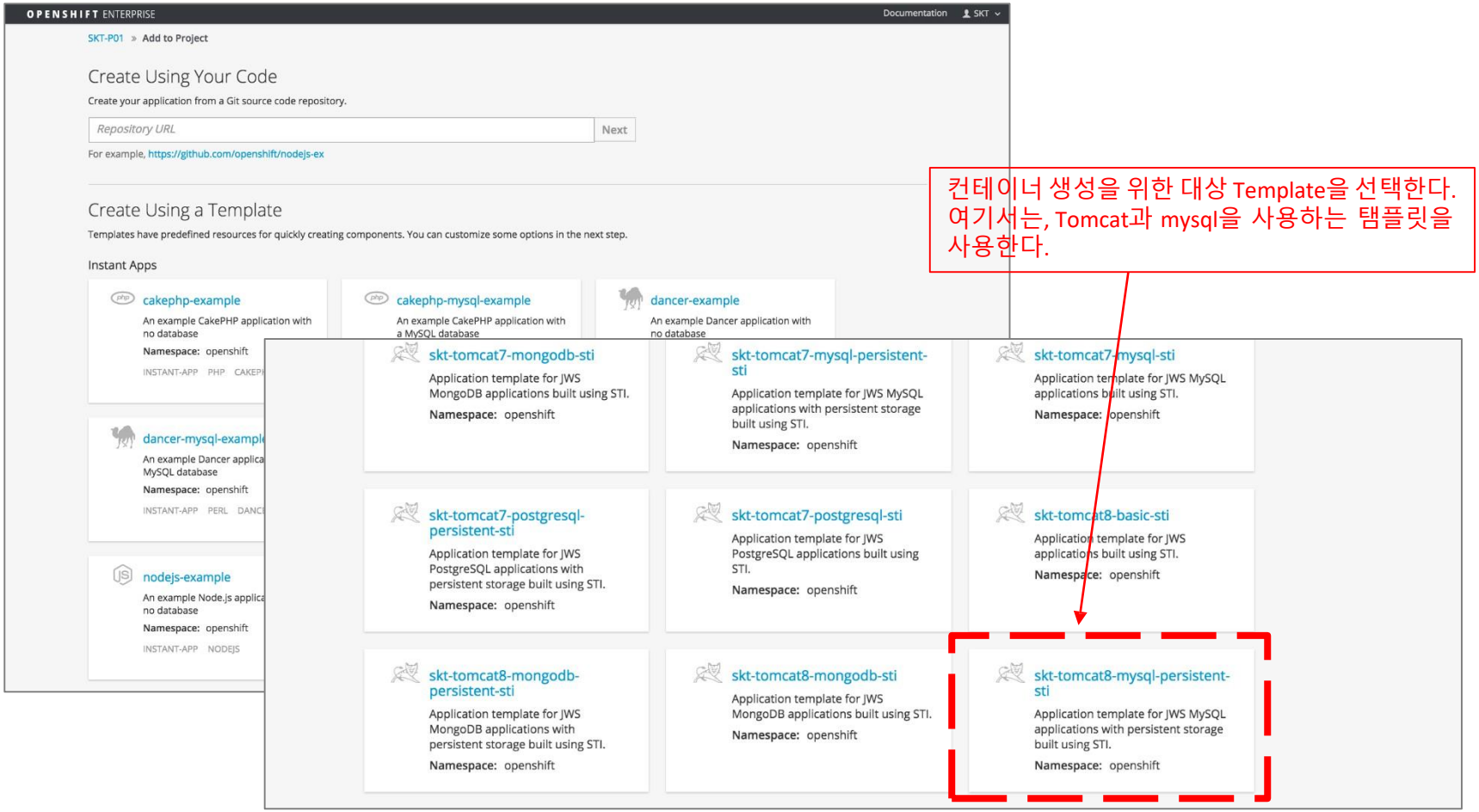
6.4 컨테이너 생성(1/5)



Add to Project 버튼을 클릭하여 컨테이너 생성을 위한 위저드를 실행한다.

6. 활용예제

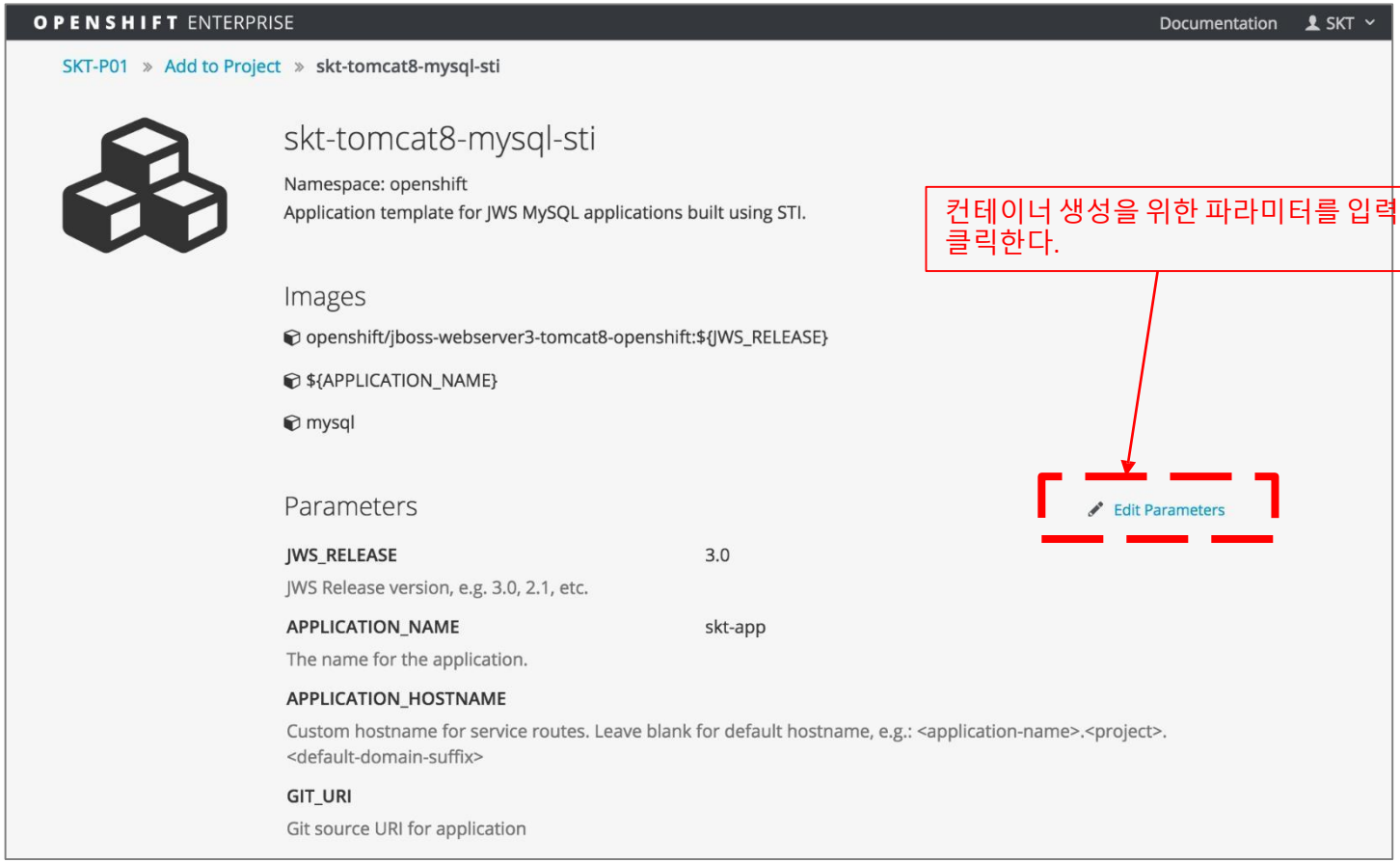
6.4 컨테이너 생성(2/5)



컨테이너 생성을 위한 대상 Template을 선택한다. 여기서는, Tomcat과 mysql을 사용하는 탬플릿을 사용한다.

6. 활용예제

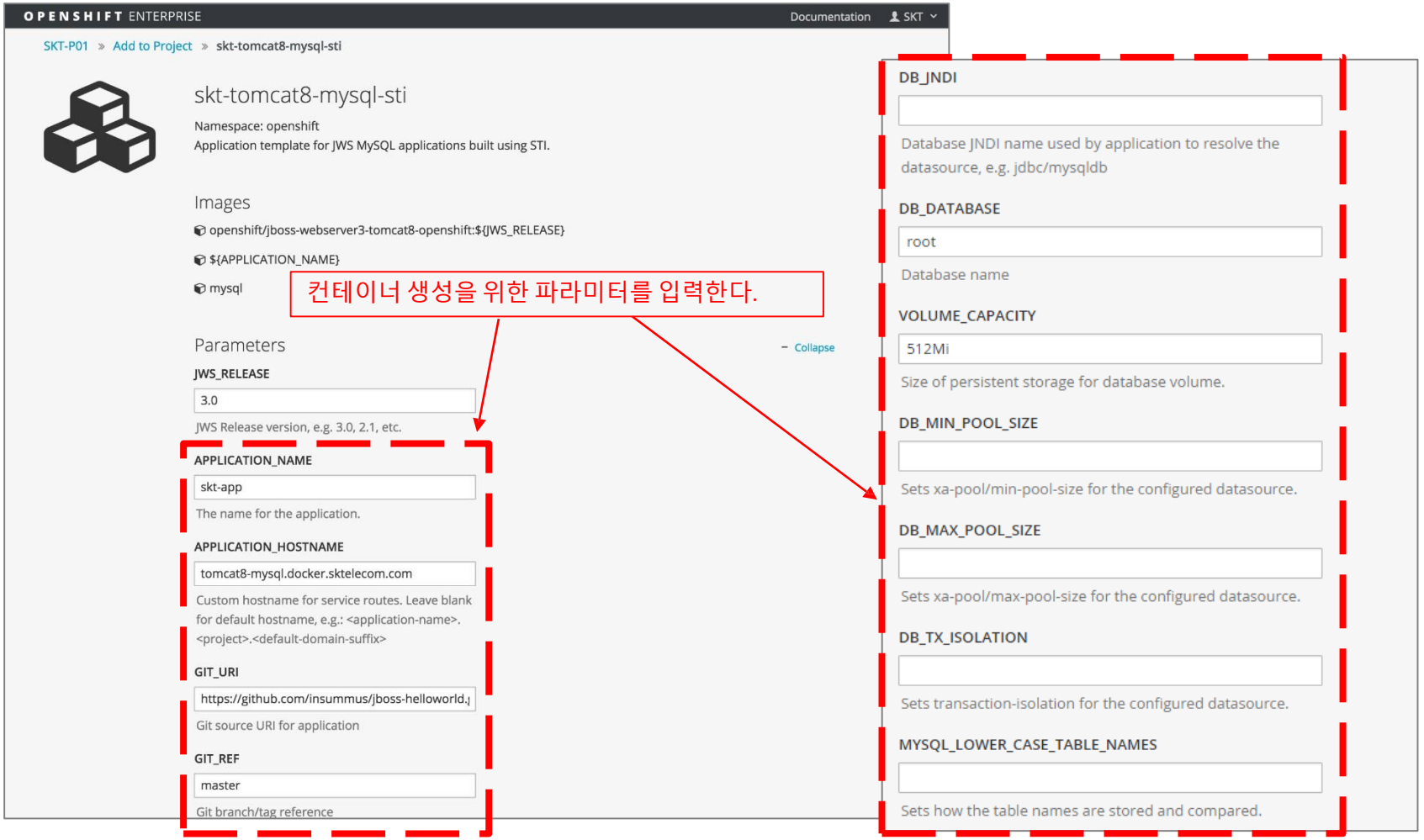
6.4 컨테이너 생성(3/5)



컨테이너 생성을 위한 파라미터를 입력하기 위해서 클릭한다.

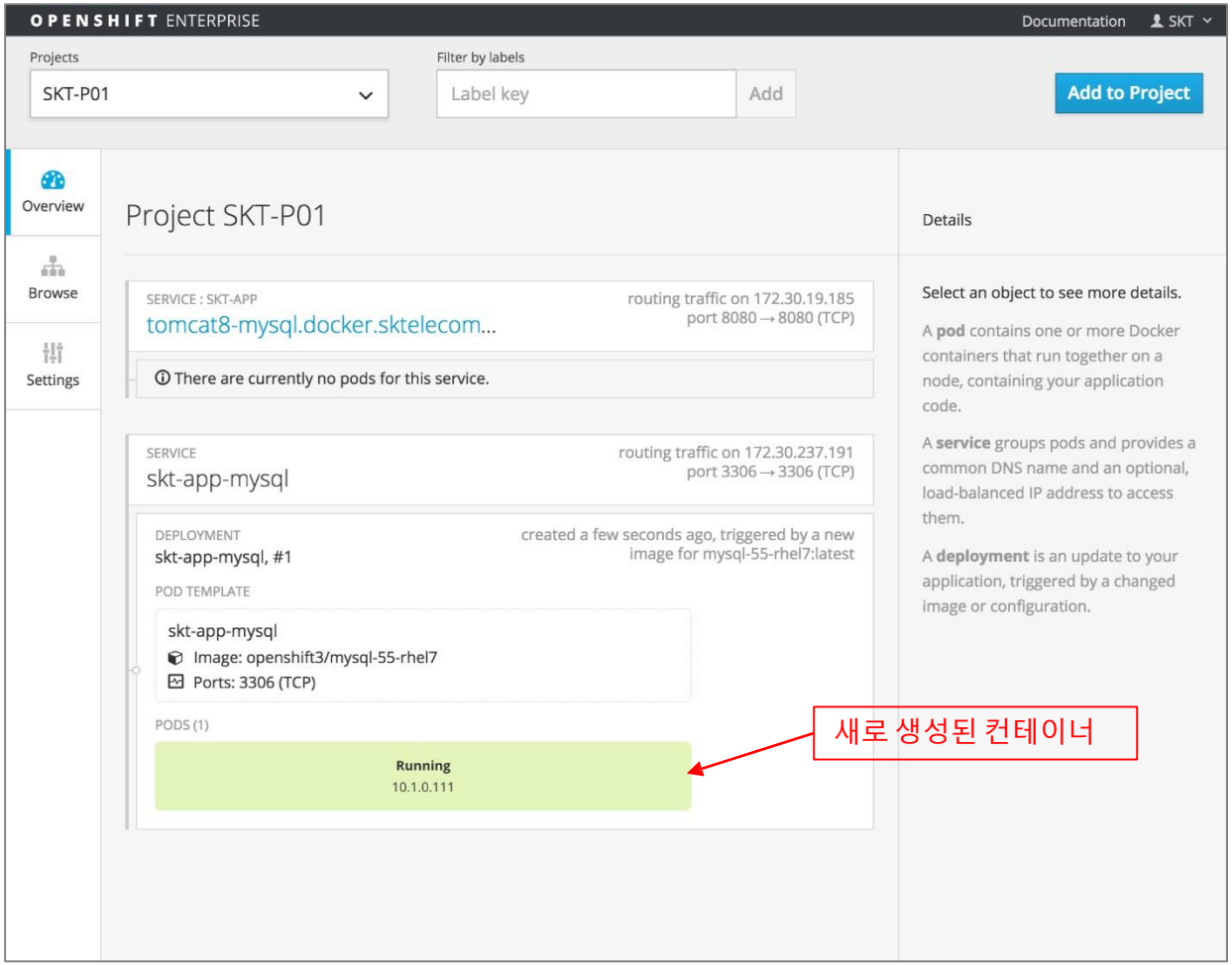
6. 활용예제

6.4 컨테이너 생성(4/5)



6. 활용예제

6.4 컨테이너 생성(5/5)



7. FAQ

Q **OpenShift는 어떤 사용자를 위한 것인가요?**

A OpenShift는 공용 또는 사설 PaaS 클라우드를 구축하려는 서비스 제공 업체, 기업, 정부 및 교육 기관을 위한, 표준 컨테이너 기반 컨테이너 플랫폼 소프트웨어입니다. 산업 분야는 IT 및 통신 업체에서 SaaS 및 전자 상거래, 금융 및 의료에 이르기까지 다양합니다.

Q **Kubernetes와 다른 점은 어떤 것인가요?**

A OpenShift는 Kubernetes를 포함하고 있으며, Kubernetes가 제공하지 못하는 다양한 기능(예를 든다면 CI/CD 기능, 빌드/배포 기능, SDN 기능)을 제공하고 있으며, 엔터프라이즈급의 안정성을 제공하고 있습니다.

8. 용어정리

용어	설 명
Application	OpenShift v3 에는 특정 애플리케이션 용어나 개념이 더 이상 존재하지 않는다.
Cartridges vs. Images	OpenShift v3 에서 카트리지(Cartridge)를 가장 쉽게 교체 할 수 있는 용어는 이미지(Image)다. 이미지는 패키징 관점에서 볼 때 카트리지 이상을 차지하므로 캡슐화와 유연성이 향상된다. 그러나 카트리지 개념에는 이미지에 존재하지 않는 빌드, 배포 및 라우팅을 위한 논리도 포함되어 있다. OpenShift v3 에서 이러한 추가 요구 사항은 S2I (Source-to-Image) 및 템플릿 구성을 통해서 해당 기능을 제공한다.
Domain vs. Project	OpenShift v3 에서 소개되는 프로젝트(Project) 라는 단어는 기본적으로 OpenShift v2 의 도메인(Domain) 단어를 대체한다. 프로젝트에는 OpenShift v2의 도메인에 속하지 않는 몇 가지 기능이 있다.
Gear vs. Container	OpenShift v2 에서 사용하던 기어(Gear)라는 용어는 OpenShift v3에서는 컨테이너(Container)라는 용어로 변경되었다. 컨테이너는 이미지와 일대일로 깔끔하게 맵핑되는 반면, 많은 카트리지가 단일 장비에 추가될 수 있다. 컨테이너의 경우 배열 개념이 POD 에 의해 충족된다.
Broker vs. Master	OpenShift v3 의 마스터(Master)는 OpenShift v2에서 브로커(Broker) 를 대체한다. 또한, OpenShift v2에서 브로커가 사용하는 MongoDB 및 ActiveMQ 계층은 더 이상 필요하지 않다. 키 값 저장소 etcd가 일반적으로 각 마스터와 함께 설치되기 때문이며, 모든 관리 정보는 etcd에 저장된다.

Open Source Software Installation & Application Guide