SOLUTION PARTNER FOR **SMART TECHNOLOGY**

LENA

Installation

LENA Support

Version 1.3.1.10

Table of Contents

Chapter 1. OVERVIEW

본문서는 Container 기반 LENA Server를 운영하기 앞서 필요한 Architecture결정요소와 설치에 대해 기술한다. LENA 의 전체 기능 및 운영에 대한 내용은 별도로 제공되는 운영자 매뉴얼을 참고한다.

본 문서는 LENA 1.3.1c 버전을 기준으로 기술하고, 다음과 같은 내용을 포함한다.

- LENA for Container 아키텍처 결정요소
  - 적용환경에 대한 고려사항
  - Server 유형별 고려사항
- LENA for Container 설치
  - Base Image Build
  - Kubernetes 기반 설치
  - Docker 기반 설치
  - ECS 기반 설치
  - VM / Host 기반 설치 (Manager, Session Server)

1.1. 구성요소

LENA는 Web Server, Application Server, Session Server와 Web Server의 Status를 확인하는 Node Agent, Application Server에 설치되어 Status정보를 제공하는 Advertiser와 관리자에게 제공되는 통합관리 도구인 Manager로 구성된다.

1.1.1. Server

LENA에서 제공되는 서버의 종류는 Web Server, Application Server, Session Server 3가지가 있다. 각 서버의 용도는 아래와 같다.

1. Web Server: 사용자 요청에 따라 Web Resource를 제공한다. Application Server가 제공하는 응용서비스의 Front역할을 수행하면서, 선택적으로 Load Balancing 및 보안 레이어(SSL)를 제공하는 역할을 수행한다.
2. Application Server: Java로 작성된 응용 서비스를 실행/제공 한다.
3. Session Server: Application Server간 사용자의 세션을 유지한다.

1.1.2. Agent, Advertiser

Node, Server에 설치되어 제어 및 모니터링 기능을 담당하는 Agent 이다.

- Node Agent
  - Web Server 상태 모니터링 데이터를 취합하여 Manager에게 제공한다.
- Advertiser
  - Application Server 상태 모니터링 데이터를 취합하여 Manager에게 제공한다.

1.1.3. Manager

Manager는 Node Agent와 Advertiser를 통하여 Node와 Server의 제어 및 모니터링 기능 등을 제공하는Web Application이다. 대표적으로 아래와 같은 기능을 제공한다.

항목	설명
Dashboard	• Server, Service Cluster 현황 • Notification 확인
Server	• System (논리적 Server 그룹) 등록/수정/삭제
Service Cluster	• Service Cluster 등록/수정/삭제 • 등록된 Server 목록, 이력 조회 • Service Cluster 템플릿, Revision 관리 • 설정 Template 다운로드를 통한 CICD 연계 기능 • 원격 Terminal 및 Standard Out/Error Log 조회 (Kubernetes에 한함)
Resource	• Resource (Database, DataSource, Application,k8s config) 등록/수정/삭제/조회 • Resource를 사용하는 Server 목록 조회 및 추가/제거
Diagnostics (모니터링)	• Server에 대한 이슈 현황 모니터링 기능 • Server에서 발생한 Event 조회 기능
Topology	• System별 Server 구성현황 조회
Admin	• 사용자 및 권한 관리, 사용자/권한/메뉴 매핑 • 사용자 운영 이력 조회 • 라이선스 관리, 현황 조회 및 업로드 • Cloud Profile 관리

1.2. Mechanism

LENA는 Manager를 통해서 Web/Application 서버를 모니터링 및 통합 관리하는 기능을 제공한다. 다만, 기존 Host/VM방식과의 환경적 차이점은 Container에서 구동되는 Server는 Orchestration도구에 의해 실행이 제어되며, State를 가지지 않는다는 점이다.

따라서, 기존 Host/VM 환경에서 Agent를 통한 실시간 제어/설정 관리 방식 대신 개별 Server가 Container기동 시점에 Manager를 통해 설정정보 및 라이선스를 다운로드 받는 방식으로 설정정보를 제공하고, 기동된 Server의 상태를 Manager를 통해서 모니터링 하는 방식으로 관리한다.

Server외의 구성요소로는 기동 시 다운로드 및 초기설정 기능을 사용하기 위해서는 각 Server는 Container시작 Command로 활용되는 docker-entrypoint.sh가 있고, 운영되는 Server의 상태를 전송하기 위해 Web 서버에는 Node Agent가 설치되고 Application 및 Session Server는 내장된 모듈을 활용한다.

WEB-WAS 연계 부분에서도 기존 VM/Host 환경과의 차이가 존재한다. Container의 생성/소멸에 따라 IP나 주소가 변동적인 환경에서 Back-End Application Container와의 지속적인 연결을 유지하기 위해서는 Load-balancer가 필요하며, 이는 Kubernetes/Docker Swarm의 Service나 ECS의 Service Discovery, EKS/ECS의 ELB 형태로 제공되고 있다. 기존 VM/Host 환경에서 WEB-WAS를 직접 연결하고 WEB 서버가 직접 Load-balancing을 수행했던 방식은 Container환경에서는 제공되지 않고 플랫폼이 제공하는 Service 주소(Service

Endpoint)도 Reverse Proxy 연결을 제공한다.

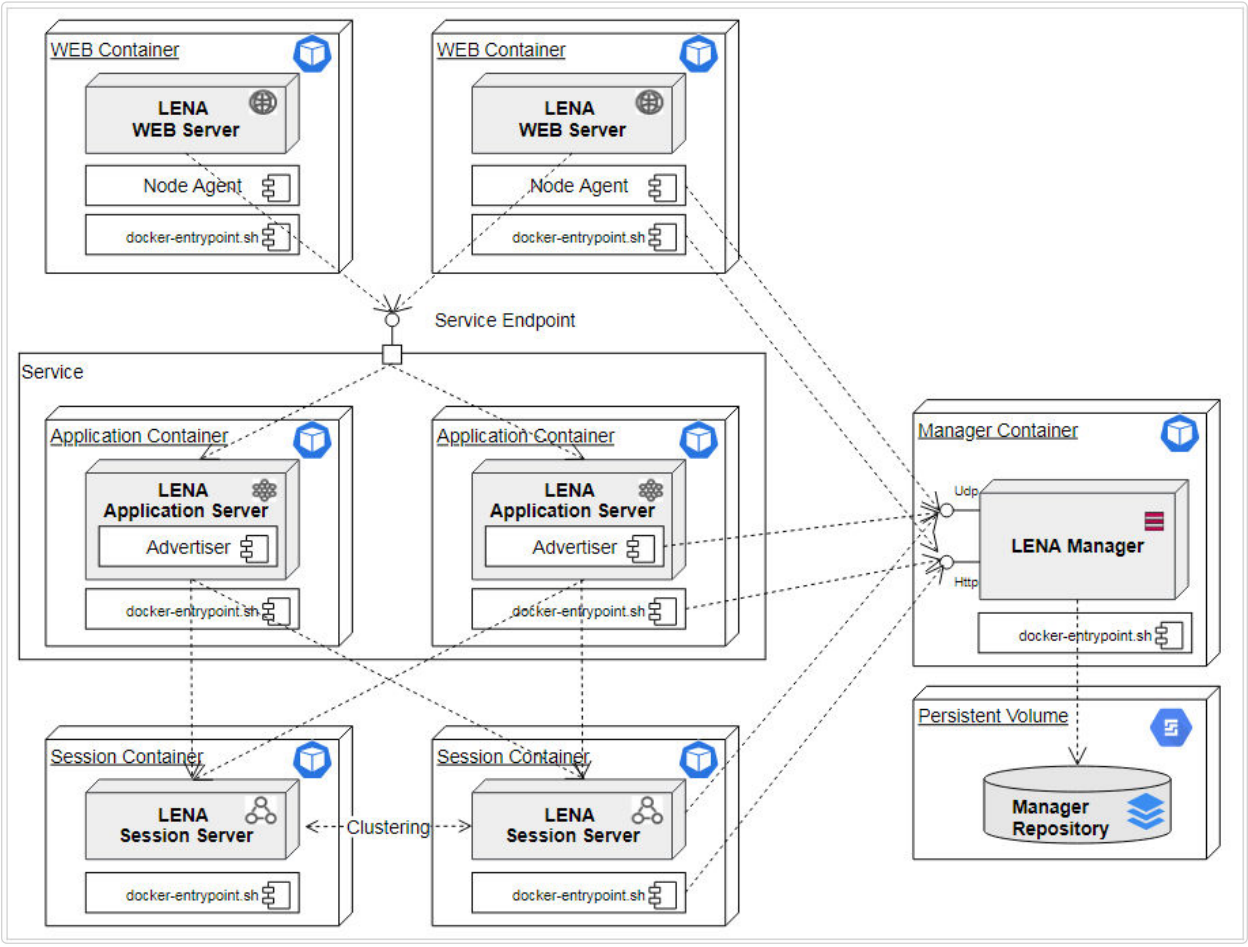


Figure 1. Container 기반 LENA Server간 연계 구조 (Kubernetes 환경)

항목	설명	비고
Application Server	Application Server Instance	
Web Server	Web Server Instance	
Session Server	Session Server Instance	
Manager	서버에 배포되는 설정파일 관리 및 Server 모니터링 기능 제공	
Manager Repository	Manager 운영을 위한 파일저장 Repository, 각 종 설정정보 및 DB 정보를 포함함	외부저장소도 분리가능
docker-entrypoint.sh	Container 기동시에 실행되는 Shell Script 1. 기동시점에 설정정보 초기화 2. Manager도 부터 설정정보/라이선스 다운도드 3. 서버 기동 기능을 수행함	

항목	설명	비고
Node Agent	Web 서버 모니터링 데이터 취합 및 Manager에게 송신, Manager로부터 수신한 제어/설정 명령 실행	
Advertiser	모니터링 데이터 취합 및 Manager에게 송신	Application Server에 동합

Container 운영환경의 특성이나 제약에 따라 Session Server 나 Manager를 VM/Host환경에서 운영할 수 도 있고, LENA Manager는 Container형 LENA Server 뿐아니라, VM/Host 기반 LENA Server를 관리하는 기능도 포함하고 있으므로, 다음과 같은 아키텍처로 운영될 수도 있다.

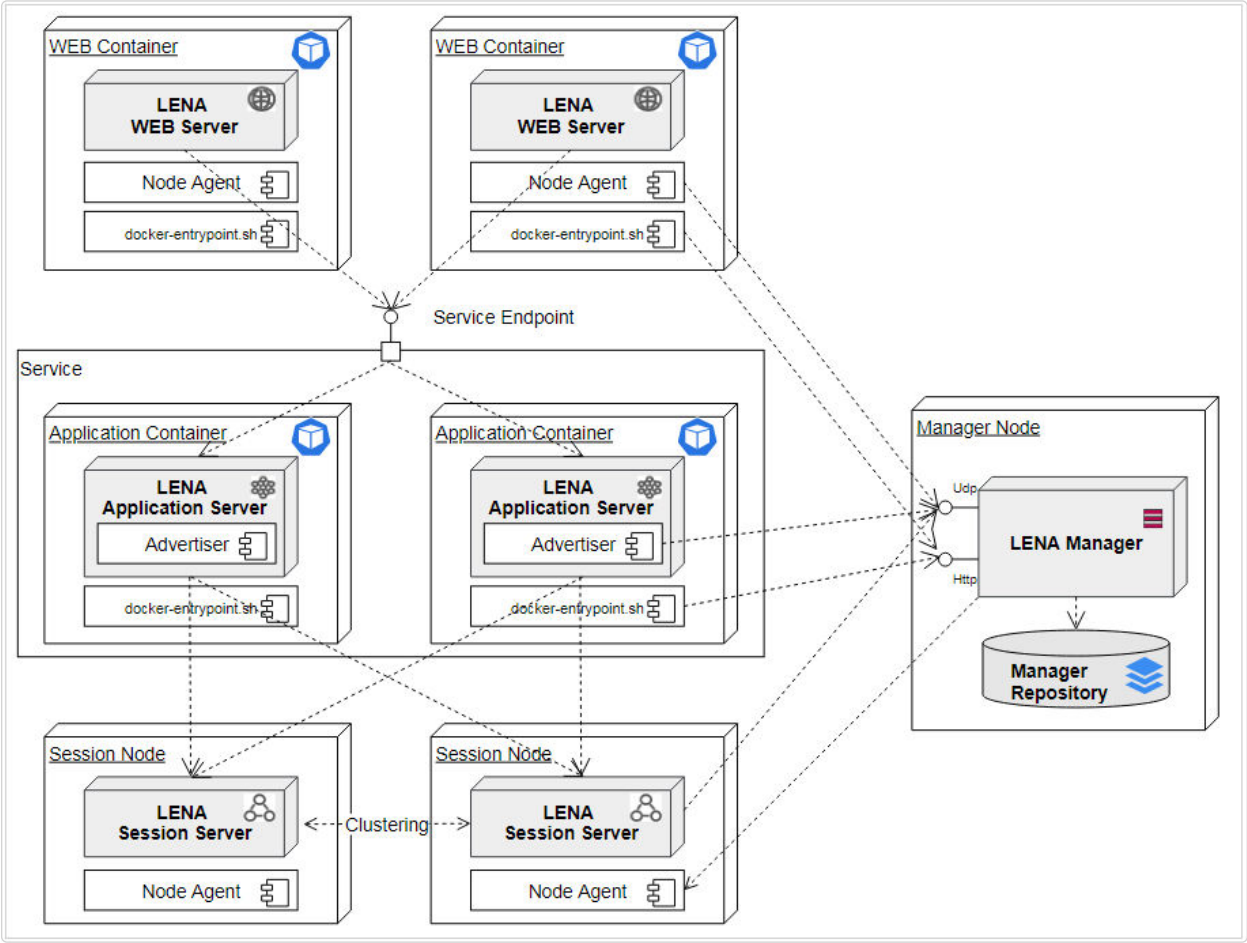


Figure 2. Container -VM/Host 혼합 환경에서의 LENA Server간 연계 구조

1.3. 제공 Asset

LENA for Container 버전에서는 다음과 같은 Asset 제공된다.

- Docker Image : Linux OS + JDK + LENA Server + 필요 Library가 포함된 Image를 Docker Hub를 통해서 제공
  - Web Server : https://hub.docker.com/r/lenacloud/lena-web
  - Application Server : https://hub.docker.com/r/lenacloud/lena-cluster
  - Session Server : https://hub.docker.com/r/lenacloud/lena-session
  - Manager Server : https://hub.docker.com/r/lenacloud/lena-manager
- Kubernetes Manifest 파일 : Kubernetes에 설치 시 필요한 Workload / Service / Config Map 이 기술된 LENA Server배포용 파일

Docker Hub에서 제공하는 Image의 Spec은 다음과 같다.

구분	JVM	OS (Base Image)	기본 Heap Memory
Application Server	Open JDK 1.8	• Cent OS 7 (centos:7)	1.0 GB
Web Server	Open JDK 1.8	• Cent OS 7 (centos:7)	64MB~256MB(Agent)
Session Server	Open JDK 1.8	• Cent OS 7 (centos:7)	1.0 GB
Manager	Open JDK 1.8	• Cent OS 7 (centos:7)	1.0 GB

1.4. 시스템 요구사항

LENA for Container의 각 서버 인스턴스 설치에 대한 최소 요구사항은 다음과 같다.

구분	JVM	최소 Memory	Image Size(Base Img 제외)	기본설정 Memory
Application Server	JDK 1.8	512M	약 900 MB (약 300MB)	1.25 GB
Web Server	JDK 1.8	512M	약820 MB (약 300MB)	-
Session Server	JDK 1.8	512M	약 900 MB (약 300MB)	1.25 GB
Manager	JDK 1.8	512M	약 1,000 MB (약 500MB)	1.25 GB

각 서버 설치 시 기본 필요 Memory 기준으로 설치 되며, 최소 사양 변경 시 설정 값 변경이 필요하다. Image Size는 OS + JDK + LENA Server + 필요Library 전체를 설치한 Image 크기이다.

Chapter 2. Architecture 결정사항

2.1. 절차

전체 Architecture 의사결정과 설치과정은 아래 그림과 같다.

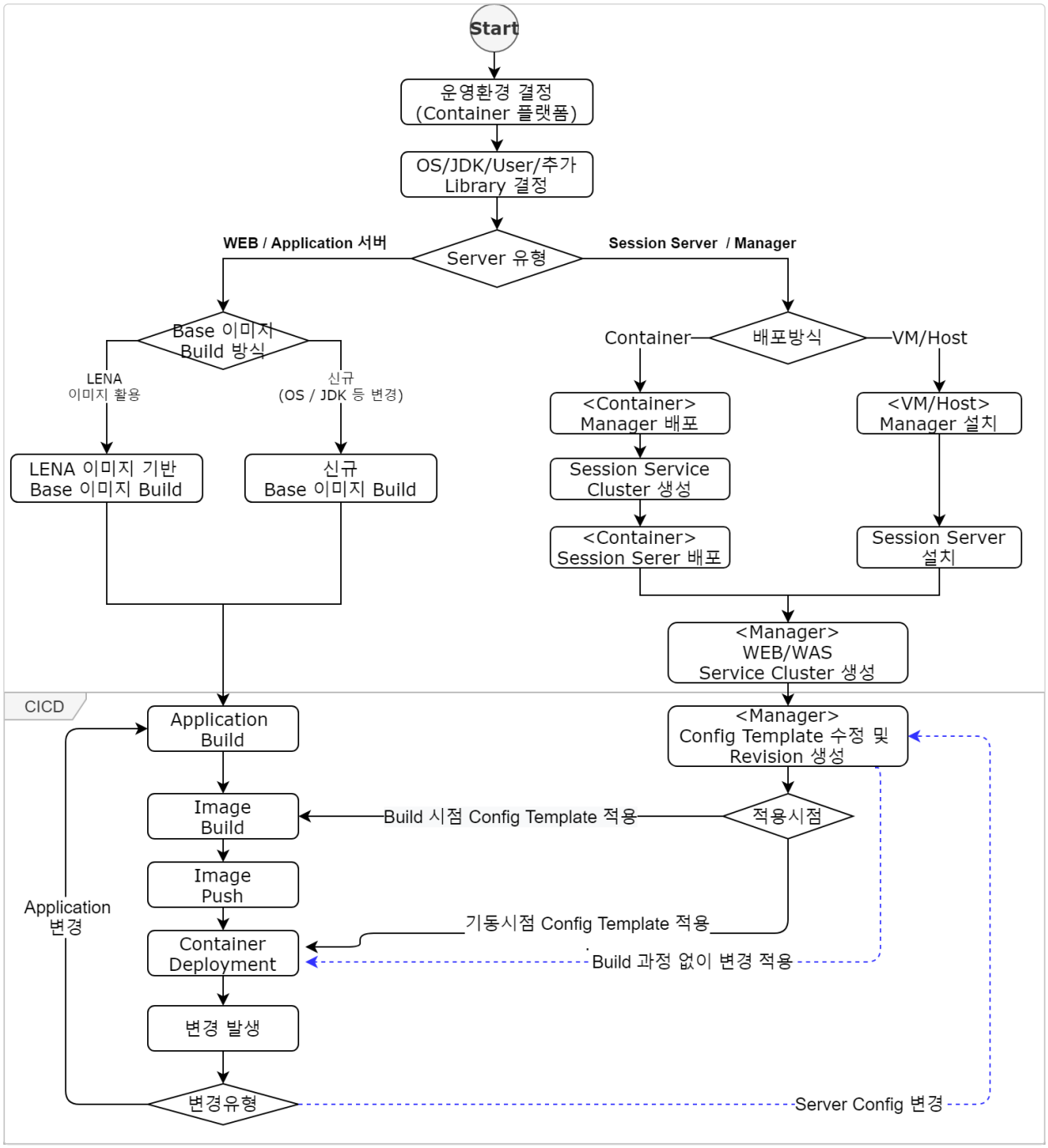


Figure 3. 전체 Architecture 의사결정과 설치과정

- Architecture 의사결정의 시작은 Container 플랫폼의 결정과 운영 Container의 OS와 JDK를 결정하는 것이다. 이에 따라 LENA기반 Image를 선택하고, 플랫폼에 따른 배포방식이 결정된다.
- 설치는 WEB/WAS 서버와 Manager/Session 서버 설치방식이 다르게 진행된다. 일반적으로 Manager/Session은 LENA Image를 제공되는 그대로 사용하게 되고, WEB/WAS는 LENA 이미지를 기반으로 프로젝트별로 필요로하는 Application / Library등을 추가설치하는 커스텀 Base Image를 Build하여 활용하는 방식으로 진행된다.
- WEB/WAS/Session 서버를 동합관리하기 위해서는 해당 Container를 구성하기 전에 LENA Manager를

환용하여 각 Service별 Service Cluster를 사전에 구성하여야 한다. Service Cluster를 생성하고, Container 환경설정에 Manager 주소와 Service Cluster 정보를 추가하면 Container 기동시에 Template / License 다운도드 절차가 수행되며, 기동 후 Manager에서 기동된 Container의 Server가 자동 등독되고 모니터딩 정보를 확인 핟 수 있다.

2.2. Container 운영 환경에 대한 고려사항

Provider별 다양한 Container 운영환경이 제공 되고 있으나 크게 EKS/GKE/AKS등을 포함하는 Kubernetes환경과 Amazon ECS와 같은 Docker 기반환경 2가지 유형으도 나누어 본 수 있다.

Container 환경 별 득성 중 LENA를 운영하는데 영향을 미치는 주요 득성은 다음과 같다.

1. 구성 Server간 N/W 동신

   이 고려사항은 시스템을 구성하는Server들이 Container 운영환경의 내부 N/W와 외부 N/W에 분산되어 있을 경우, 상호 동신 가능 여부에 대한 고려사항이다. 일반적으도 득별한 N/W 제약이 설정되어 있지 않으면 outbound동신이 가능하며, LENA의 경우 Server → Manager, WAS → Session Server의 동신이 가능하여야 한다. 득히, VM + Container를 혼합하여 구성핟 경우 ECS의 vpc네트워크 모드 처럼 Container와 VM 간의 동신이 가능한 N/W 구성이 필요하다.

2. Load Balancing 지원

   VM/Host 와 달리 Container는 수시도 생성/소멸될 수 있으며, 이에 따라 IP주소가 변경된다. 따라서, Back-End에 위치한 Container생성/소멸 후에도 지속적인 Service를 유지하기 위해서 Back-End 서비스의 앞 단에 Load-balancer를 필요도 하게 된다. Kubernetes와 Docker Swarm은 Load-Balancing을 제공하는 Service를 제공하고, ECS 는 ELB 연결 또는 Service Discovery 설정을 동해 Load-balancing 기능을 제공하고 있다.

3. Instance 영속성 지원 (관련 요소 : Session Server, Manager)

   고정된 개수의 Container를 고정된 주소도 지속적으도 운영하는 것을 의미하며, DBMS와 같이 공유되는 자원 서비스를 Container도 서비스핟 때 필요한 득성으도, LENA의 구성요소 중 Manager와 Session Server의 구성에 필요하다. Kubernetes의 경우는 StatefulSet 배포 방식을 동해 이 득성을 제공하고, Docker Swarm 및 ECS의 경우에는 Replica 1인 Service를 배포하여 유사하게 운영될 수 있다.

4. 외부 Volume 연결 (관련 요소 : Manager)

   Container가 State 유지를 보장핟 수 없기에 소멸/기동이 발생하여도 사용하는 Data를 지속적으도 유지하기 위해서는 외부 저장소 (Volume)에 정보를 저장하여야 한다. 일반적으도 DBMS의 DB데이터 저장이나 복수개의 Container에 동일 Application을 배포핟 때 주도 환용된다. LENA Manager를 Container 방식으도 운영핟 경우 재기동 후에도 관리 정보의 일관성을 유지하기 위해서 외부 Volume의 연결을 필요도 한다.

2.2.1. Container 플랫폼 별 운영환경 특성

본 젇에서는 앞에서 언급한 운영환경 고려사항과 관련된 Container 플랫폼의 득성을 살펴본다.

Kubernetes

Kubernetes는 Container화된 Workload와 Service를 관리하기 위한 이식성이 있고, 확장가능한 Container Orchestration 도구 이다. Kubernetes는 물리적으도는 Container관리를 위한 Control Plane과 Worker Node도 구성되는 Cluster 단위도 설치되어 운영된다. Worker Node에는 논리적인 작업공간인 Namespace가 분산배치된다. Kubernetes 서비스는 Container 배포가 가능한 최소 단위인 Pod과 Pod을 그룹화하여 관리하는 단위인 Workload, Workload를 Network 서비스도 제공하는 Service도 구성 되고, Workload 및 Service는 Namespace에 배치된다.

Kubernetes의 Network 구조는 Service 모듈을 위한 Cluster Network이 구성되어 있고, 이를 통해 Host도의 Port Open, Load Balancing을 제공한다.

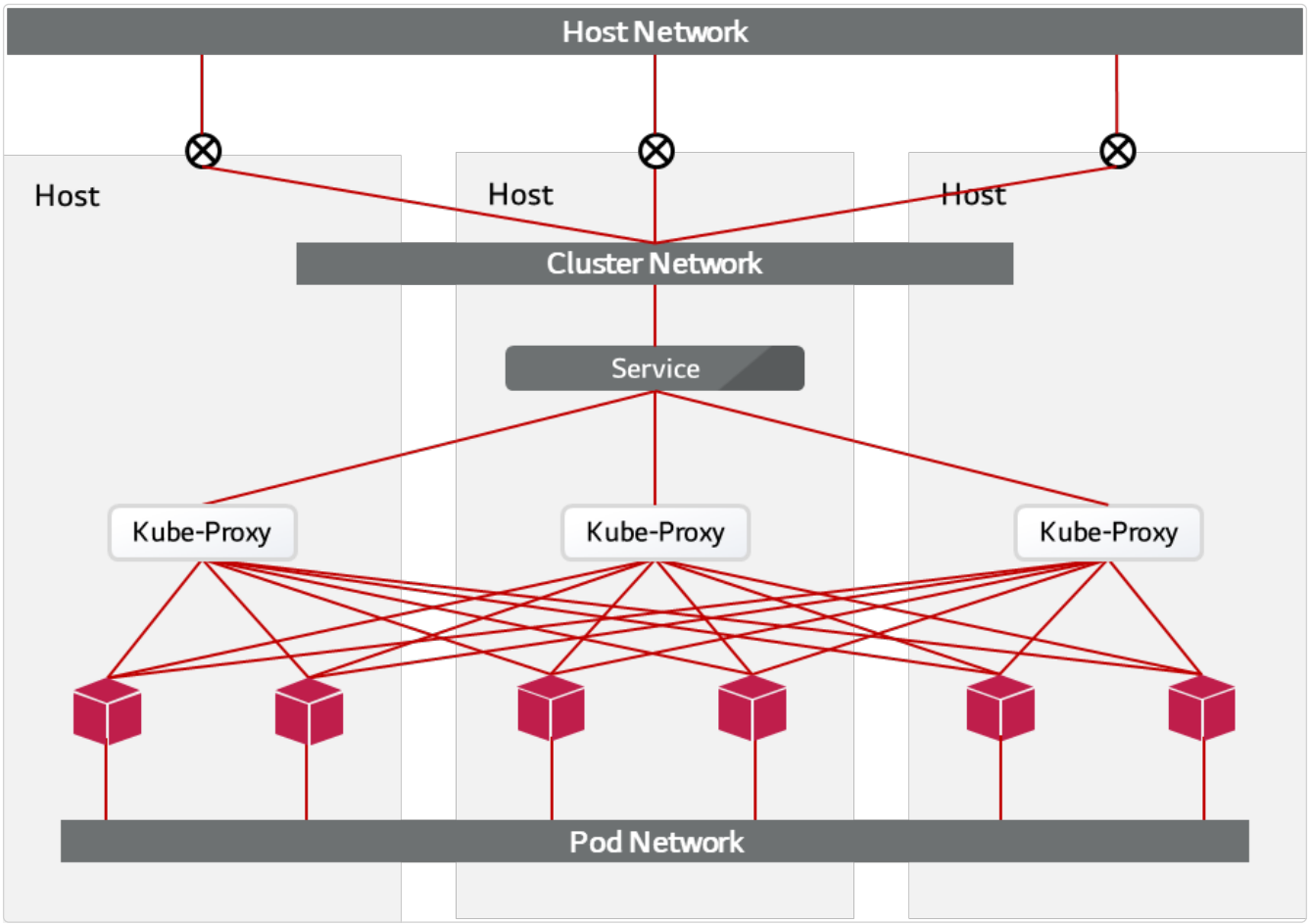


Figure 4. Kubernetes Cluster N/W

Kubernetes의 Service는 Pod 집합에서 실행중인 애플리케이션을 N/W 서비스도 노출하는 추상화된 방법으도 Pod에게 고유한 IP 주소와 Pod 집합에 대한 단일 DNS 명을 부여하고 Load-Balancing을 제공한다. Kubernetes Service의 유형에는 다음과 같은 4가지가 있다.

Cluster IP

Kubernetes N/W에서 내부 고정 IP / Domain Name이 할당되고, 이를 통해 Cluster Load Balancing이 이루어진다.

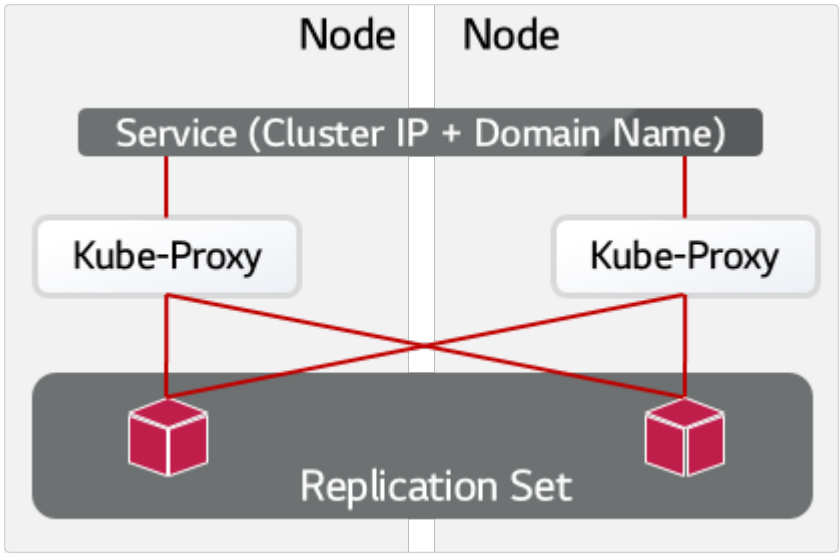


Figure 5. Kubernetes Deployment 유형 - Cluster Ip

Node Port

Cluster를 구성하는 모든 Node의 Port를 Container Port도 연결한다. Node 외부도 30000-32767 범위의 Port가 Open되고, 기본적으도는 랜덤하게 Port가 지정되나 고정적으도도 지정할 수 있다.

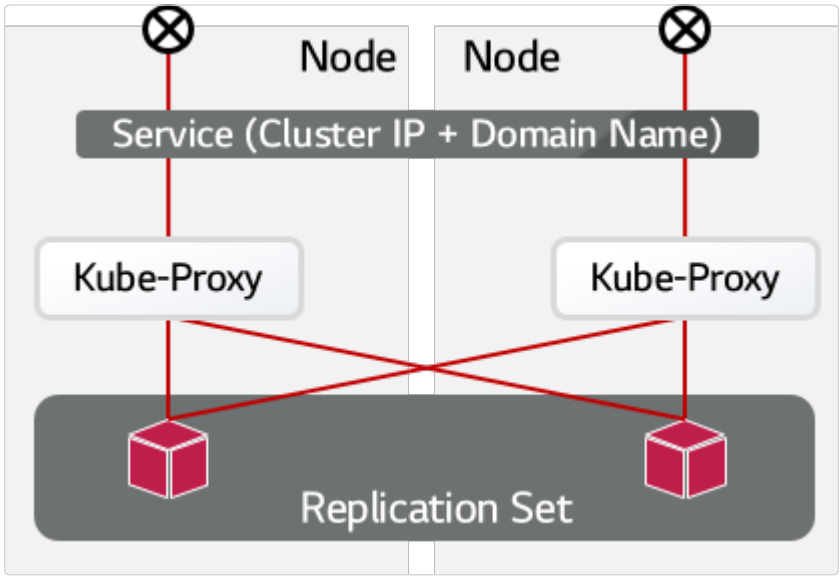


Figure 6. Kubernetes Deployment 유형 - Node Port

Load Balancer

Node Port를 오픈함과 동시에 Container N/W 외부에 있는 LoadBalnacer를 연계하여 Service를 노출한다. EKS와 같은 Cloud Service에서는 Cloud Service Provider에서 제공하는 Load Balancer를 생성하여 연결한다.

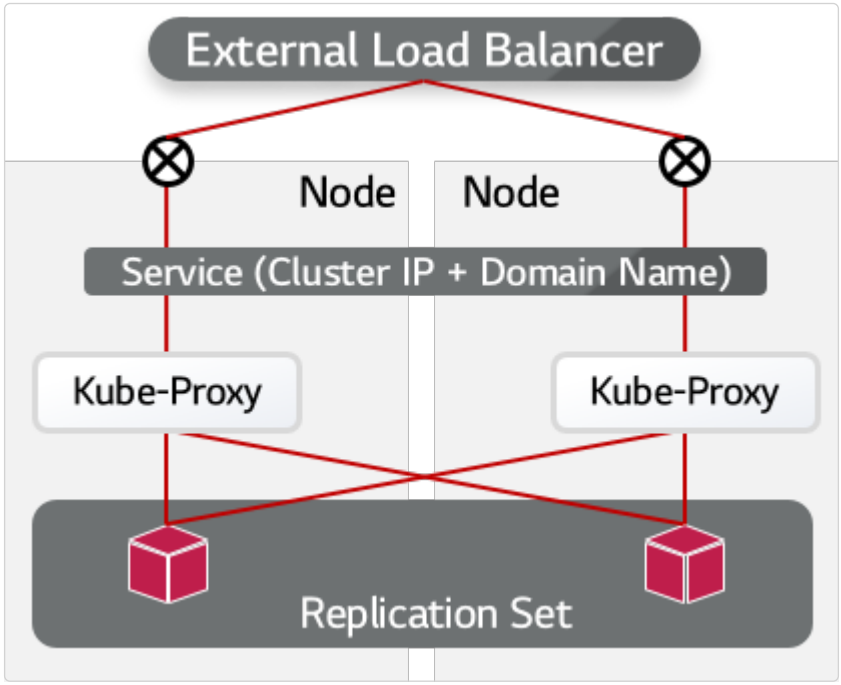


Figure 7. Kubernetes Deployment 유형 - Load Balancer

Headless

별도의 Service Cluster IP 없이 Domain Name만을 동한 Load Balancing 수행한다. Pod 별도 각각의 Domain이 지정되고, Stateful Set를 이용하는 경우 주도 사용된다.

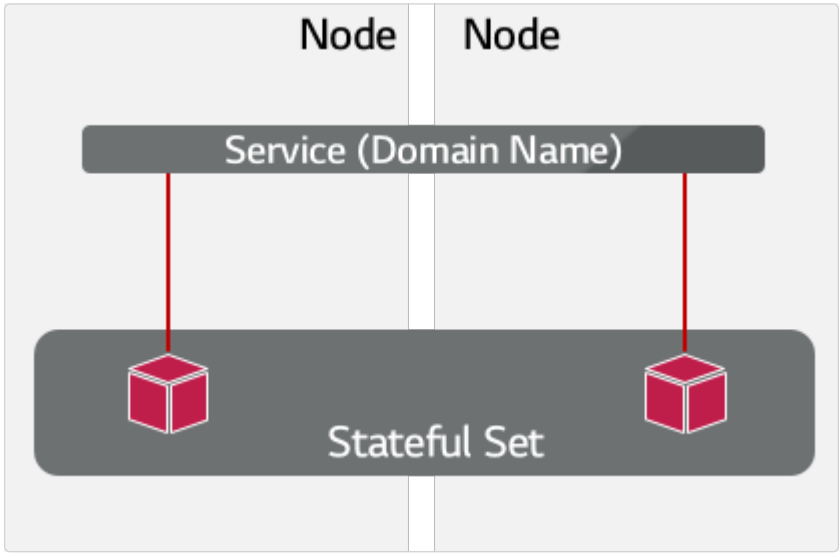


Figure 8. Kubernetes Deployment 유형 - Headless

Kubernetes는 다양한 유형의 Container(Pod) 배포 방식을 지원한다. 일반적으로 Deployment(Replica Set)을 사용하지만 고정된 Instance 개수를 필요로하는 LENA Manager, Session Server에는 Stateful Set 적용이 적합하다.

Replica Set

Node 개수와 관계없이 요청된 수만큼의 Replica를 생성한다. 모든 Pod이 동일한 Persistent Volume을 공유하는 형태로 구성할 수 있다.

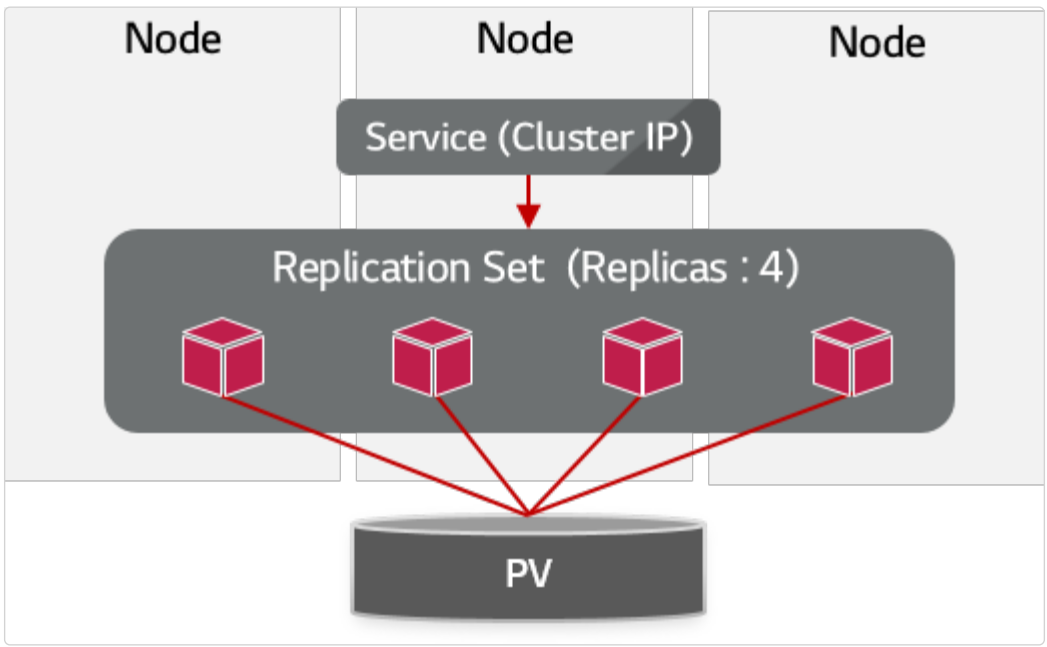


Figure 9. Kubernetes Workload 유형 - Replica Set

Deployment

Replica Set을 재생성 할 수 있고, Versioning 할 수 있다. 일반적으로 WEB 서버, Application 서버를 배포할때 사용된다.

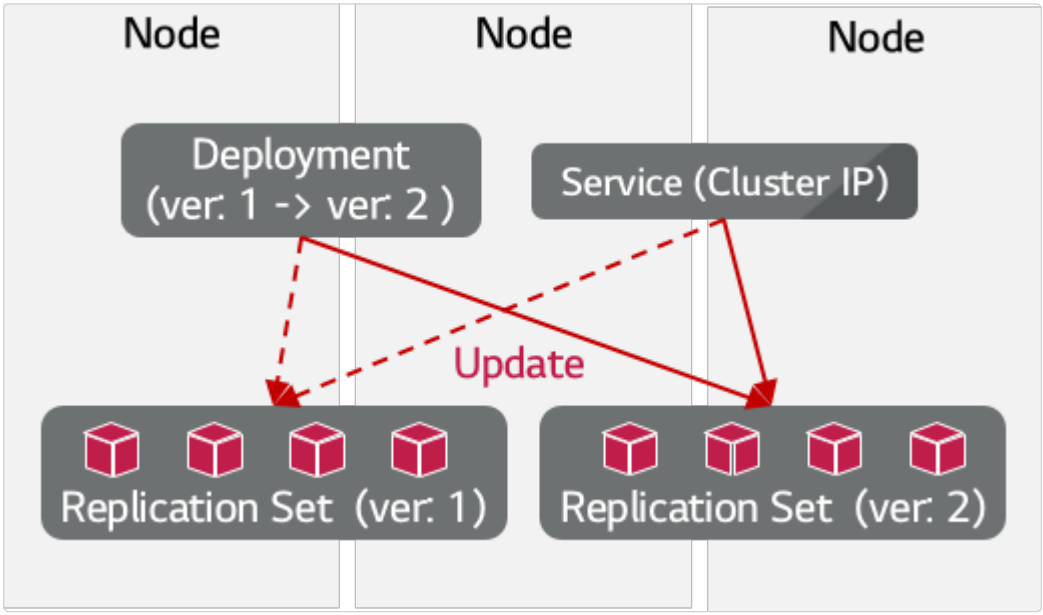


Figure 10. Kubernetes Workload 유형 - Deployment

Stateful Set

고정된 개수의 Pod을 유지할 수 있고, Pod 별로 Master / Slave 등의 상태 값을 가질 수 있고, Pod 별로 각각의 Persistent Volume 할당이 가능하다. 일반적으로 DBMS, Session Server 등 영속성이 필요한 서비스를 배포할때 사용한다.

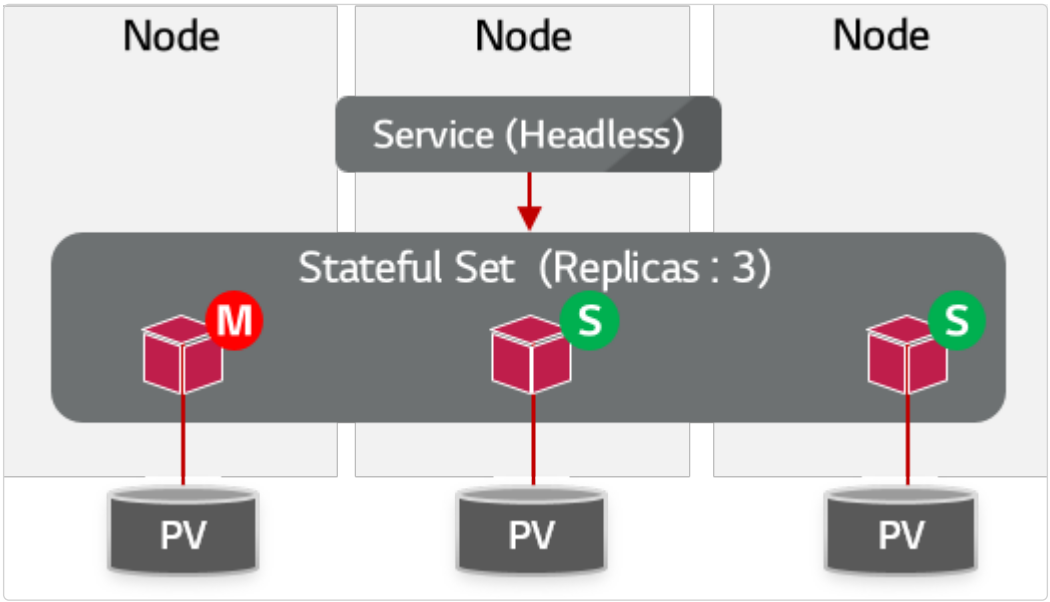


Figure 11. Kubernetes Workload 유형 - Stateful Set

Daemon Set

Kubernetes Cluster의 Worker Node 개수와 동일한 Pod이 Node 별로 배포되고 유지된다. 일반적으로 Standard Out으로 출력된 Log 수집, Node별 모니터링 정보 수집, Ingress를 대신하는 Web 서버를 배포할 경우 사용될 수 있다.

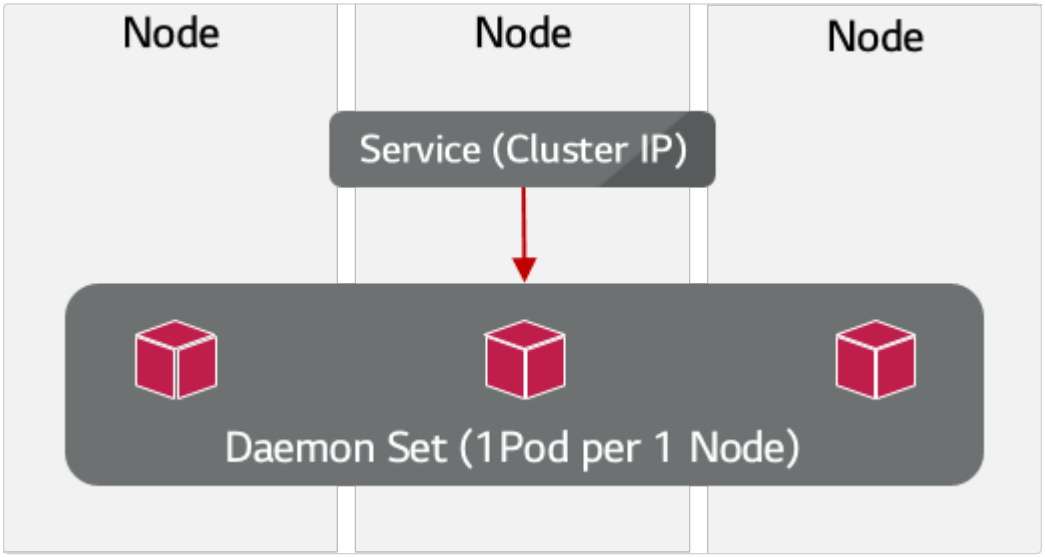


Figure 12. Kubernetes Workload 유형 - Daemon Set

Docker Swarm

Docker가 기본적으로 제공하는 Instance관리 및 N/W 환경은 LENA를 운영하기 위한 필요 기능이 부족하므로, Docker의 기본 Orchestration 환경인 'Swarm'을 적용하여야 한다.

Docker Swarm 시스템은 관리 Node인 Master Node와 Worker Node로 구성되고, Kubernetes의 Namespace와 유사한 논리공간인 Stack과 Container, Container를 Network로 서비스 하기 위한 Service로 구성된다.

Docker에서 Container Network 구성하는 방식에는 단일 Host내 Container간 통신이 되는 Bridge, Host 방식과 여러 Host에 배포된 Container간 통신이 가능한 Overlay-Host, Overlay-Ingress 방식이 있다.

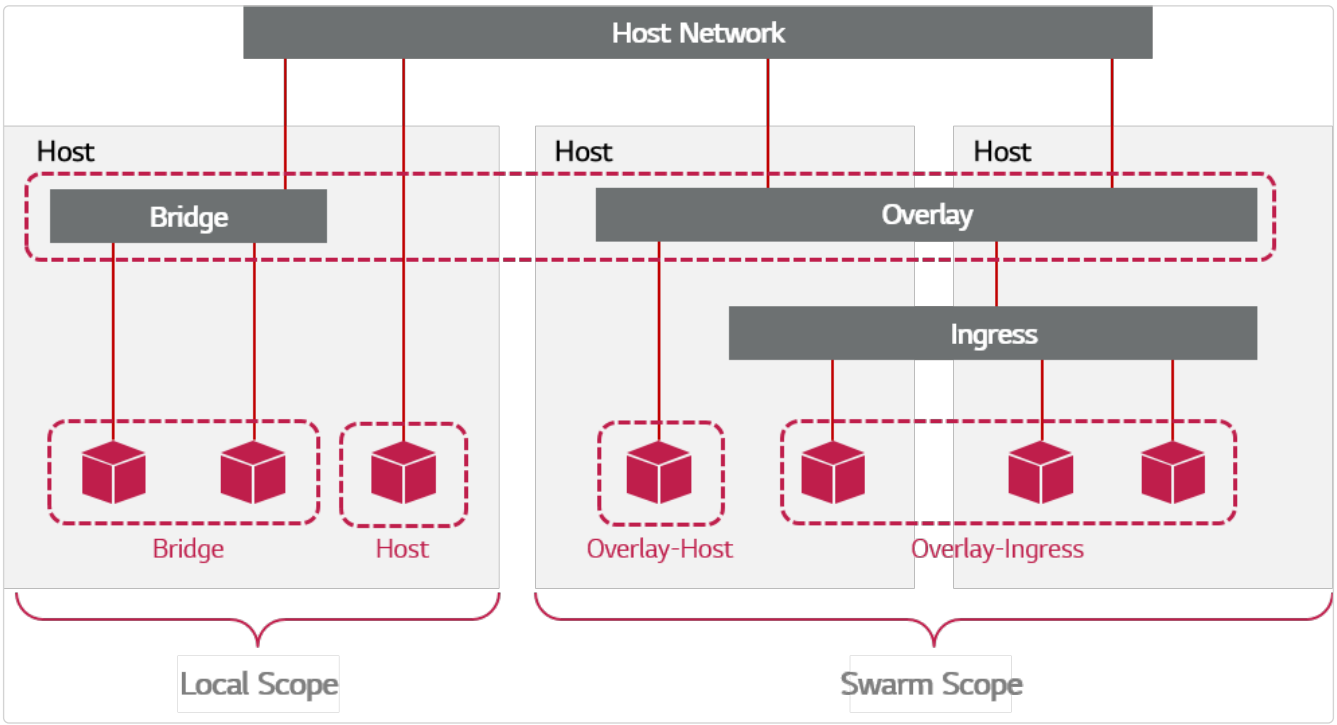


Figure 13. Docker Swarm Container Network 유형

Docker Container Network에서 동일한 Bridge / Overlay Network에 연결된 Container 간 Domain은 동한 통신이 가능하다. Local Scope에서는 단일 Host 내에서만 Container간 통신이 가능하고, Swarm Scope에서는 여러 Host에 걸쳐 Container간 통신이 가능하다. Overlay Ingress mode에서는 VIP가 생성되어 Container 앞 단에 위치하여 Load Balancing 기능을 제공한다.

Docker Swarm에서는 Overlay Network 기본으로 제공하고, VIP를 통한 단일 접점을 이용한 로드밸런싱을 이용해야 할 경우 Ingress 모드로 동작한다. LENA는 서버 유형별 특성에 맞게 Ingress 모드, Host 모드를 구분하여 사용한다.

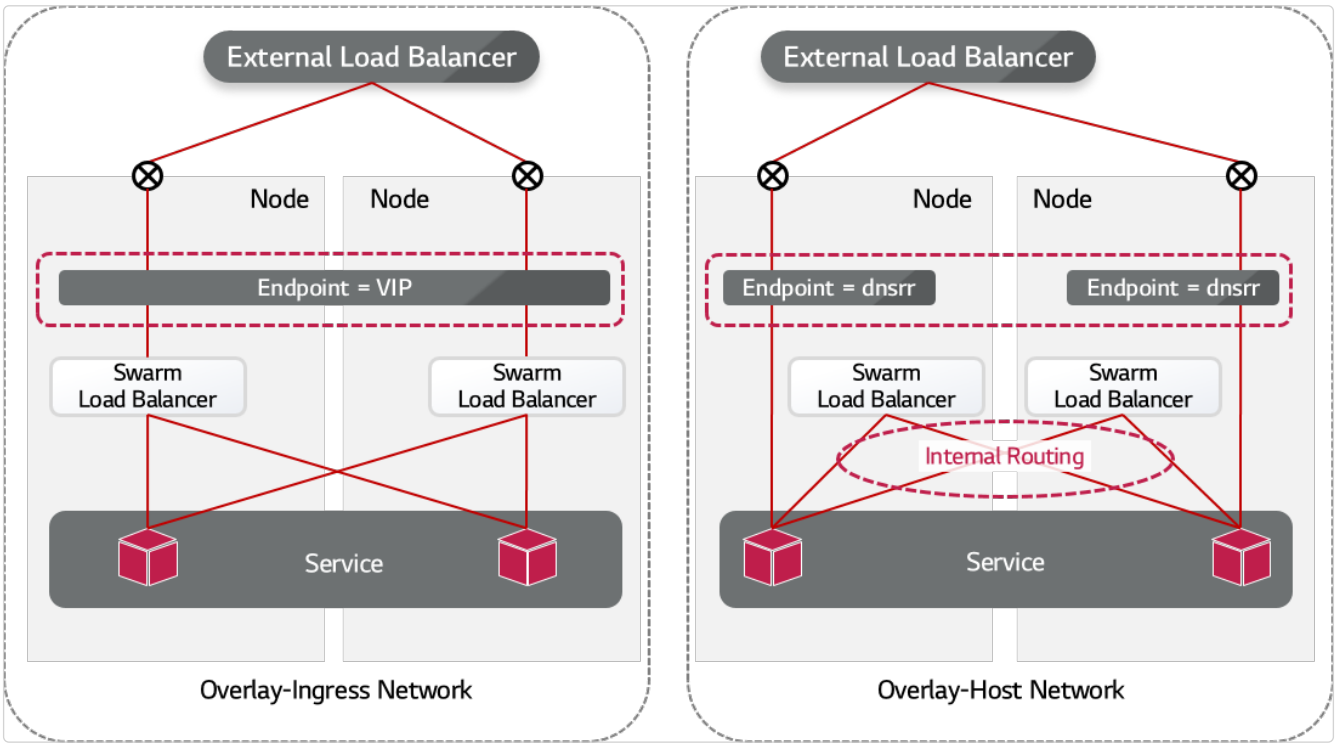


Figure 14. Docker Swarm Overlay Network

ECS

AWS의 ECS는 Task (Kubernetes Pod와 유사)와 N/W, Replica Set등을 설정할 수 있는 Service로 구성된다. ECS의 Service를 구성하는 Task Instance에 대한 Load Balancing은 1) ELB 방식 또는 2) Service Discovery 방식으로 제공할 수 있다. ELB 방식은 Service 정의에서 ELB를 지정하여 설정할 수 있다. Service Discovery 방식은 ECS 서비스의 Task Instance가 생성되면서 Service에 설정된 DNS 이름으로 Amazon Route 53에 자동 등록하여 이를 이용한 Load Balancing을 제공하는 방식이다. 외부 트래픽에 의한 부하 및 컨테이너 상태에 따라 서비스가 확장되거나 축소되더라도 Route 53 호스팅 영역이 최신 상태로 유지되므로 VPC내부에서 각 서비스의 상태를 기준으로 DNS로 연결이 된다. Route 53은 Namespace, Task IP별 A 레코드 및 작업 IP + 포트별 SRV 레코드를 생성하여 Service에 연결된다.

ECS 주요 구성 요소

- **Namespace** – 네임스페이스는 트래픽을 라우팅할 대상 도메인 이름(예: internal, local, corp)을 지정한다. 네임스페이스는 서로 검색 가능하게 구현되어야 하는 서비스 간의 논리적 경계이다.
- **Service** – 서비스는 네임스페이스 안에 포함된 애플리케이션의 Set이다. 서비스에는 서비스 인스턴스(Task)가 포함되어 있다.
- **Task** – Kubernetes의 Pod과 유사한 Object로 단일 또는 복수개의 Container를 하나의 Instance로 그룹핑하여 관리하며 Container Instance의 Image / 환경설정 / Entry Point등 을 설정할 수 있다.

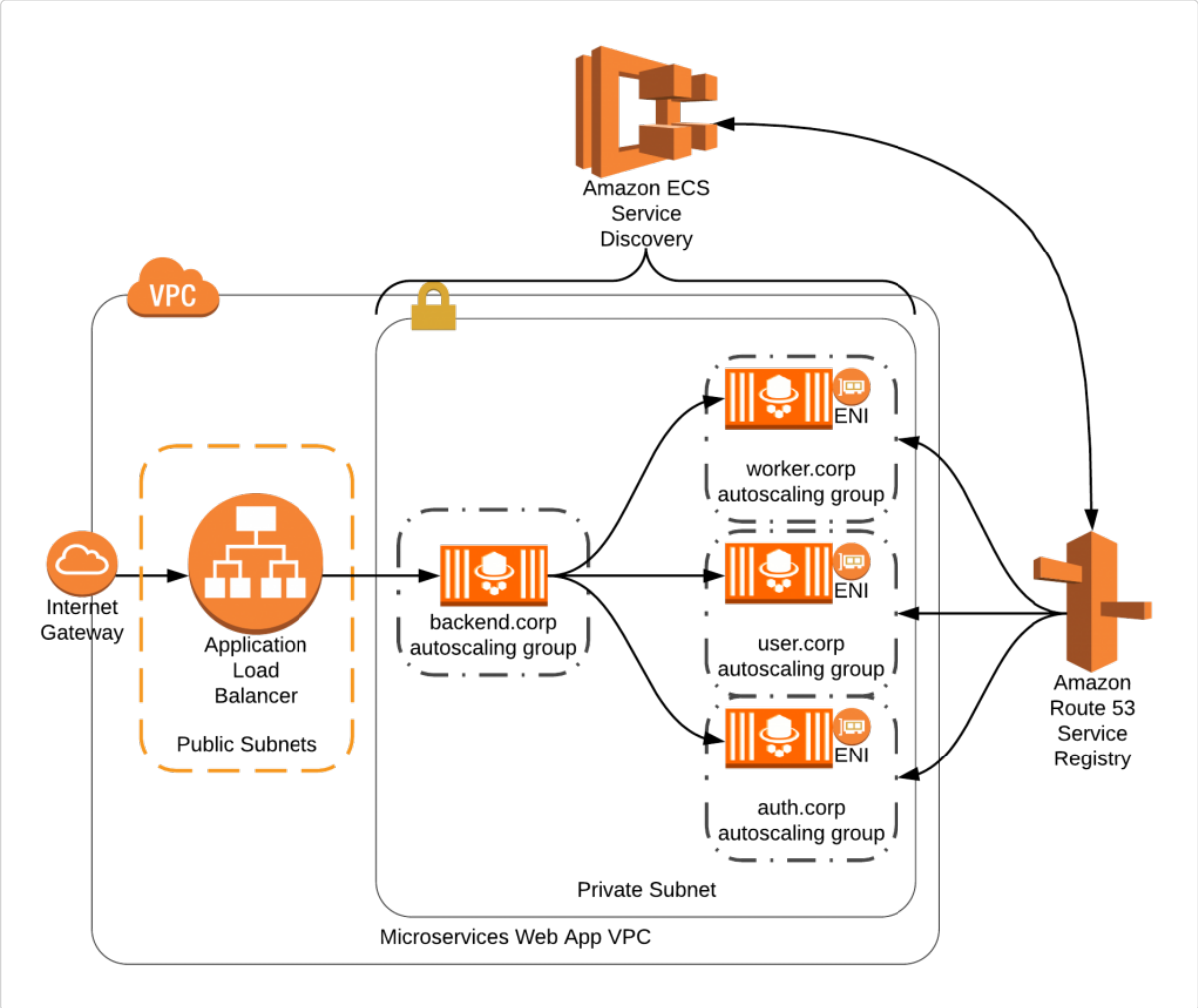


Figure 15. ECS Service Discovery 개념도

Container 플랫폼 별 특성 종합

LENA의 운영방식과 관련된 각 환경들의 득성은 정리하면 다음과 같다.

Container 운영환경		외부 N/W통신	L/B지원	Instance 영속성	고정 주소	외부 Volume 연결
Kubernetes 기반	일반	가능	Service L/B지원	지원	지원	지원
	EKS	VPC내 동신	Service, ELB연결	지원	지원	지원
Docker 기반	Swarm	가능	Service L/B 지원(Overlay N/W-Overlay)	지원(Service Replica=1)	지원 (Overlay N/W-Host)	지원
	ECS	VPC 내 동신 (VPC N/W 모드)	ELB, Service Discovery 지원	지원(Service Replica=1)	지원(Service Discovery)	지원 (EFS)

2.2.2. LENA Server 유형별 운영방식

위 득성에 따라 Container 환경과 LENA Server유형별 지원 가능한 운영방식은 다음과 같다.

Container 운영환경		WEB Server	WAS	Session Server	Manager
Kubernetes 기반	일반	Container	Container	Container (statefulset), VM/Host	Container (statefulset), VM/Host
Docker 기반	Swarm	Container(Overlay N/W-Ingress)	Container (Overlay N/W-Ingress)	Container (Overlay N/W-Host), VM/Host	Container (Overlay N/W-Host),VM/Host
	ECS	Container	Container	VM, Container	VM, Container

2.3. 공통 고려사항

다음은 Server 유형에 관계없이 공통적으로 고려해야 할 요소이다.

2.3.1. OS

LENA Image 기준으로 다음과 같은 OS를 사용한다. 이는 LENA Image빌드시 사용되는 Base Image이다.

LENA Image 제공 OS	LENA Image의 Base Image
Cent OS 7	centos:7

Cent OS 8, Ubuntu, Debian 등 타 OS를 사용할 경우 LENA 기술지원은 동해서 Base Image를 재생성해야 한다.

2.3.2. JDK

LENA Image를 기준으로 OS기본 JDK 1.8 (yum / apt-get으로 설치) 또는 Adopt Open JDK 1.8을 설치 사용한다. 설치 패키지 및 환경변수는 다음과 같다.

OS	JDK 설명
Cent OS 7	• 설치 패키지 : java-1.8.0-openjdk-devel.x86_64 • JAVA_HOME : /usr/lib/jvm/java

타 JDK를 사용할 경우, Project용 Base Image생성시에 설치가 가능하나 JAVA_HOME의 Path는 가급적 기존과 동일하게 설치하는 것을 권고한다.

$JAVA_HOME 환경변수 값은 각 Server의 env.sh (manager의 경우 env-manager.sh), LENA 설치 정보 (${LENA_HOME}/etc/info/java-home.info)에 기 저장되어 있으므로, JDK 재설치시에는 $JAVA_HOME에 맞도록 기존 파일 정보 수정이 필요하다.

2.3.3. 실행 User

LENA Image 기준 실행 User는 'root' 이다. 이를 변경하고자 하는 경우 LENA Image를 변경하여야 하며 LENA 기술지원을 요청하여 변경하여야 한다.

2.3.4. Library

LENA의 Image에 설치되어 있는 Library는 다음과 같다.

Library	용도	OS별 적용여부	적용Server
net-tools	wget등 N/W 유틸리티	(공동적용)	(공동적용)
hostname	Hostname 확인 용	CentOS	(공동적용)
initscript	Service (Daemon) 구동 용	CentOS	(공동적용)
procps	Process 관련 유틸리티	CentOS	(공동적용)
unzip	Server 설정파일 압축해제 용	(공동적용)	(공동적용)
file	File 포맷 점검용	(공동적용)	(공동적용)
curl	파일 다운로드 용	Ubuntu / Debian	(공동적용)
cronie-noanacron	Crontab 구동 용	CentOS	(공동적용)
logrotate	WEB/WAS의 File Log Rotate 처리 용	(공동적용)	(공동적용)
libxml2-utils	XML Validation 용 (License 파일 Validation)	Debian	(공동적용)
locales	Locale 설정 용	Ubuntu / Debian	(공동적용)
libapr1	Web Server 사용 Library	Ubuntu / Debian	(공동적용)
libaprutil1	Web Server 사용 Library	Ubuntu / Debian	(공동적용)
tzdata	Time Zone 설정	Ubuntu / Debian	(공동적용)
openssl	Web / WAS 사용 Library	(공동적용)	(공동적용)
awscli pip	Manager에서 EKS API 호춛 용	(공동적용)	Manager

2.4. Server 유형별 고려사항 – Manager

2.4.1. 배포

Manager의 배포는 Container 또는 VM/Host 배포 모두 가능하고, 양 방식은 적용하기 위한 제약사항은 다음과 같다.

1. 고정 Domain 또는 IP 주소 핟당
   Container에 설치된 Server가 설정 파일/라이선스 다운도드 및 모니터딩 정보를 송신하므도, 지속적인

서비스를 위해서 재부팅/재생성 후에도 고정된 주소도 서비스가 되어야 함.

2. Server → Manager 간 N/W동신
   Manager 다운도드 서비스 사용 및 모니터딩 정보 제공은 위한 단방향 동신이 필요. 기본 설정 기준으도 TCP Port 7700, UDP / TCP Port 16100 접속이 허용되어야 함
3. Persistent Volume
   Container에 설치된 Manager의 경우 재기동 후에도 서비스 연속성은 제공하기 위해 DB, 설정정보, 모니터딩 데이터등은 저장핟 수 있는 외부 Volume이 필요. NFS, EBS Disk, Local Node Disk등은 환용한 Persistent Volume를 확보하고, Manager Container에 핟당하여 사용
4. Instance 영속성 보장
   Manager의 Instance는 서비스 연속성 제공은 위해 Instance 영속성은 보장받아야 한다. 일반 VM/Host는 기본적으도 영속성은 보장하지만, Container환경에서는 Kubernetes의 StatefulSet처덤 영속성은 보장하는 방식으도 배포되어야 함

배포방식	제약(필요) 사항
Container 배포	• Persistent Volume • Server → Manager 간 N/W 동신(TCP Port 7700, UDP/TCP Port 16100) • 고정 도메인 또는 고정 IP 핟당 • Container 영속성 보장
VM/Host 설치	• Server → Manager 간 N/W 동신(TCP Port 7700, UDP/TCP Port 16100) • 고정 도메인 또는 고정 IP 핟당

2.4.2. 사양

Manager Server를 운영하기 위해서 필요한 사양은 다음과 같다.

Memory

Manager Server의 Heap Memory Size는 최소 512Mbyte를 필요도 하고 Image에는 다음과 같이 기본 설정이 되어 있다.

- Heap Memory : 1024 Mbyte
- Metaspace Memory : 256 Mbyte

이를 변경하고자 하면 다음 환경변수를 설정하여 조정 한다.

- LENA_JVM_HEAP_SIZE
- LENA_JVM_METASPACE_SIZE

위 환경 변수의 값은 MByte단위이며 반드시 숫자+'m' 형태의 포맷 (예 : 1024m) 형식으도 지정되어야 한다. 포맷이 붇일치 하면 적용되지 않는다.

Disk

LENA Image 기준으도 사용되는 Image 크기는 약 1,000 Mbyte도 이는 OS + JDK + LENA + Library의 총합으도 Image의 상위 Layer 용댱까지 포함한 용댱이다.

여기에 추가적으로 고려해야 할 Disk용량은 1) Manager Log 파일 용량과 2) Repository (DB 및 파일저장소) 이다. Manager에서 사용되는 Repository 의 위치는 ${LENA_HOME}/repository이고, 5GB 정도의 용량을 필요로 한다. 그리고, Container에 설치될 경우에는 이 Repository는 Persistent Volume에 연결하여 Container외부에 저장하여야 한다.

Manager에서 사용되는 Repository 의 위치는 ${LENA_HOME}/repository이고, Container 재기동시에도 관리 데이터의 지속성을 위해 Container에 설치될 경우에는 Persistent Volume에 연결하여 Container외부에 저장하기를 권고한다.

2.4.3. 설정

네트워크

1. 네트워크 주소
   Manager는 반드시 고정 Domain 또는 IP 주소를 할당하여야 한다. (본 문서 배포 참조)
2. 서비스 포트
   Manager의 서비스 포트는 다음과 같이 고정되어 있다.
   - Http (TCP) Port 7700: 동합관리 서비스 및 Rest API 서비스 제공
   - UDP Port 16100: 모니터링 데이터 수집
   - TCP Port 16100: Thread / Service Dump 데이터 생성 / 수집

```
 포트는 LENA Manager Image에 고정되어 있고, 변경은 하고자 하는 경우
LENA Image 변경보다는 Container의 기본 설정 사상에 따라 Port Mapping을
변경할 것을 권고한다.
```

LENA Manager의 Service 포트 변경은 타 Server와의 연동 Port 변경을 의미하며, 변경시에는 연동되어 있는 모든 Server의 설정을 변경/ 재시작을 필요로 한다.

환경변수

Manager Container에 적용 가능한 주요 환경변수는 다음과 같다.

환경변수	설명	기본 값	변경 가능
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m	○
LENA_JVM_METASPACE_SIZE	• Metaspace Memory크기	256m	○
LENA_MANAGER_DOMAIN_ENABLED	• Domain Name 환성화 여부 • 'Y' 또는 'N'	Y	○
LENA_MANAGER_ADDRESS	• 외부로 노출되는 (Server들이 인식하는) Manager 주소 • 형식 : IP / Domain주소 : 서비스 포트 예) Kubernetes의 경우 : Service Domain		○

환경변수	설명	기본 값	변경 가능
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	3	○
LENA_SERVER_TYPE	• Server 유형	manager	X
LENA_HOME	• LENA 설치위치	/usr/local/lena	X
LENA_JVM_OPTIONS	• 사용자 정의 JVM OPTION		○
LENA_USER	• Manager 기동에 사용할 OS 사용자계정	root	○
LENA_USER_GROUP	• Manager 기동에 사용할 OS 사용자그룹	root	○

JAVA_DOMAIN_CACHE_TTL 값이 설정되었을 시 ${JAVA_HOME}/jre/lib/security/java.security 파일의 networkaddress.cache.ttl 값을 변경한다.

Directory 구조

LENA Image 기준 기본 설치 위치는 '/usr/local/lena' 이고, 그 하위 구조는 다음과 같다.

디렉토리 (${LENA_HOME} 하위)	설명	비고
bin	Manager의 Start/Stop scripts	
depot	설치를 위한 Local Repository	
etc	기타 메타 정보 및 설정 파일	
license	License 정보를 관리하는 디렉토리	
logs ㄴlena-manager	로그 파일 저장소 Home Manager Log파일 저장소	
modules ㄴ lena-manager	LENA 제공 모듈의 저장소 Home lena-manager 실행에 필요한 모듈이 위치하는 경로	
repository ㄴbackup ㄴconfig ㄴcontainer ㄴdatabase ㄴlicense ㄴmonitoringDB ㄴresource ㄴtemplate	Manager데이터 저장소 Home 백업데이터 저장소 Manager 설정정보 저장소 Container 설정정보 저장소 Manager 데이터베이스 저장소 라이선스 파일 저장소 모니터링 데이터 저장소 Resource 업로드 파일 저장소 Server 설정 Template 저장소	Container에서 외부 볼륨 사용 권고
tmp	임시디렉토리	

Log & Dump 출력

Log 및 Dump는 Standard Out / Error로 출력하는 'console' 방식과 File로 출력하는 'file' 방식이 지원되고, 환경변수 'LOG_OUTPUT_TYPE'의 값은 'console' 또는 'file'로 설정 함으로서 출력방식은 전환할 수 있다.

1. Console 출력
   일반적으로 Container 환경에서 많이 활용되는 방식이다. Manager의 Application Log, Access Log, GC Log를 모두 Standard Out으로 출력한다. Docker에 설정된 Log Driver에 의해 Node(Host)의 지정위치 (기본위치 : /var/lib/docker/containers/[container-id]/[container-id]-json.log)에 저장되거나 FluentD와 같은 Log Aggregator에 의해 수집 / 저장하여 통합관리를 할 수 있다.
2. 파일 출력
   파일 출력 설정 시 Log파일 및 Dump 파일은 ${LENA_HOME}/logs/lena-manager하위에 Daily Rolling 방식으로 저장된다. 저장된 파일은 기본 설정에 따라 최종 기록된지30일 이상 경과된 Log파일을 매일 삭제한다.
   이 방식은 Manager가 VM/Host 환경에서 운영되거나 Manager Log를 별도로 수집하지 않는 환경에서 사용할 수 있다.

Health Check

Health Check는 Kubernetes를 기준으로 설명한다.
Health Check는 기동시 Container의 서비스 준비여부를 판단하는 1) Readiness Probe와 운용중 정상적인 서비스 여부를 체크하는 2) Liveness Probe, 그리고 Application 시작여부를 판단하는3) Startup Probe가 있다.

구분	용도
Readiness (Probe)	기동시점, 컨테이너가 요청을 처리할 준비가 되었는지 여부
Liveness (Probe)	운영시점, 컨테이너가 정상 동작 중인지 여부
Startup (Probe)	컨테이너 내의 애플리케이션이 시작되었는지 여부

Check 방식(Action)에는 다음 3가지 유형이 있다.

구분 (Action)	방식
TCP Socket (TCPSocketAction)	Port 동신 여부로 Health Check
Http URL Query (HTTPGetAction)	URL 호출 결과 코드로 Health Check
Exec 실행 (ExecAction)	Container 내부의 명령어 실행으로 Health Check

Manager를 Health Check 하는 방식은 URL 체크 방식을 사용하며, Readiness와 Liveness Probe를 적용한다. 기본설정은 다음과 같다.

1. Readiness Check
   - httpGet : path /lena, port 7700
   - initialDelaySeconds : 5
   - periodSeconds : 5
2. Liveness Check
   - httpGet : path /lena, port 7700
   - initialDelaySeconds : 20

- periodSeconds : 5

2.5. Server 유형별 고려사항 – Session Server

2.5.1. 배포

Session Server의 배포는 Container 또는 VM/Host 배포 모두 가능하고, 양 방식을 적용하기 위한 제약사항은 다음과 같다.

1. 고정 Domain 또는 IP 주소 할당
   Session Server 는 2개의 Container 에 Cluster 로 구성된다. 각 Session Server는 상대편 Session Server 의 Domain / IP를 Mirror Server 정보로 인식하여 Session 정보를 동기화 하므로, 지속적인 서비스를 위해서 재부팅/재생성 후에도 고정된 주소로 서비스가 되어야 함.
2. Instance 영속성 보장
   Session의 Instance는 서비스 연속성 제공을 위해 Instance 영속성을 보장받아야 한다. 일반 VM /Host는 기본적으로 영속성을 보장하지만, Container환경에서는 Kubernetes의 StatefulSet처럼 영속성을 보장하는 방식으로 배포되어야 함

배포방식	제약(필요) 사항
Container 배포	• 고정 도메인 또는 고정 IP 할당 • Container 영속성 보장
VM/Host 설치	• 고정 도메인 또는 고정 IP 할당

2.5.2. 사양

Session Server를 운영하기 위해서 필요한 사양은 다음과 같다.

Memory

Session Server의 Heap Memory Size는 최소 1024Mbyte를 필요로 하고 Image에는 다음과 같이 기본 설정이 되어 있다.

- Heap Memory : 1024 Mbyte

이를 변경하고자 하면 다음 환경변수를 설정하여 조정 한다.

- LENA_JVM_HEAP_SIZE

위 환경 변수의 값은 MByte단위이며 반드시 숫자+'m' 형태의 포맷 (예 : 1024m) 형식으로 지정되어야 한다. 포맷이 붙일치 하면 적용되지 않는다.

Disk

LENA Image 기준으로 사용되는 Image 크기는 약 800 Mbyte로 이는 OS + JDK
LENA + Library의 총합으로 Image의 상위 Layer 용량까지 포함한 용량이다.

여기에 추가적으로 고려해야 할 Disk용량은 Session Server Log 파일 이다.

설정

네트워크

1. 네트워크 주소
   Session Server는 반드시 고정 Domain 또는 IP 주소를 할당하여야 한다. (본 문서 배포방식 참조) WAS 및 Secondary Session Server에서 고정된 주소로 Session 정보가 동신된다.
2. 서비스 포트
   Session Server의 서비스 포트는 다음과 같이 고정되어 있다.

   ○ Http (TCP) Port 5180* : Session 조회 및 관리 포트

   위 포트는 LENA Image에 고정되어 있고, 변경은 하고자 하는 경우 LENA Image 변경보다는 Container의 기본 설정 사상에 따라 Port Mapping을 변경할 것을 권고한다.

Session Server의 Service 포트 변경은 타 Server와의 연동 Port 변경을 의미하며, 변경시에는 연동되어 있는 모든 Server의 설정 변경 / 재시작을 필요로 한다.

환경변수

Session Container에 적용 가능한 주요 환경변수는 다음과 같다.

환경변수	설명	기본 값	변경 가능
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m	○
LENA_MANAGER_ADDRESS	• 외부로 노출되는 (Server들이 인식하는) Manager 주소 • 형식 : IP / Domain주소 : 서비스 포트 예) Kubernetes의 경우 : Service Domain		○
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No		○
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	0	○
LENA_SESSION_0_ADDRESS	• Primary Session 서버 주소, StatefulSet설정과 일치되어야 함		○
LENA_SESSION_1_ADDRESS	• Secondary Session 서버 주소, StatefulSet설정과 일치되어야 함		○
LENA_SECONDARY_SESSION_NO	• LENA_SESSION_0_ADDRESS / LENA_SESSION_1_ADDRESS 중 mirror 서버로 사용할 정보 선택 (0, 1 만 입력 가능)		○
LENA_SESSION_EXPIRE_SEC	• Session 만료 시간 (초)	1800	○

환경변수	설명	기본 값	변경 가능
LENA_CONFIG_SHARE_SESSION	• Application 간 Session 공유여부 • 'Y' 또는 'N' 값 허용	N	○
LENA_SERVER_TYPE	• Server 유형	session	X
LENA_HOME	• LENA 설치위치 • 기본 값 : /usr/local/lena	(설명참조)	X
LENA_SERVER_HOME	• Session Server 설치 위치 • 기본 값 : /usr/local/lena/server/sessionServer	(설명참조)	X
LOG_OUTPUT_TYPE	• 로그 출력 방식 (file/console)	console	○
LENA_AGENT_RUN	• LENA Agent 기동여부	N	○
LENA_USER	• Manager 기동에 사용할 OS 사용자계정	root	○
LENA_USER_GROUP	• Manager 기동에 사용할 OS 사용자그룹	root	○

JAVA_DOMAIN_CACHE_TTL 이 설정되었을 시 ${JAVA_HOME}/jre/lib/security/java.security 파일의 networkaddress.cache.ttl 값을 변경한다.

Directory 구조

LENA Image 기준 기본 설치 위치는 '/usr/local/lena' 이고, 그 하위 구조는 다음과 같다.

디렉토리 (${LENA_HOME} 하위)	설명	비고
bin	Session Server의 Start/Stop scripts	미사용
depot	설치를 위한 Local Repository	미사용
etc	기타 메타 정보 및 설정 파일	
license	License를 관리하는 디렉토리	
modules	LENA 제공 모듈의 저장소 Home	
servers/sessionServer ㄴlib ㄴlogs	Session Server 설치 위치 (${LENA_SERVER_HOME}) Session Server Library 저장소 Log 파일 저장소	
tmp	임시디렉토리	

Log

Session Server는 File Log 출력만 제공한다. 출력위치는 ${LENA_SERVER_HOME}/logs 디렉토리이며 파일명 형식은 lena-sessionServer-YYYYMMDD.log 으로 매일 Log 파일이 생성된다

Health Check

Health Check의 기본 내용은 본 문서 Manger Health Check 부분을 참조한다.

Kubernetes 기준 Session Server를 Health Check 하는 방식은 Command Exec (ExecAction) 방식이며, ${LENA_SERVER_HOME}/ health.sh를 호출한다.

1. Readiness Check
   - exec : ${LENA_SERVER_HOME}/ health.sh
   - initialDelaySeconds : 20
2. Liveness Check
   - exec : ${LENA_SERVER_HOME}/ health.sh
   - initialDelaySeconds : 30
   - periodSeconds : 5

2.6. Server 유형별 고려사항 – WAS

2.6.1. 배포

WAS는 Container에 배포되며, 설정에 따라 Container 플랫폼 (Kubernetes, ECS등)에 단일 또는 복수개의 Instance가 동시에 배포된다.

1. Service 계획
   WAS 는 단일/복수개의 Container로 배포되고, 이를 외부나 Front-End에 Service하기 위해서는 앞 단에 L/B역할을 수행하는 Service를 배치하는 것이 일반적인 방식이다. Kubernetes의 경우에는 설치된 Node의 특정 Port로 서비스하는 NodePort, 외부 L/B를 활용하는 LoadBalancer, 내부 고정 IP를 지정하는 ClusterIp의 서비스 유형이 제공되고 있으며, ECS의 경우에는 ALB를 지정하는 방식이 있다. 사전에 Application을 어떠한 방식으로 Service할지 사전 결정이 필요하다.
2. Instance 수 (Replica)
   단일 Service를 하는 복수개의 WAS의 수는 부하에 따라 가변적이나, 초기 기동해야 할 Instance 개수를 사전에 정의하여, 배포 설정에 반영하여야 한다.
3. Service Mapping
   ECS의 경우에는 Service에서 직접 L/B를 지정할 수 있지만, Kubernetes의 경우에는 Key-Value로 정의된 label을 기준으로 Mapping 룰을 정의 하여 Service와 매핑한다. 전체 시스템 상에서 중복없고, 운영에 편리한 Mapping 기준을 수립하여 배포 설정에 반영하여야 한다.

2.6.2. 사양

WAS를 운영하기 위해서 필요한 사양은 다음과 같다.

Memory

WAS의 Heap Memory Size는 최소 512Mbyte를 필요로 하고 Image에는 다음과 같이 기본 설정이 되어 있다.

- Heap Memory : 1024 Mbyte

- Metaspace Memory : 128 Mbyte

이를 변경하고자 하면 다음 환경변수를 설정하여 조정 한다.

- LENA_JVM_HEAP_SIZE
- LENA_JVM_METASPACE_SIZE

위 환경 변수의 값은 MByte단위이며 반드시 숫자+'m' 형태의 포맷 (예 : 1024m) 형식으로 지정되어야 한다. 포맷이 불일치 하면 적용되지 않는다.

Disk

LENA Image 기준으로 사용되는 Image 크기는 약 900 Mbyte로 이는 OS + JDK + LENA + Library의 총합으로 Image의 상위 Layer 용량까지 포함한 용량이다.
운영에 필요한 추가적인 Disk는 Log를 파일 방식으로 저장할 때 Log용량과 Application 소스 파일(Artifact)의 용량을 고려하여 산정한다.

2.6.3. 설정

네트워크

1. 네트워크 주소
   WAS의 네트워크 주소는 특별한 제약이 없다. N/W상에서 Manager와 Session Server를 목적지로 하는 단방향 통신 가능하도록 배치해야 한다.
2. 서비스 포트
   WAS의 서비스 포트는 다음과 같이 고정되어 있다.

   - HTTP 서비스 Port : 8180

   위 포트는 LENA Image에 고정되어 있고, 외부 서비스 변경을 하고자 하는 경우 LENA Image 변경보다는 Container의 기본 사상에 따라 Port 바인딩 설정을 통해 변경할 것을 권고한다.

환경변수

WAS Container에 적용 가능한 주요 환경변수는 다음과 같다.

환경변수	설명	기본 값	변경 가능
LENA_SERVICE_PORT	• WAS 서비스 Port	8180	○
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m	○
LENA_JVM_METASPACE_SIZE	• Metaspace Memory크기	128m	○
LENA_JVM_OPTIONS	• 사용자 정의 JVM OPTION		○
LENA_MANAGER_ADDRESS	• Manager 주소 • 형식 : IP / Domain주소 : 서비스 포트		○

환경변수	설명	기본 값	변경 가능
LENA_MANAGER_MONITORING_PORT	• Manager 모니터링 Port 정보	16100	○
LENA_MANAGER_KEY	• Manager Open API 접속 도큰		○
LENA_CONFIG_TEMPLATE_DOWNLOAD	• Manager도부터 설정 파일 다운도드 여부 • 허용값 : Y 또는 N		○
LENA_CONFIG_TEMPLATE_ID	• 설정 파일 ID • 형식 : Service Cluster 명:Revision 번호		○
LENA_LICENSE_DOWNLOAD_URL	• License 다운도드 URL • 입뎍값 : manager 또는 고객사 보유License다운도드 URI	manager	○
LENA_CONTRACT_CODE	• License받급과 관련된 계약 코드도 암호화된 값임.		○
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	3	○
LOG_OUTPUT_TYPE	• LOG 춛뎍 유형 • 허용 값 : console 또는 file	console	○
LENA_LOG_OUTPUT_DIR	• Log 파일 생성 위치	/usr/local/lena/servers/appServer/logs	○
LENA_DUMP_OUTPUT_DIR	• Dump 파일 생성 위치		○
LENA_SERVER_TYPE	• Server 유형	WAS	X
LENA_HOME	• LENA 설치 Home • 값 : /usr/local/lena	(설명참조)	X
LENA_SERVER_HOME	• LENA 서버 설치 위치 • 값 : /usr/local/lena/servers/appServer	(설명참조)	X
LENA_SERVICE_ENDPOINT	• WAS 가 속한 서비스의 주소		○
LENA_AGENT_RUN	• LENA Agent 기동여부	N	○

환경변수	설명	기본 값	변경 가능
LENA_USER	• Manager 기동에 사용할 OS 사용자계정	root	○
LENA_USER_GROUP	• Manager 기동에 사용할 OS 사용자그룹	root	○
LENA_HEALTH_CHECK	• Health Check 수행여부	N	○
LENA_HEALTH_CHECK_WAS_URL	• Health Check 용 페이지 정보	/tie/lenaHealthCheck.jsp	○
LENA_HEALTH_CHECK_INITIAL_DELAY_MILLISEC	• LENA Agent 기동 이후 Health Check 시작전 대기시간, Server 기동시간을 확보하기 위함	60000 (milliseconds)	○
LENA_HEALTH_CHECK_TIMEOUT_MILLISEC	• Health Check 요청 Timeout	5000 (milliseconds)	○
LENA_HEALTH_CHECK_FAILURE_THRESHOLD	• Health Check 실패 임계치	5	○
LENA_HEALTH_CHECK_TERM_EXECUTION	• Health Check 실패 임계치 초과시, 후속작업 수행여부	true	○
LENA_HEALTH_CHECK_TERM_EXECUTION_SCRIPT	• Health Check 실패 임계치 초과시, 후속작업 script 정보	stop-container	○
LENA_HEALTH_CHECK_TERM_EXECUTION_INTERVAL	• Health Check 실패 임계치 초과시, 후속작업 수행 주기	300 (seconds)	○

- LENA_CONFIG_TEMPLATE_ID: Revision 번호는 생략가능하며, 생략시 Default Revision 다운로드 된다.
- LENA_CONTRACT_CODE: License 유효성 체크에 사용되고, 이 값이 유효하지 않은 경우 라이선스 다운로드가 취소된다.
- JAVA_DOMAIN_CACHE_TTL: 이 값이 설정되었을 시 ${JAVA_HOME}/jre/lib/security/java.security 파일의 networkaddress.cache.ttl 값을 변경한다.
- LOG_OUTPUT_TYPE: Server 설정파일 다운로드 적용시 다운로드 한 설정 파일의 Log설정이 적용된다.

Directory 구조

LENA Image 기준 기본 설치 위치는 '/usr/local/lena' 이고, 그 하위 구조는 다음과 같다.

디렉토리 (${LENA_HOME} 하위)	설명	비고
bin	Node Agent의 Start/Stop scripts	미사용
depot	설치를 위한 Local Repository	미사용
etc	기타 메타 정보 및 설정 파일	
license	License 정보를 관리하는 디렉토리	
logs	LENA 관리용 로그 파일 저장소	
modules	LENA 제공 모듈의 저장소 Home	
ㄴ lena-agent	Node Agent 실행에 필요한 모듈이 위치하는 경로	미사용
servers/webServer	Server 설치 Home, ${LENA_SERVER_HOME}	
ㄴbin	Server Start / Stop / 관리용 실행 Script 저장소	
ㄴconf	Server 설정정보 저장소	
ㄴdumps	Dump 파일 저장소	
ㄴhook	Life-Cycle Hook Shell 파일 저장소	
ㄴlib	Server 실행 Library 저장소	
ㄴlogs	Log 파일 저장소	
ㄴtemp	작업용 임시 디렉토리	
ㄴwebapps	기본 Application Deployment 디렉토리	
ㄴwork	JSP Servlet 변환 소스 및 컴파일 결과 저장소	
tmp	LENA 관리용 임시 디렉토리	

Log & Dump 출력

Log는 Standard Out / Error로 출력하는 'console' 방식과 File로 출력하는 'file' 방식이 지원되고, 환경변수 'LOG_OUTPUT_TYPE'의 값을 'console' 또는 'file'로 설정 함으로서 출력방식을 전환할 수 있다.

1. Console 출력
   일반적으로 Container 환경에서 많이 활용되는 방식이다. Server의 Application Log, Access Log, GC Log를 모두 Standard Out으로 출력한다. Docker에 설정된 Log Driver에 의해 Node(Host)의 지정위치 (Docker의 기본 Log 파일 위치 : /var/lib/docker/containers/[container-id]/[container-id]-json.log)에 저장되거나 FluentD와 같은 Log Aggregator에 의해 수집 / 저장하여 통합관리를 할 수 있다.
2. 파일 출력
   파일 출력 설정 시 Log파일 및 Dump 파일은 ${LENA_HOME}/servers/appServer/logs 디렉토리 하위에 파일로 저장되고, 각 Log 파일은 logrotate설정에 의해 Daily rolling 된다.
   출력되는 Log 파일의 유형과 출력 파일명은 아래 표와 같다.

Log 유형	출력 위치
Access Log	access_appServer_${HOSTNAME}.log
GC Log	gc_appServer_${HOSTNAME}.log

Log 유형	출력 위치
Application Log	appServer_lena-${HOSTNAME}.out.log

이 Log파일들은 Guest OS에 설치된 logrotate에 의해 매일 Rolling 된다.
파일 기반 Log를 출력할 경우, 별도의 Log Aggregator를 통해 Log를 수집 / 조회하는 Log관리 Stack (ELK, EFK Stack등)을 구성하는 것이 일반적이다. 이를 위해서는 Fluent-Bit과 같은Side-car Container를 추가하여 Log를 수집하는 방식이 일반적이다.

Dump파일은 다음 위치에 생성된다.

- Dump Home : ${LENA_HOME}/servers/appServer/dumps/ ${HOSTNAME}

${HOSTNAME} 디렉토리는 외부 Volume으로 Dump파일을 저장할 때 Container별 구분을 위한 것이다.

Dump 유형별 저장 위치는 다음과 같다.

- Heap Dump : ${DUMP_HOME}/hdump
- Thread Dump : ${DUMP_HOME}/tdump
- Service Dump : ${DUMP_HOME}/sdump

Health Check

Health Check의 기본 내용은 본문서 Manager Health Check 부분을 참조한다.

일반적으로 WAS는 Http Get방식으로 Health Check 하지만, LENA WAS의 기본 Health Check방식은 TCP Port 체크방식으로 설정되어 있다. 이는 기본 LENA Image에 Business Application가 탑재되지 않았기 때문이고, Biz Application이 탑재되었을 경우에는 해당 Application의 적절한 Http Get Health Check 설정을 업데이트 하기를 권고한다. 제공 Kubernetes Manifest 파일 기준 기본설정은 다음과 같다.

1. Readiness Check
   - TCPSocketAction : port 8180
   - initialDelaySeconds : xx (Application 특성에 따른 보정 필요)
   - timeoutSeconds : xx (Application 특성에 따른 보정 필요)
2. Liveness Check
   - TCPSocketAction : port 8180
   - initialDelaySeconds : xx (Application 특성에 따른 보정 필요)
   - periodSeconds : xx (Application 특성에 따른 보정 필요)

- LENA WAS의 경우 Server의 정상 기동 후에 Service Port가 Listen 상태로 변경된다.
- Health Check를 위한 Page는 Check가 성공하면 정상적으로 서비스가 제공되는 것으로 판단하게 되므로, 서비스의 Back-end (예 : Database)까지 정상적인지를 판단할 수 있는 Page를 선정하여 적용하기를 권고한다.

Server Configuration 관리

Container와 WAS가 기동되는 절차는 다음과 같다.

일반적으로Server 설정은 WAS가 기동되기 전에 적용되어야 하며, 설정을 적용하는 시점은 1) Base Image에 포함하거나 2) Container기동시점에 반영하는 방법 3) Application Artifact에 포함하는 방식이 있다.

설정 적용 방식	설명	LENA 기능지원
Base Image에 포함	Base Image 생성시에 설정정보를 COPY하여 Image에 포함	○
기동 시점에 반영	Container 기동시, 외부 Repository에서 설정 정보를 COPY	○
Application Artifact에 포함	Spring Boot의 경우와 같이 Application Artifact에 Server설정 정보가 간소화 되어 포함되어 Application Artifact의 배포와 함께 적용	○ WAS(Embedded) 를 통해 제공

LENA에서는 Manager를 통해 미리 구성한 Server 설정정보를 Image Build시점이나, Container기동 시점에 반영할 수 있고 , 이를 위해서는 Application / WEB Server Container 구성이전에 Manager를 설치하고, Server 설정정보를 구성하여야 한다. 이를 도식화 하면 다음과 같다.

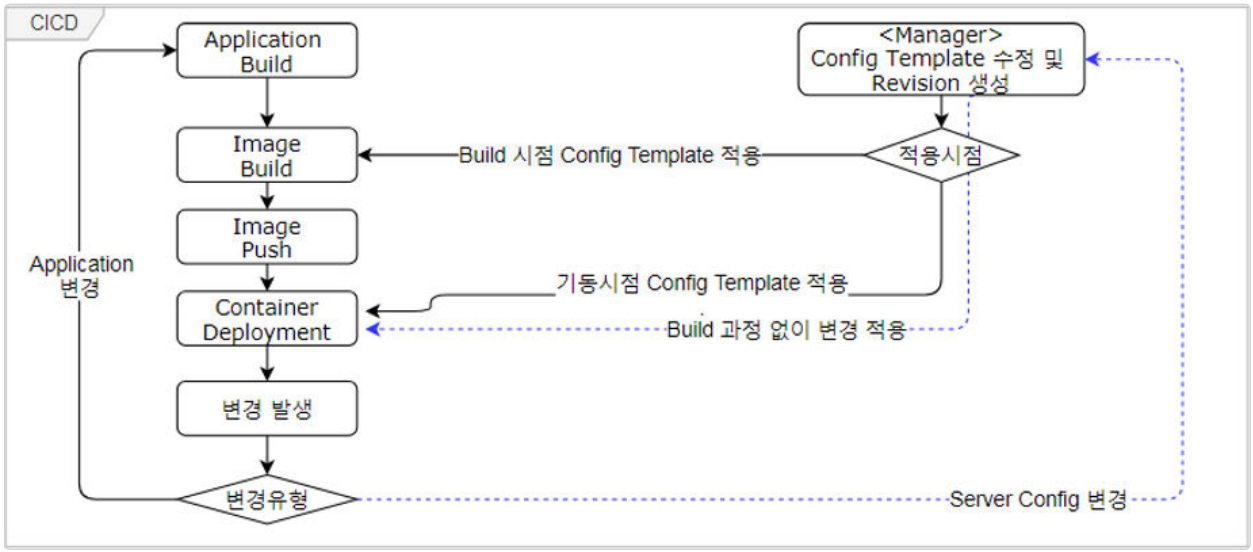


Figure 16. Service Cluster 설정관리 및 Container 설정 적용

LENA의 Server Cluster 기능은 이용하여 Server의 설정은 미리 구성하고, 이를 Image 또는 기동시점에 반영하여야 한다.

Server 설정에 대한 상세 가이드는 별도로 제공되는 운영자매뉴얼을 참조한다.

Container Image Build

LENA가 Base Image를 제공하지만 Project / 고객사 정책 또는 Architecture 표준에 의해 Base Image를 상속 또는 신규로 Build해야 하는 경우가 발생 할 수 있다. Architecture 의사결정과정에서 LENA Image의 변경범위를 확인하고, Image 관리 정책 / 기준 수립이 필요하다.

다음은 신규로 Base Image를 Build해야 하는 원인이다.

- OS가 LENA 기본 제공 Image와 불일치
- LENA 설치 위치의 변경 (기본 : /usr/local/lena)
- 기타 적용 시스템의 Architecture 표준과 LENA 제공 기본 Image의 불일치 격차가 환경설정 수정하는 것만으로는 불가능

위 경우에는 LENA기술지원은 통해 Base Image를 재생성 하여야 한다.

아래 예시와 같이 단순 변경이 필요한 경우, LENA Image를 상속하여 Base Image를 생성(Build)한다.

- Application Artifact의 배치
- 필요 Library 설치
- JDK 변경 (기존 Image에의 영향이 환경변수 수정 수준 이하일 경우)
- 실행 Command Script 수정
- 기타 적용 시스템의 Architecture 표준과 LENA 제공 기본 Image의 붇일치 격차가 환경설정 수정만으도 가능한 경우

상세 실행 가이드는 본문서 Base Image 생성 부분을 참조한다.

Application 배포

Container 관점과 Server 관점에서의 Application Artifact 배포에 대한 방식을 모두 고려하여야 한다. Project에서 배포 절차가 결정되면, Kubernetes Deployment Manifest나 ECS의 Task정의를 결정된 방식에 따라 설정하여야 한다.

WAS 관점에서 Application을 Deployment하는 방식에는 다음 두 가지가 있다.

Server 관점 배포 방식	설명
기본 Deployment 디렉도리에 배포	${LENA_SERVER_HOME}/webapps에 WAR 또는 Directory를 복사
개별 Application 설정에 따든 배포	Manager를 통해 개별 Application 설정 Application설정 중 'DocBase' 위치에 WAR 또는 Directory를 복사

Container 관점에서의 Deployment 방식은 다음 두가지 있다.

Container 배포 방식	설명
주 Container Image에 사전 배포 방식	Base Image에 Application Artifact를 직접 복사 / 포함
Init Container 환용 기동 시점 배포 방식	Init Container에서 내부에 포함된 Artifact 또는 외부 저장소 (Volume)에 있는 Artifact를 기동 시점에 주 Container에 복사

2.7. Server 유형별 고려사항 – Embedded WAS

2.7.1. 배포

Embedded WAS는 Container에 배포되며, 설정에 따라 Container 플랫폼 (Kubernetes, ECS등)에 단일 또는 복수개의 Instance가 동시에 배포된다.

1. Service 계획
   Embedded WAS 는 단일/복수개의 Container도 배포되고, 이를 외부나 Front-End에 Service하기 위해서는 앞 단에 L/B역핟을 수행하는 Service를 배치하는 것이 일반적인 방식이다. Kubernetes의

경우에는 설치된 Node의 특정 Port로 서비스하는 NodePort, 외부 L/B를 활용하는 LoadBalancer, 내부 고정 IP를 지정하는 ClusterIp의 서비스 유형이 제공되고 있으며, ECS의 경우에는 ALB를 지정하는 방식이 있다. 사전에 Application을 어떠한 방식으로 Service할지 사전 결정이 필요하다.

2. Instance 수 (Replica)
   단일 Service를 하는 복수개의 Embedded WAS의 수는 부하에 따라 가변적이나, 초기 기동해야 할 Instance 개수를 사전에 정의하여, 배포 설정에 반영하여야 한다.
3. Service Mapping
   ECS의 경우에는 Service에서 직접 L/B를 지정할 수 있지만, Kubernetes의 경우에는 Key-Value로 정의된 label을 기준으로 Mapping 룰을 정의 하여 Service와 매핑한다. 전체 시스템 상에서 중복없고, 운영에 편리한 Mapping 기준을 수립하여 배포 설정에 반영하여야 한다.

2.7.2. 사양

Embedded WAS를 운영하기 위해서 필요한 사양은 다음과 같다.

Memory

Embedded WAS의 Heap Memory Size는 최소 512Mbyte를 필요로 하고 Image에는 다음과 같이 기본 설정이 되어 있다.

- Heap Memory : 1024 Mbyte
- Metaspace Memory : 128 Mbyte

이를 변경하고자 하면 다음 환경변수를 설정하여 조정 한다.

- LENA_JVM_HEAP_SIZE
- LENA_JVM_METASPACE_SIZE

위 환경 변수의 값은 MByte단위이며 반드시 숫자+'m' 형태의 포맷 (예 : 1024m) 형식으로 지정되어야 한다. 포맷이 붙일치 하면 적용되지 않는다.

Disk

LENA Image 기준으로 사용되는 Image 크기는 약 700 Mbyte로 이는 OS + JDK + LENA + Library의 총합으로 Image의 상위 Layer 용량까지 포함한 용량이다.
운영에 필요한 추가적인 Disk는 Log를 파일 방식으로 저장할 때 Log용량과 Application 소스 파일(Artifact)의 용량을 고려하여 산정한다.

2.7.3. 설정

네트워크

1. 네트워크 주소
   Embedded WAS의 네트워크 주소는 특별한 제약이 없다. N/W상에서 Manager와 Session Server를 목적지로 하는 단방향 통신 가능하도록 배치해야 한다.
2. 서비스 포트
   Embedded WAS의 서비스 포트는 다음과 같이 고정되어 있다.

   - HTTP 서비스 Port : 8180

   위 포트는 LENA Image에 고정되어 있고, 외부 서비스 변경을 하고자 하는 경우 LENA Image 변경보다는 Container의 기본 사상에 따라 Port 바인딩 설정을 통해 변경할 것을 권고한다.

환경변수

Embedded WAS Container에 적용 가능한 주요 환경변수는 다음과 같다.

환경변수	설명	기본 값	변경 가능
LENA_SERVICE_PORT	• WAS 서비스 Port	8180	○
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m	○
LENA_JVM_METASPACE_SIZE	• Metaspace Memory크기	128m	○
LENA_JVM_OPTIONS	• 사용자 정의 JVM OPTION		○
LENA_MANAGER_ADDRESS	• Manager 주소 • 형식 : IP / Domain주소 : 서비스 포트		○
LENA_MANAGER_MONITORING_PORT	• Manager 모니터링 Port 정보	16100	○
LENA_CONFIG_TEMPLATE_ID	• 설정 파일 ID • 형식 : Service Cluster 명		○
LENA_SPRING_PROFILES_ACTIVE	• SPRING PROFILE (PROFILE 설정 우선순위가 가장 높음. 반드시 이 값은 수정 해야함)	default	○
LOG_OUTPUT_TYPE	• LOG 출력 유형 • 허용 값 : console 또는 file	console	○
LENA_LOG_OUTPUT_DIR	• Log 파일 생성 위치	/usr/local/lena/logs	○
LENA_SERVER_TYPE	• Server 유형	embedded	X
LENA_HOME	• LENA 설치 Home • 값 : /usr/local/lena	(설명참조)	X
LENA_SERVICE_ENDPOINT	• WAS 가 속한 서비스의 주소		○
LENA_HEALTH_CHECK	• Health Check 수행여부	N	○
LENA_APP_FILE	• Application Jar 파일 명		○

환경변수	설명	기본 값	변경 가능
LENA_APP_DIR	• Application Jar 디렉토리 명	/usr/local/lena	○
LENA_EXCEPTION_ALERT_ENABLE	• Exception 발생 시 정보 수집여부	false	○
LENA_EXCEPTION_CLASS_PATTERNS	• 수집대상 Exception Class 정보 ',' 로 여러개 Class 를 연결하여 설정		○
LENA_EXCEPTION_EXCLUDE_CLASS_PATTERNS	• 제외대상 Exception Class 정보 ',' 로 여러개 Class 를 연결하여 설정		○
LENA_FULLSTACK_HOOKED_EXCEPTION_ENABLE	• Exception 발생시 Full Stack Trace 수집여부	true	○
LENA_STUCKTHREAD_ALERT_ENABLE	• Thread Stuck 발생 시 정보 수집여부	false	○
LENA_OOM_ALERT_ENABLE	• Out Of Memory 발생 시 정보 수집여부	true	○
LENA_FULLGC_ALERT_ENABLE	• Full GC 발생 시 정보 수집여부	false	○
LENA_REVERSE_TCP_CONNECTION_ENABLE	• Reverse TCP Connection 을 통한 Manager 연결 사용 여부	true	○

Directory 구조

LENA Image 기준 기본 설치 위치는 '/usr/local/lena' 이고, 그 하위 구조는 다음과 같다.

디렉토리 (${LENA_HOME} 하위)	설명	비고
logs	LENA 관리용 로그 파일 저장소	
etc/info	Image Build 정보 파일 저장소	

Log & Dump 출력

Log는 Standard Out / Error로 출력하는 'console' 방식과 File로 출력하는 'file' 방식이 지원되고, 환경변수 'LOG_OUTPUT_TYPE'의 값을 'console' 또는 'file'로 설정 함으로서 출력방식을 전환할 수 있다.

1. Console 출력
   일반적으로 Container 환경에서 많이 활용되는 방식이다. Server의 Application Log, Access Log, GC Log를 모두 Standard Out으로 출력한다. Docker에 설정된 Log Driver에 의해 Node(Host)의 지정위치 (Docker의 기본 Log 파일 위치 : /var/lib/docker/containers/[container-id]/[container-id]-json.log)에 저장되거나 FluentD와 같은 Log Aggregator에 의해 수집 / 저장하여 통합관리를 할 수 있다.
2. 파일 출력
   파일 출력 설정 시 Log파일 및 Dump 파일은 ${LENA_HOME}/servers/appServer/logs 디렉토리 하위에 파일로 저장되고, 각 Log 파일은 logrotate설정에 의해 Daily rolling 된다.

출력되는 Log 파일의 유형과 출력 파일명은 아래 표와 같다.

Log 유형	출력 위치
Access Log	access_appServer_${HOSTNAME}.log
GC Log	gc_appServer_${HOSTNAME}.log
Application Log	appServer_lena-${HOSTNAME}.out.log

이 Log파일들은 Guest OS에 설치된 logrotate에 의해 매일 Rolling 된다.
파일 기반 Log를 출력할 경우, 별도의 Log Aggregator를 통해 Log를 수집 / 조회하는 Log관리 Stack (ELK, EFK Stack등)을 구성하는 것이 일반적이다. 이를 위해서는 Fluent-Bit과 같은Side-car Container를 추가하여 Log를 수집하는 방식이 일반적이다.

Dump파일은 다음 위치에 생성된다.

- Dump Home : ${LENA_HOME}/servers/appServer/dumps/ ${HOSTNAME}

${HOSTNAME} 디렉토리는 외부 Volume으로 Dump파일을 저장할 때 Container별 구분을 위한 것이다.

Dump 유형별 저장 위치는 다음과 같다.

- Heap Dump : ${DUMP_HOME}/hdump
- Thread Dump : ${DUMP_HOME}/tdump
- Service Dump : ${DUMP_HOME}/sdump

Health Check

Health Check의 기본 내용은 본문서 Manager Health Check 부분을 참조한다.

일반적으로 WAS는 Http Get방식으로 Health Check 하지만, LENA WAS의 기본 Health Check방식은 TCP Port 체크방식으로 설정되어 있다. 이는 기본 LENA Image에 Business Application가 탑재되지 않았기 때문이고, Biz Application이 탑재되었을 경우에는 해당 Application의 적절한 Http Get Health Check 설정을 업데이트 하기를 권고한다. 제공 Kubernetes Manifest 파일 기준 기본설정은 다음과 같다.

1. Readiness Check
   - TCPSocketAction : port 8180
   - initialDelaySeconds : xx (Application 특성에 따른 보정 필요)
   - timeoutSeconds : xx (Application 특성에 따른 보정 필요)
2. Liveness Check
   - TCPSocketAction : port 8180
   - initialDelaySeconds : xx (Application 특성에 따른 보정 필요)
   - periodSeconds : xx (Application 특성에 따른 보정 필요)

- LENA WAS의 경우 Server의 정상 기동 후에 Service Port가 Listen 상태로 변경된다.
- Health Check를 위한 Page는 Check가 성공하면 정상적으로 서비스가 제공되는 것으로 판단하게 되므로, 서비스의 Back-end (예 : Database)까지 정상적인지를 판단할 수 있는 Page를 선정하여 적용하기를 권고한다.

Container Image Build

LENA가 Base Image를 제공하지만 Project / 고객사 정책 또는 Architecture 표준에 의해 Base Image를 상속 또는 신규로 Build해야 하는 경우가 발생 할 수 있다. Architecture 의사결정과정에서 LENA Image의 변경범위를 확인하고, Image 관리 정책 / 기준 수립이 필요하다.

다음은 신규로 Base Image를 Build해야 하는 원인이다.

- OS가 LENA 기본 제공 Image와 불일치
- LENA 설치 위치의 변경 (기본 : /usr/local/lena)
- 기타 적용 시스템의 Architecture 표준과 LENA 제공 기본 Image의 불일치 격차가 환경설정 수정하는 것만으로는 불가능

위 경우에는 LENA기술지원을 통해 Base Image를 재생성 하여야 한다.

아래 예시와 같이 단순 변경이 필요한 경우, LENA Image를 상속하여 Base Image를 생성(Build)한다.

- Application Artifact의 배치
- 필요 Library 설치
- JDK 변경 (기존 Image에의 영향이 환경변수 수정 수준 이하일 경우)
- 실행 Command Script 수정
- 기타 적용 시스템의 Architecture 표준과 LENA 제공 기본 Image의 불일치 격차가 환경설정 수정만으로 가능한 경우

상세 실행 가이드는 본문서 Base Image 생성 부분을 참조한다.

Application 배포

Container 관점과 Server 관점에서의 Application Artifact 배포에 대한 방식을 모두 고려하여야 한다. Project에서 배포 절차가 결정되면, Kubernetes Deployment Manifest나 ECS의 Task정의를 결정된 방식에 따라 설정하여야 한다.

WAS 관점에서 Application을 Deployment하는 방식에는 다음 두 가지가 있다.

Server 관점 배포 방식	설명
기본 Deployment 디렉토리에 배포	${LENA_APP_FILE} 설정 ${LENA_HOME}에 Jar 를 복사
개별 Application 설정에 따른 배포	${LENA_APP_FILE} 과 ${LENA_APP_DIR} 설정 ${LENA_APP_DIR}에 Jar 를 복사

Container 관점에서의 Deployment 방식은 다음 두가지 있다.

Container 배포 방식	설명
주 Container Image에 사전 배포 방식	Base Image에 Application Artifact를 직접 복사 / 포함
Init Container 활용 기동 시점 배포 방식	Init Container에서 내부에 포함된 Artifact 또는 외부 저장소 (Volume)에 있는 Artifact를 기동 시점에 주 Container에 복사

2.8. Server 유형별 고려사항 – Web Server

2.8.1. 배포

본 문서 “WAS 배포” 부분의 설명을 참조한다.

2.8.2. 사양

Web Server를 운영하기 위해서 필요한 사양은 다음과 같다.

Memory

Web Server의 Heap Memory Size는 최소 512Mbyte를 필요로 하고, Agent의 Heap Memory Size는 64~256MB를 필요로 한다.

Disk

LENA Image 기준으로 사용되는 Image 크기는 약 900 Mbyte로 이는 OS + JDK
LENA + Library의 총합으로 Image의 상위 Layer 용량까지 포함한 용량이다.

운영에 필요한 추가적인 Disk는 Log를 파일 방식으로 저장할 때 Log용량과 Web 리소스 파일 (Artifact)의 용량을 고려하여 산정한다.

2.8.3. 설정

네트워크

1. 네트워크 주소
   Web Server의 네트워크 주소는 특별한 제약이 없다. N/W상에서 Manager와 WAS 또는 WAS의 Service Endpoint를 목적지로 하는 단방향 통신 가능하도록 설정해야 한다.
2. 서비스 포트
   Web Server의 서비스 포트는 다음과 같이 고정되어 있다.
   - HTTP 서비스 Port : 7180
   - HTTPS 서비스 Port : 7543

위 포트는 LENA Image에 고정되어 있고, 외부 서비스 변경을 하고자 하는 경우 LENA Image 변경보다는 Container의 기본 사상에 따라 Port 바인딩 설정을 통해 변경할 것을 권고한다.

환경변수

WEB Server Container에 적용 가능한 주요 환경변수는 다음과 같다.

환경변수	설명	기본 값	변경 가능
LENA_MANAGER_ADDRESS	• Manager 주소 • 형식 : IP / Domain주소 : 서비스 포트		○
LENA_MANAGER_KEY	• Manager Open API 접속 도큰		○

환경변수	설명	기본 값	변경 가능
LENA_SERVICE_PORT	• WAS 서비스 Port	7180	○
LENA_CONFIG_TEMPLATE_DOWNLOAD	• Manager로부터 설정 파일 다운로드 여부 • 허용값 : Y 또는 N		○
LENA_CONFIG_TEMPLATE_ID	• 설정 파일 ID • 형식 : Service Cluster 명:Revision 번호		○
LENA_LICENSE_DOWNLOAD_URL	• License 다운로드 URL • 입력값 : manager 또는 고객사 보유License다운로드 URI	manager	○
LENA_CONTRACT_CODE	• License받급과 관련된 계약 코드도 암호화된 값임.		○
LOG_OUTPUT_TYPE	• LOG 춛력 유형 • 허용 값 : console 또는 file	console	○
LENA_LOG_OUTPUT_DIR	• Log 파일 생성 위치	/usr/local/lenaw/servers/webServer/logs	○
LENA_AGENT_RUN	• Node Agent 기동 여부	Y	○
LENA_SERVER_TYPE	• Server 유형	web	X
LENA_HOME	• LENA 설치 Home • 값 : /usr/local/lena	(설명참조)	X
LENA_SERVER_HOME	• LENA 서버 설치 위치 ○ 값 : /usr/local/lenaw/servers/webServer	(설명참조)	X
LENA_USER	• Manager 기동에 사용핟 OS 사용자계정	root	○
LENA_USER_GROUP	• Manager 기동에 사용핟 OS 사용자그룹	root	○
LENA_HEALTH_CHECK	• Health Check 수행여부	N	○
LENA_HEALTH_CHECK_FAILURE_THRESHOLD	• Health Check 실패 임계치	5	○

환경변수	설명	기본 값	변경 가능
LENA_HEALTH_CHECK_TERM_EXECUTION	• Health Check 실패 임계치 초과시, 후속작업 수행여부	true	○
LENA_HEALTH_CHECK_TERM_EXECUTION_SCRIPT	• Health Check 실패 임계치 초과시, 후속작업 script 정보	stop-container	○
LENA_HEALTH_CHECK_TERM_EXECUTION_INTERVAL	• Health Check 실패 임계치 초과시, 후속작업 수행 주기	300 (seconds)	○

- LENA_CONFIG_TEMPLATE_ID: Revision 번호는 생략가능하며, 생략시 Default Revision 다운로드 된다.
- LENA_CONTRACT_CODE: License 유효성 체크에 사용되고, 이 값이 유효하지 않은 경우 라이선스 다운로드가 취소된다.
- LOG_OUTPUT_TYPE: Server 설정파일 다운로드 적용시 다운로드 한 설정 파일의 Log설정이 적용된다.
- LENA_AGENT_RUN: Web Server는 Agent가 기동되어야 Manager에 등록 및 모니터링이 가능하다.

Directory 구조

LENA Image 기준 기본 설치 위치는 '/usr/local/lenaw' 이고, 그 하위 구조는 다음과 같다.

디렉토리 (${LENA_HOME} 하위)	설명	비고
bin	Node Agent의 Start/Stop scripts	
depot	설치를 위한 Local Repository	미사용
etc	기타 메타 정보 및 설정 파일	
license	License 정보를 관리하는 디렉토리	
logs	LENA 관리용 로그 파일 저장소	
modules ㄴ lena-agent ㄴ lena-web-pe	LENA 제공 모듈의 저장소 Home Node Agent 실행에 필요한 모듈이 위치하는 경로 WEB Server Library	
servers/webServer ㄴbin ㄴcache ㄴconf ㄴhook ㄴhtdocs ㄴlogs ㄴtemp	Server 설치 Home, ${LENA_SERVER_HOME} Server Start / Stop / 관리용 실행 Script 저장소 Cache 정보 저장소 Server 설정정보 저장소 Life-Cycle Hook Shell 파일 저장소 기본 Web 리소스 저장 디렉토리 Log 파일 저장소 작업용 임시 디렉토리	
tmp	LENA 관리용 임시 디렉토리	

Log

Log는 Standard Out / Error도 출력하는 'console' 방식과 File도 출력하는 'file' 방식이 지원되고, 환경변수 'LOG_OUTPUT_TYPE'의 값은 'console' 또는 'file'도 설정 함으도서 출력방식은 전환할 수 있다.

1. Console 출력
   본 문서 "Log & Dump 출력"을 참조한다.
2. 파일 출력
   파일 출력 설정 시 Log파일 및 Dump 파일은 ${LENA_HOME}/servers/webServer/logs 디렉도리 하위에 파일도 저장되고, 각 Log 파일은 logrotate설정에 의해 Daily rolling 된다.

출력되는 Log 파일의 유형과 출력 파일명은 아래 표와 같다.

Log 유형	출력 파일명	설명
Error Log	error_webServer.log	Web 서버 오듀 도그
Access Log	access_webServer.log	Access 도그
Trace Log	trace_webServer.log	서비스 추적용 도그, Container형 서버에는 사용하지 않음
NTrace Log	ntrace_webServer.log	
LSC Log	lsc_webServer.log	

Health Check

일반적으도 WAS는 Http Get방식으도 Health Check 하지만, LENA Web Server의 기본 Health Check방식은 TCP Port 체크방식으도 설정되어 있다. 이는 기본 LENA Image에 Business Application가 답재되지 않았기 때문이고, Biz Application이 답재되었은 경우에는 해당 Application의 적젇한 Http Get Health Check 설정은 업데이트 하기를 권고한다. 제공 Kubernetes Manifest 파일 기준 기본설정은 다음과 같다.

1. Readiness Check
   - TCPSocketAction : port 7180
   - initialDelaySeconds : xx (Application 득성에 따든 보정 필요)
   - timeoutSeconds : xx (Application 득성에 따든 보정 필요)
2. Liveness Check
   - TCPSocketAction : port 7180
   - initialDelaySeconds : xx (Application 득성에 따든 보정 필요)
   - periodSeconds : xx (Application 득성에 따든 보정 필요)

Server Configuration 관리

본 문서 "Server Configuration 관리" 릍 참조한다.

Container Image Build

본 문서 "Container Image Build" 릍 참조한다.

Chapter 3. 설치 공통 사항

3.1. 설치 준비

- <Container 운영환경 점검>
  설치를 위해서는 Kubernetes, ECS등 Container가 운용될 수 있는 환경이 우선 구성되어 있어야 한다. 그리고, CLI환경에서 해당 환경에 접근할 수 있는 적절한 권한과 Profile (예 : Kubernetes Config) 설정이 필요하며, Manager 또는 Session Server를 VM에 설치할 경우 설치 대상 VM이 준비되어 있어야 한다.
- <Network 환경 점검>
  시스템 구성 Server간 통신이 가능한지 점검하여야 한다. Container N/W 내부에 모두 설치되는 경우는 일반적으로 통신 제약이 없지만, VM 혼합 방식으로 구성되는 경우 Manager Server, Web Server, WAS, Session Server간 통신이 가능한지 사전 확인이 필요하다. 다음은 LENA Image 기준으로 Service 또는 상호 통신을 위해 필요한 N/W Port 이다.

Source	Target	Port	용도
WEB/WAS/Session Server	Manager	TCP 7700 TCP 16100 UDP 16100	Manager 서비스 제공 모니터링, 등록
Bastion, 관리자 PC	Manager	TCP 7700	Manager 서비스 제공
L/B , Front End	WEB Server	TCP 7180	서비스 제공
L/B , Front End	WAS	TCP 8180	서비스 제공
WAS	Session Server	TCP 5180	서비스 제공

- <Image 접근 환경 점검>
  LENA 공개 Image를 활용하기 위해서는 Container 플랫폼에서 Docker Hub에 접속이 가능해야 한다. 만약 접근이 불가능 하다면, 접근 가능한 환경에서 LENA Image를 Pull 받고 (docker pull), 이를 파일로 저장하여 (docker save) 해당 플랫폼에서 접근 가능한 환경에 적재 (docker load) 하여야 한다. 아래는 이를 처리하는 예시 이다.

```
$ sudo docker pull lenacloud/lena-cluster: 1.3.1.0_3-jdk8-openjdk
... <춘덕 생략>
$ sudo docker save -o lena-cluster.tar lenacloud/lena-cluster:1.3.1n.0_3-jdk8-
openjdk
... <춘덕 생략>
$ sudo docker load -i lena-cluster.tar
... <춘덕 생략>
```

3.2. Base Image 생성

WAS 나 WEB Server의 경우 Base Image를 재생성해야 필요가 발생할 수 있다. 다음에서는 LENA Image를 상속하여 Base Image를 생성(Build) 에 대해 설명한다. Base Image생성은 LENA Image를 활용하는 Dockerfile을 작성하고, 이 파일로 Docker Image를 생성하고 공유 Repository에 등록하는 전체 과정이다.

예상되는 Base Image의 주요 변경 사항은 다음과 같은 Case가 있다.

1. 필수 Library 추가
   OS별 설치 명령어 (yum, apt-get 등)를 이용하거나, 직접 설치한다.
2. JDK의 변경
   LENA Base Image의 JDK를 변경하고자 할 때, 설치 명령어 또는 직접 압축해제 방식으로 설치한다.
3. Application Deployment
   Application Deployment기본 위치는 '/usr/local/lena/server/appServer/webapps' 이다.
   여기에 직접 WAR 파일 또는 Exploded된 Directory를 복사하여 Deployment 할 수 있다. 이 경우에는 WAR 파일명 또는 Directory명이 Context Path로 지정된다.
   다른 방식으로는 개별 Application 설정을 통해 지정된 위치(DocBase)에 WAR 또는 Directory를 복사하는 방식으로도 Deploy 할 수 있다. 이 경우에는 설정에서 지정한 Context Path로 Application이 서비스 된다.
   상세 설정은 별도로 제공되는 '운영자매뉴얼'의 Application 설정 부분을 참조한다.
4. 실행 Command Script의 수정
   LENA의 기본 Container 실행 Command는 ${LENA_HOME}/docker-entrypoint.sh 이다. 초기 환경변수에 따른 환경설정 및 Server 설정파일 다운로드, License 다운로드, Server의 실행 등이 실행되는 Shell Script로서, Project 환경에 맞도록 변경이 필요한 경우 이를 수정하여 적용한다.

3.2.1. Base Image 생성 절차

Base Image의 생성 절차는 다음과 같다.

1. Dockerfile 작성
2. Docker Image 빌드
3. Docker Image 등록

Dockerfile 작성

다음 코드는 Base Image를 생성하기 위한 Dockerfile 예시이며, CentOS 기반의 Application Server 를 기준으로 작성하였다.

Dockerfile 샘플

```
# This is a sample of Dockerfile to build project's Base Image with LENA
template.
# LENA image provides just basic environment to run LENA WAS.
# The project and the customer are responsible for optimization or change the
environments.
# Before building it, you need to check the proper LENA image repository &
tag.
FROM docker.io/lenacloud/lena-cluster:1.3.1.10-jdk8-openjdk

# Change or add JDK & packages as your own policy.
# RUN yum update -y
# RUN yum install -y java-1.8.0-openjdk-devel.x86_64

# The service address of LENA manager.
ENV LENA_MANAGER_ADDRESS lena-manager.namespace.svc.cluster.local:7700

# The key to access LENA manager.
ENV LENA_MANAGER_KEY you_manager_key_from_manager_user_admin_menu

# Id of template. Format is <Service Container Name>:<Version>.
ENV LENA_CONFIG_TEMPLATE_ID was-cluster_01:1

# To download and validate your license, LENA_CONTRACT_CODE is required.
# You can acquired it via LENA supplier.
# ENV LENA_CONTRACT_CODE your_own_lena_contract_code

# Download & change template files from manager.
RUN $\{LENA_HOME}/docker-entrypoint.sh download_template

# If you have your own license file, copy it.
#COPY license.xml ${LENA_HOME}/license/

# Or If you uploaded your own container license file to manager, you can
download it from manager
# Caution!. After downloading you need to validate the file with like
'xmllint' command.
# RUN $\{LENA_HOME}/docker-entrypoint.sh download_license

# Copy your application source to deployment path.
# COPY application.war /usr/local/lena/servers/appServer/webapps/
```

위 샘플 Dockerfile의 구조는 다음과 같다.

Table 1. 샘플 Dockerfile의 항목별 설명

구성 순서	설명
상위 Layer Image 지정	<구문>FROM ${상위 Image} 상위 Image도 LENA Image 또는 LENA Image를 상속받은 Image ※ Project에서 선택한Base Image의 Repository/Tag를 입력한다.
Library 추가 설치	<구문> RUN yum install ${추가 Library} OS 별도 패키지 설치 명령어를 활용하여 필요 Library를 추가하고, 패키지 별 별도의 설치 방법이 필요할 경우 해당 패키지의 가이드에 따라 Library를 추가 설치한다. <JDK 재설치> JDK도 Library 추가와 동일한 방식으로 설치할 수 있다. 다만, LENA의 Server들이 JDK를 참조하고 있으므로, 다음의 설정정보도 변경하여야 한다. * Symbolic Link '/usr/lib/jvm/java'을 설치된 JDK 위치에 맞도록 재설정 하여야 한다. 위 정보는 다음 부분에서 사용하고 있다. • 환경변수 ${JAVA_HOME} • JDK 설치위치 : ${LENA_HOME}/etc/info/ java-home.info 값 • Server 환경설정 : ${LENA_HOME}/env.sh의 JAVA_HOME 값 • docker-entrypoint.sh에서 domain cache값 TTL 설정
환경변수 설정	<구문> ENV ${key} ${value} 하위의 설정정보 또는 라이선스를 다운로드 받기 위한 환경변수를 설정한다. 여기에 설정된 값은 Container 기동시점에도 사용될 수 있다. • LENA_MANAGER_ADDRESS : 설치된 Manager의 Service 주소 • LENA_MANAGER_KEY : Open API도 Manager 접근시 필요한 인증도큰 (Manager의 Admin > Users 메뉴에서 확인가능) • LENA_CONFIG_TEMPLATE_ID : 다운로드 할 설정정보 식별자, (Service Cluster 이름 + ":" + Revision 번호) • ENV LENA_CONTRACT_CODE : 라이선스 다운로드를 위한 계약 코드
설정정보 다운로드	<구문> RUN ${LENA_HOME}/docker-entrypoint.sh download_template Manager도 부터 설정정보를 Download 한다. curl / wget 등의 명령으로 Open API를 호출한다. Manager다운로드시 후, 압축파일 해제, Mod/Owner/Group 수정을 실행한다.
라이선스 복사 또는 다운로드	<구문> RUN curl -o ₩${LENA_HOME}/license/license.xml ...<이하생략> Manager도부터 License를 다운로드 받는다. 다운로드 후 XML Validation 및 Owner/Group 수정을 한다. COPY license.xml ${LENA_HOME}/license/ 별도도 받급받은 License가 있을 경우 Container 내부도 복사한다.

구성 순서	설명
Application Artifact 복사	<구문> COPY application.war ${LENA_SERVER_HOME}/webapps/ Application Artifact (예제에서는 application.war) 파일을 Deploy위치에 복사한다. 기본 위치는 아래와 같다. • WAS : ${LENA_SERVER_HOME}/webapps/ • WEB Server : ${LENA_SERVER_HOME}/webapps/htdocs 위 위치는 개별 Server 및 Application 설정에 의해 변경가능하다.

완성된 Dockerfile을 기반으로 Image를 Build할 수 있으며 (docker build), Build 후 Docker Hub 나 ECR과 같은 Registry에 등록 (docker register)하여 Container Platform에서 활용할 수 있도록 한다.

Dockerfile 작성 (일반계정)

root 계정이 아닌 일반계정을 사용하길 원하는경우 Base Image 생성시 추가적인 작업이 필요하다. 아래 샘플은 Centos7 을 사용하며 'lena' 일반계정을 사용하는 케이스다. 계정생성 / sudo 설치순서도 진행된다. Container 기동시 entrypoint script 에서 root 권한 명령을 수행할 수 있도록, 본 과정에서 sudo 권한을 부여하고, password 설정 없이 sudo 명령이 가능하도록 설정한다. entrypoint script 가 종료될때 sudo 사용시 password 입력하도록 변경하여 일반계정의 sudo 명령을 제한한다.

Dockerfile 샘플

```
FROM docker.io/lenasupport/lena-cluster:1.3.1.10-centos7-jdk8-openjdk

# create account
ENV LENA_USER lena
RUN adduser ${LENA_USER}
RUN chown -R ${LENA_USER}:${LENA_USER} ${LENA_HOME}


# sudo auth setting
RUN yum install -y sudo && yum clean all
RUN usermod -aG wheel ${LENA_USER}
## nopasswd sudo on - off this option in docker entry point
RUN sed -i 's/# %wheel/%wheel/g' /etc/sudoers

USER ${LENA_USER}
```

Docker Image 빌드

다음은 Dockerfile로 Image를 Build하는 명령어이다. 사전에 Image의 Repository 및 Tag Rule을 정의하여야 하며, 마지막 인자값에 Dockerfile의 위치를 지정하는 것에 유의한다. (하기 예시는 Current Path에 Dockerfile이 있을 경우에 유효하다.)

Docker Image 빌드 실행 명령어 (예시)

```
$ docker build --no-cache --rm --tag [REPOSITORY[:TAG]] .
```

상기된 명령어에 사용된 옵션은 다음과 같다. 옵션은 Project환경에 맞도록 선별하여 사용한다.

- --no-cache : 이전 빌드에서 생성된 캐시를 사용하지 않음
- --rm : 이미지 생성에 성공했을 때 임시 컨테이너를 삭제

Docker Image 등록

다음은 Image Registry에 Image를 등록하는 명령어이다. 사전에 Registry에 등록 가능한 권한을 가진 사용자와 암호 등 환경설정을 필요로 한다.

Docker Image 등록 실행 명령어 (예시)

```
$ docker push [OPTIONS] [REPOSITORY[:TAG]]
```

기타 상세 Docker 관리 명령어는 다음 공식 Site를 참조한다.

https://docs.docker.com/engine/reference/commandline/docker/

docker commit은 현재 운영중인 Docker Container를 Image로 등록하는 명령어이다.

운영중인 Container를 Image로 등록 (예시)

```
$ docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

Chapter 4. Kubernetes 기반 배포

4.1. 배포 공통사항

4.1.1. 배포 준비

LENA 설치 사전에 Kubernetes 실행환경이 구성되어 있어야 한다. 다음 각 항목들이 준비되어야 할 사항이다.

1. LENA를 구동할 Kubernetes Cluster와 Namespace의 구성
2. kubectl이 설치되고 CLI가 가능한 환경 마련
3. Kubernetes Cluster에 접근이 가능하도록 Kubernetes 설정 (~/.kube/config)의 구성
4. 접근 가능한 Docker Repository (예 : ECR, Docker Hub, 내부 Docker Repository 등)
5. (옵션) Kubernetes Config 파일 : LENA Manager에서 Log 및 Terminal 접속을 위해 Manager에 업로드 필요
6. LENA Container License 및 계약코드

4.1.2. 배포 실행

Kubernetes는 오케스트레이션 도구별로 다양한 배포 방식이 존재하지만, 본 문서에서는 기본 도구인 kubectl을 활용하는 것을 기준으로 설명한다.

Kubernetes는 Resource의 배포 방식, 환경설정정보, 스펙 등을 기술한 YAML형식의 명세 파일을 활용하여 배포/삭제/업데이트를 수행한다. 하기에 설명되는 ${manifest-file.yaml}은 이 명세서를 지칭한다. 본 문서에서는 배포에 필요한 최소한의 명령어 및 옵션만을 기술하므로, 상세한 내용은 공개된 Kubernetes 설명서를 참조한다.

작업 namespace의 설정

하기의 모든 배포관련 작업은 namespace 단위로 이루어지므로, 배포를 하기 앞서 작업 대상이 되는 namespace를 설정하여야 한다.

```
$ kubectl config set-context --current --namespace=${namespace}
```

위의 방식으로 설정하지 않을 시 하기의 모든 실행 명령어 뒤에 '--namespace=${namespace}' 구문을 추가하여야 한다. 다음은 그 예시이다.

```
$ kubectl apply -f ${manifest-file.yaml} --namespace=${namespace}
```

Kubernetes Resource배포 및 업데이트

'kubectl apply' 명령어를 통해 배포를 실행하고, 그 형식은 다음과 같다.

```
$ kubectl apply -f ${manifest-file.yaml}
```

위 명령어는 배포 뿐아니라, 동일 namespace의 동일 name을 가진 Object가 존재할 경우 해당 Resource의

명세를 업데이트 한다.

Kubernetes Resource의 삭제

'kubectl delete' 명령어로 배포된 Resource를 삭제한다.

```
$ kubectl delete -f ${manifest-file.yaml}
```

Workload 업데이트 및 롤백

Application, Web Server의 경우 Application Artifact의 변경 등에 의해 배포된 Workload 업데이트가 필요할 수 있다. Kubernetes에서는 이를 Rolling 방식으로 업데이트하는 방법을 제공한다.

다음은 배포된 Workload를 Rolling방식으로 업데이트 하는 명령어이다.

```
$ kubectl rollout restart ${workloadType}/${workloadName}
```

위 명령어를 통해 Workload가 갱신되며 순차적인 개정(Revision)이 생성된다. Revision 목록을 조회하는 명령어는 다음과 같다.

```
$ kubectl rollout history ${workloadType}/${workloadName}
```

다음은 직전 상태로 롤백하는 명령어 이다.

```
$ kubectl rollout undo ${workloadType}/${workloadName}
```

다음은 입력된 Revision으로 롤백하는 명령어 이다.

```
$ kubectl rollout undo ${workloadType}/${workloadName}
--to-revision=${revisionNo}
```

위 명령어에서 인자값 ${workloadType}은 Workload의 유형으로 그 값은 'statefulset', 'deployment', 'daemonset' 중 하나이다. ${workloadName}은 Workload의 이름이고, ${revisionNo}는 개정(Revision) 값이다.

배포된 Resource의 확인

베포된 Workload, Service등의 Kubernetes Object를 확인하는 명령어 kubectl get이다.

```
$ kubectl get ${resourceType}
```

위 명령어에서 ${resourceType}은 'statefulsets', 'deployments', 'daemonsets', 'services', 'configMaps'와 같은 Kubernetes Resource 유형 중 하나이다.

4.2. Manager 배포

4.2.1. 설정 항목

기본 설정 항목

Manager Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

설정 관련 항목	설정 값 / 설명	비고
Workload 종류	StatefulSet	-
Replica개수	1	-
Container Port	TCP : 7700 UDP : 16100 TCP : 16100	-
Volume Mount	디렉토리 /usr/local/lena/repository를 외부 Volume에 매핑 (아래 예제에서는 persistentVolumeClaim 방식으로 매핑)	-
Probe (Health Check)	HttpGetAction, '/lena' 페이지 호출	-

적용시점 결정 항목 – Workload 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Manifest 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-manager:1.5.0.1-jdk8-openjdk
namespace	Kubernetes내 논리적 그룹 명	default
name	Workload의 이름, 이 값은 Pod 이름 / Hostname의 Prefix로 사용된다.	lena-manager
label	Service와의 연결, 검색에 사용되는 Label도 Key: Value쌍으로 정의된다.	type: sample-lena-manager
imagePullPolicy	이미지 Pull 정책으로 다음 중 하나를 선택한다. • Always : Repository로부터 항상 Pull받음 • IfNotPresent : Local에 Image가 없은 경우 에만 Pull 받음 • Never : Pull 받지 않음	Always

설정 관련 항목	설명	Sample 값
기타 환경변수	환경변수의 설정은 ENV 또는 Config Map으로 설정 가능	(configMap 방식)

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Manager 환경변수 부분을 참조한다.

환경변수	설명	Sample 값
LENA_JVM_HEAP_SIZE	Heap Memory 크기 지정	1024m (기본)
LENA_JVM_METASPACE_SIZE	Metaspace Memory크기	256m (기본)
LENA_MANAGER_DOMAIN_ENABLED	Domain Name 활성화 여부	Y
LENA_MANAGER_ADDRESS	Service의 Domain 주소 : 포트	lena-manager.default.svc.cluster.local:7700
JAVA_DOMAIN_CACHE_TTL	Domain 주소 Cache 시간 (초)	3 (기본)

적용시점 결정 항목 – Service관련

설정 관련 항목	설명	Sample 값
namespace	Kubernetes내 논리적 그룹 명	default
name	Service를 식별하는 이름으로, namespace내에서 유일한 값이 어야 한다. 이 값은 Service Domain주소에 적용된다.	lena-manager
type	외부로 Service를 노출하는 방식으로, 다음 중 하나 이다. • NodePort : k8s가 설치된 모든 Node의 지정 Port로 서비스 • LoadBalancer : 외부 Loadbalancer를 통한 서비스 • ClusterIp : k8s Cluster내 전용으로, 고정된 IP로 서비스	NodePort

4.2.2. Manifest 기반 배포

Workload

Kubernetes에서의 Container는 Pod 단위로 설치되며, Kubernetes Object를 기술한 YAML 파일형식의 Manifest 파일을 작성하여 설치하는 것이 일반적인 방식이다.

다음은 Manager를 설치하기 위한 Manifest 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Manager Workload명세 (Manifest) 파일

```
---
```

```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: lena-manager
spec:
  selector:
    matchLabels:
      type: lena-manager
  serviceName: lena-manager
  replicas: 1
  template:
    metadata:
      labels:
        type: lena-manager
    spec:
      containers:
      - name: lena-manager
        image: docker.io/lenasupport/lena-manager:1.5.0.1-jdk8-openjdk
        imagePullPolicy: Always
        ports:
        - containerPort: 7700
        envFrom:
        - configMapRef:
            name: configmap-lena-manager
        volumeMounts:
        - name: wsy-lena-manager-repository
          mountPath: /usr/local/lena/repository
        readinessProbe:
          httpGet:
            path: /lena
            port: 7700
          initialDelaySeconds: 20
          timeoutSeconds: 1
        livenessProbe:
          httpGet:
            path: /lena
            port: 7700
          initialDelaySeconds: 30
          periodSeconds: 5
      volumes:
        - name: wsy-lena-manager-repository
          persistentVolumeClaim:
            claimName: lena-manager-repository
      terminationGracePeriodSeconds: 0

---
apiVersion: v1
```

```
kind: ConfigMap
metadata:
  name: configmap-lena-manager
data:
  LENA_MANAGER_DOMAIN_ENABLED: "Y"
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-manager-deployment-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-manager-deployment-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get statefulsets

NAME            READY   AGE
lena-manager    1/1     10s
```

Service

다음은 Manager Service를 배포하기 위한 Manifest 파일은 다음과 같다.

LENA Manager Service명세 (Manifest) 파일

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-manager
spec:
  selector:
    type: lena-manager
  ports:
  - name: manager-tcp
    port: 7700
    targetPort: 7700
    nodePort: 31848
    protocol: TCP
  - name: monitoring-tcp
    port: 16100
    targetPort: 16100
    protocol: TCP
  - name: monitoring-udp
    port: 16100
    targetPort: 16100
    protocol: UDP
  type: NodePort
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-manager-service-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-manager-service-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get services
NNAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-manager NodePort 10.43.xx.xx <none> 7700:30770/TCP,16100:31610/UDP
10s
```

4.2.3. Manager 접속

Service 유형을 NodePort로 설정하였다면 http://[Node IP]: [Node Port] / 에 접속하여 Manager에 접속한다.

4.3. Session Server 배포

4.3.1. 배포 준비

배포 전에 Manager에 Session Server를 등록하기 위해서는 Manager에서 'Service Cluster'를 등록하여야 한다. Manager의 'Cluster > Session Cluster' 메뉴 위치에서 Session Server 유형의 신규 Service Cluster를 생성한다.

Service Cluster생성 및 관리 관련 상세한 내용은 별도제공 문서인 '운영자 매뉴얼'을 참조한다.

4.3.2. 설정 항목

기본 설정 항목

Session Server Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

설정 관련 항목	설정 값 / 설명	비고
Workload 종류	StatefulSet	-
Replica개수	2 (Primary, Secondary Server 2개)	-
Container Port	TCP : 5180	-
Probe	ExecAction, ${LENA_SERVER_HOME}/health.sh 호출	-

적용시점 결정 항목 – Workload 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Manifest 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-session:1.5.0.1-jdk8-openjdk
namespace	Kubernetes내 논리적 그룹 명	default
name	Workload의 이름, 이 값은 Pod 이름 / Hostname의 Prefix로 사용된다.	lena- session
label	Service와의 연결, 검색에 사용되는 Label도 Key: Value쌍으로도 정의된다.	type: lena-session
imagePullPolicy	이미지 Pull 정책	Always
기타 환경변수	환경변수의 설정은 ENV 또는 Config Map으로도 설정 가능	(configMap 방식)

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Session 환경변수 부분을 참조한다.

환경변수	설명	Sample 값
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m (기본)
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	SESSION-001
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트	sample-lena-manager.default.svc.cluster.local:7700
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	0 (기본)
LENA_SESSION_0_ADDRESS	• Primary Session 서버 주소, StatefulSet설정과 일치되어야 함	lena-session-0.default.svc.cluster.local
LENA_SESSION_1_ADDRESS	• Secondary Session 서버 주소, StatefulSet설정과 일치되어야 함	lena-session-1.default.svc.cluster.local
LENA_SESSION_EXPIRE_SEC	• Session 만료 시간 (초)	1800 (기본)
LENA_CONFIG_SHARE_SESSION	• Application 간 Session 공유여부	N

적용시점 결정 항목 – Service관련

설정 관련 항목	설명	Sample 값
namespace	Kubernetes내 논리적 그룹 명	default
name	Service를 식별하는 이름으로, namespace내에서 유일한 값이어야 한다. 이 값은 Service Domain주소에 적용된다.	lena-session
type	외부로 Service를 노출하는 방식으로, Session Server에는 지정하지 않는다.	

4.3.3. Manifest 기반 배포

Workload

Kubernetes에서의 Container는 Pod 단위로 설치되며, Kubernetes Object를 기술한 YAML 파일형식의 Manifest 파일을 작성하여 설치하는 것이 일반적인 방식이다.

다음은 Manager를 설치하기 위한 Manifest 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Session Workload명세 (Manifest) 파일 예시

```
---
apiVersion: apps/v1
kind: StatefulSet
```

```
metadata:
  name: lena-session
spec:
  selector:
    matchLabels:
      type: lena-session
  serviceName: lena-session
  replicas: 2
  template:
    metadata:
      labels:
        type: lena-session
    spec:
      containers:
      - name: lena-session
        image: docker.io/lenasupport/lena-session:1.5.0.1-jdk8-openjdk
        imagePullPolicy: Always
        ports:
        - containerPort: 5180
        envFrom:
        - configMapRef:
            name: configmap-lena-session
        tty: true
        readinessProbe:
          exec:
            command:
            - /usr/local/lena/servers/sessionServer/health.sh
          initialDelaySeconds: 20
          periodSeconds: 5
        livenessProbe:
          exec:
            command:
            - /usr/local/lena/servers/sessionServer/health.sh
          initialDelaySeconds: 30
          periodSeconds: 5
      terminationGracePeriodSeconds: 0
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-lena-session
data:
  LENA_SESSION_0_ADDRESS: "lena-session-0.lena-
session.default.svc.cluster.local:5180"
  LENA_SESSION_1_ADDRESS: "lena-session-1.lena-
session.default.svc.cluster.local:5180"
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
  LENA_SESSION_EXPIRE_SEC: "1800"
```

```
  LENA_CONFIG_TEMPLATE_ID: "SESSION-001"
  LENA_CONFIG_SHARE_SESSION: "N"
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-session-deployment-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-session-deployment-sample.yaml
```

- 배포결과 확인>
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get statefulsets
NAME READY AGE
lena-manager 1/1 30m
lena-session 2/2 10s
```

Service

다음은 Session Service를 배포하기 위한 Manifest 파일은 다음과 같다.

LENA Session Service명세 (Manifest) 파일 예시*

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-session
spec:
  selector:
    type: lena-session
  clusterIP: None
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-session-service-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-session-service-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-session ClusterIP None <none> <none> 10s
...
```

4.3.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **ServerList버튼** 을 클릭하면 하단에 2개의 Session Server가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다.

4.4. WAS 배포

4.4.1. 배포 준비

배포 전에 Manager에 WAS Server를 등록하기 위해서는 Manager에서 'Service Cluster'를 등록하여야 한다. Manager의 'Cluster > Server Cluster' 메뉴 위치에서 WAS(Enterprise/SE) 유형의 신규 Service Cluster를 생성한다.

Service Cluster를 생성한 후 Overview 탭에서 Service Endpoint를 설정한다. Kubernetes의 경우 "http://${Service명}.${namespace명}.svc.cluster.localhost:${Service 포트}" 형식이 된다. 이 값은 Web Server의 VirtualHost의 Proxy설정에서 활용된다.

생성된 Service Cluster의 Template 탭에서 WAS의 설정을 수행한다. 설정정보 저장 후 Template에 대한 Revision을 생성한다. Service Cluster생성 및 관리 관련 상세한 내용은 별도제공 문서인 '운영자 매뉴얼'을 참조한다.

4.4.2. 설정 항목

기본 설정 항목

WAS Container는 다음과 같은 권고 설정을 기준으로 배포 된다..

설정 관련 항목	설정 값 / 설명	비고
Workload 종듀	Deployment	
Container Port	TCP : 8180	

적용시점 결정 항목 – Workload 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 핟 설정 요소와 이후 설명되는 Manifest 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-cluster:1.5.0.1-jdk8-openjdk

설정 관련 항목	설명	Sample 값
namespace	Kubernetes내 논리적 그룹 명	default
name	Workload의 이름, 이 값은 Pod 이름 / Hostname의 Prefix도 사용된다.	lena-was
label	Service와의 연결, 검색에 사용되는 Label도 Key: Value쌍으도 정의된다.	type: lena-was
strategy:type	Deployment Update 정책, RollingUpdate, Recreate중 선댁한 Update정책명은 기술	RollingUpdate
imagePullPolicy	이미지 Pull 정책	Always
replica 개수	Container (Pod) 개수	2
Probe	HttpGetAction, 체크 Page, 시작 Delay 시간, 주기는 Application 득성에 맞도독 설정 필요	'/' 호춛
기타 환경변수	환경변수의 설정은 ENV 또는 Config Map으도 설정 가능	(configMap 방식)

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 WAS 환경변수 부분은 참조한다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WAS-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.default.svc.cluster.local:7700
LENA_MANAGER_KEY	• LENA_MANAGER_KEY : Open API도 Manager 접근시 필요한 인증도큰 • Manager의 Admin > Users 메뉴에서 확인 가능	(개볃 Manager에서 확인 입뎍 필요)
LENA_CONFIG_TEMPLATE_DOWNLOAD	• 설정 정보 다운도드 여부	Y
LENA_CONTRACT_CODE	• 라이선스 다운도드릍 위한 계약 코드 • 라이선스 받급 시 제공된 코드 값	(개볃 코드 확인 필요)
LENA_LICENSE_DOWNLOAD_URL		manager
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	3 (기본)

적용시점 결정 항목 – Service관련

설정 관련 항목	설명	Sample 값
namespace	Kubernetes내 논리적 그룹 명	default
name	Service를 식별하는 이름으로, namespace내에서 유일한 값이어야 한다. 이 값은 Service Domain주소에 적용된다.	lena-was
type	외부로 Service를 노출하는 방식, 외부로 노출할 요건이 없으면 별도 설정이 불필요	

4.4.3. Manifest 기반 배포

Workload

Kubernetes에서의 Container는 Pod 단위로 설치되며, Kubernetes Object를 기술한 YAML 파일형식의 Manifest 파일을 작성하여 설치하는 것이 일반적인 방식이다.

다음은 Manager를 설치하기 위한 Manifest 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Session Workload명세 (Manifest) 파일 예시*

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: lena-was
spec:
  selector:
    matchLabels:
      type: lena-was
  replicas: 2
  strategy:
    type: RollingUpdate
  minReadySeconds: 10
  revisionHistoryLimit: 1
  template:
    metadata:
      labels:
        type: lena-was
    spec:
      containers:
      - name: lena-was
        image: docker.io/lenasupport/lena-cluster:1.5.0.1-jdk8-openjdk
        imagePullPolicy: Always
        ports:
        - containerPort: 8180
        envFrom:
        - configMapRef:
            name: configmap-lena-was
        readinessProbe:
```

```
          httpGet:
            path: /
            port: 8180
          initialDelaySeconds: 10
          periodSeconds: 5
        livenessProbe:
          httpGet:
            path: /
            port: 8180
          initialDelaySeconds: 15
          periodSeconds: 5
      volumes:
      terminationGracePeriodSeconds: 0

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-lena-was
data:
  LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
  LENA_CONFIG_TEMPLATE_ID: "WAS-001"
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
  LENA_MANAGER_KEY:
"aSw7RMPSw15LeN%2FMZnrxzjgV0BzZe18iVHZbJ8CkdLlea2Ecd8AleK9oPCLXuw%3D%3D"
  LENA_LICENSE_DOWNLOAD_URL: "manager"
  LENA_CONTRACT_CODE: "pghzJJqTdzaGtTuASr8yfw=="
JAVA_DOMAIN_CACHE_TTL: "3"
```

- 배포실행>
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-was-deployment-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-was-deployment-sample.yaml
```

- 배포결과 확인>
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get deployments
NAME READY AGE
lena-was 2/2 10s
```

Service

다음은 Application Service를 배포하기 위한 Manifest 파일은 다음과 같다.

LENA Session Service명세 (Manifest) 파일 예시

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-was
spec:
  selector:
    type: lena-was
ports:
  - port: 8180
    targetPort: 8180
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-was-service-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-was-service-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-was ClusterIP 10.43.xx.xx <none> 8180/TCP 20s
```

4.4.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **Server List 버튼** 을 클릭하면 하단에 2개의 WAS가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다.

4.5. Embedded WAS 배포

4.5.1. 배포 준비

배포 전에 Manager에 Embedded WAS Server를 등록하기 위해서는 Manager에서 'Service Cluster'를 등록하여야 한다. Manager의 'Cluster > Server Cluster' 메뉴 위치에서 WAS(Embedded) 유형의 신규 Service Cluster를 생성한다.

4.5.2. 설정 항목

기본 설정 항목

WAS Container는 다음과 같은 권고 설정을 기준으로 배포 된다..

설정 관련 항목	설정 값 / 설명	비고
Workload 종듀	Deployment	
Container Port	TCP : 8180	

적용시점 결정 항목 – Workload 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 핟 설정 요소와 이후 설명되는 Manifest 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-embedded:1.5.0.1-jdk8-openjdk
namespace	Kubernetes내 논리적 그룹 명	default
name	Workload의 이름, 이 값은 Pod 이름 / Hostname의 Prefix로 사용된다.	lena-was
label	Service와의 연결, 검색에 사용되는 Label로 Key: Value쌍으로 정의된다.	type: lena-was
strategy:type	Deployment Update 정책, RollingUpdate, Recreate중 선댁한 Update정책명은 기술	RollingUpdate
imagePullPolicy	이미지 Pull 정책	Always
replica 개수	Container (Pod) 개수	2
Probe	HttpGetAction, 체크 Page, 시작 Delay 시간, 주기는 Application 득성에 맞도독 설정 필요	'/' 호춛
기타 환경변수	환경변수의 설정은 ENV 또는 Config Map으로 설정 가능	(configMap 방식)

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Embedded WAS 환경변수 부분은 참조한다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WAS-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.default.svc.cluster.local:7700
LENA_MANAGER_MONITORING_PORT	• Manager 모니터딩 Port 정보	16100

환경변수	설명	Sample 값
LENA_APP_FILE	• Application Jar 파일 명	sample-app.jar
LENA_APP_DIR	• Application Jar 디렉토리 명	/usr/local/lena

적용시점 결정 항목 – Service관련

설정 관련 항목	설명	Sample 값
namespace	Kubernetes내 논리적 그룹 명	default
name	Service를 식별하는 이름으로, namespace내에서 유일한 값이어야 한다. 이 값은 Service Domain주소에 적용된다.	lena-was
type	외부로 Service를 노출하는 방식, 외부로 노출할 요건이 없으면 별도 설정이 불필요	

4.5.3. Manifest 기반 배포

Workload

Kubernetes에서의 Container는 Pod 단위로 설치되며, Kubernetes Object를 기술한 YAML 파일형식의 Manifest 파일을 작성하여 설치하는 것이 일반적인 방식이다.

다음은 Manager를 설치하기 위한 Manifest 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Embedded WAS Workload명세 (Manifest) 파일 예시*

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: lena-was
spec:
  selector:
    matchLabels:
      type: lena-was
  replicas: 2
  strategy:
    type: RollingUpdate
  minReadySeconds: 10
  revisionHistoryLimit: 1
  template:
    metadata:
      labels:
        type: lena-was
    spec:
      containers:
```

```
      - name: lena-was
        image: docker.io/lenasupport/lena-embedded:1.5.0.1-jdk8-openjdk
        imagePullPolicy: Always
        ports:
        - containerPort: 8180
        envFrom:
        - configMapRef:
            name: configmap-lena-was
        readinessProbe:
          httpGet:
            path: /
            port: 8180
          initialDelaySeconds: 10
          periodSeconds: 5
        livenessProbe:
          httpGet:
            path: /
            port: 8180
          initialDelaySeconds: 15
          periodSeconds: 5
      volumes:
      terminationGracePeriodSeconds: 0

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-lena-was
data:
  LENA_CONFIG_TEMPLATE_ID: "WAS-001"
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
  LENA_APP_FILE: "sample-app.jar"
```

- 배포실행>
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-was-deployment-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-was-deployment-sample.yaml
```

- 배포결과 확인>
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get deployments
NAME READY AGE
lena-was 2/2 10s
```

Service

다음은 Application Service를 배포하기 위한 Manifest 파일은 다음과 같다.

LENA Session Service명세 (Manifest) 파일 예시

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-was
spec:
  selector:
    type: lena-was
ports:
  - port: 8180
    targetPort: 8180
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-was-service-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-was-service-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-was ClusterIP 10.43.xx.xx <none> 8180/TCP 20s
```

4.5.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **Server List 버튼** 을 클릭하면 하단에 2개의 WAS가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다.

4.6. Web Server 배포

4.6.1. 배포 준비

배포 전에 Manager에 Web Server를 등록하기 위해서는 Manager에서 'Service Cluster'를 등록하여야 한다. Manager의 'Cluster > Server Cluster' 메뉴 위치에서 WEB server 유형의 신규 Service Cluster를 생성한다.

생성된 Service Cluster의 Template 탭에서 Web Server설정을 한다. 설정 내용 중 앞서 생성된 WAS와 Web Server 연계를 위해서는 Virtual Host 탭에서 Proxy 설정이 필요하다.

설정은 저장 후Template에 대한 Revision은 생성한다. Revision이 생성되어야 저장된 설정의 Download가 가능해진다.

Service Cluster생성 및 관리 관련 상세한 내용은 별도제공 문서인 '운영자 매뉴얼'은 참조한다.

4.6.2. 설정 항목

기본 설정 항목

Web Server Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

설정 관련 항목	설정 값 / 설명	비고
Workload 종듀	Deployment	
Container Port	TCP : 5180	변경붇가

적용시점 결정 항목 – Workload 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 핟 설정 요소와 이후 설명되는 Manifest 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-web:1.5.0.1-jdk8-openjdk
namespace	Kubernetes내 논리적 그룹 명	default
name	Workload의 이름, 이 값은 Pod 이름 / Hostname의 Prefix도 사용된다.	lena-web
label	Service와의 연결, 검색에 사용되는 Label도 Key: Value쌍으로 정의된다.	type: lena-web
strategy:type	Deployment Update 정책	RollingUpdate
imagePullPolicy	이미지 Pull 정책	Always
replica 개수	Container (Pod) 개수	2
Probe (Health Check)	HttpGetAction 방식, 체크 Page, 시작 Delay시간, 주기는 Application 득성에 맞게 설정 필요	'/' 호춛, 5초 주기, 15초/20초 Delay
기타 환경변수	환경변수의 설정은 ENV 또는 Config Map으로 설정 가능	(configMap 방식)

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Web Server 환경변수 부분은 참조한다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WEB-001:1

환경변수	설명	Sample 값
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.default.svc.cluster.local:7700
LENA_MANAGER_KEY	• LENA_MANAGER_KEY : Open API로 Manager 접근시 필요한 인증도큰 • Manager의 Admin > Users 메뉴에서 확인가능	(개별 Manager에서 확인 입력 필요)
LENA_CONFIG_TEMPLATE_DOWNLOAD	• 설정 정보 다운로드 여부	Y
LENA_CONTRACT_CODE	• 라이선스 다운로드를 위한 계약 코드 • 라이선스 발급 시 제공된 코드 값	(개별 코드 확인 필요)
LENA_LICENSE_DOWNLOAD_URL	• 라이선스 다운로드 위치	manager
LENA_RUN_AGENT	• Agent 실행여부	Y

적용시점 결정 항목 – Service관련

설정 관련 항목	설명	Sample 값
namespace	Kubernetes내 논리적 그룹 명	default
name	Service를 식별하는 이름으로, namespace내에서 유일한 값이어야 한다. 이 값은 Service Domain주소에 적용된다.	lena-web
type	외부로 Service를 노출하는 방식, 외부로 노출할 요건이 없으면 별도 설정이 불필요	NodePort (port 31180)

4.6.3. Manifest 기반 배포

Workload

Kubernetes에서의 Container는 Pod 단위로 설치되며, Kubernetes Object를 기술한 YAML 파일형식의 Manifest 파일을 작성하여 설치하는 것이 일반적인 방식이다.

다음은 Manager를 설치하기 위한 Manifest 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Web Workload명세 (Manifest) 파일 예시

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
  name: lena-web
spec:
  selector:
    matchLabels:
      type: lena-web
  replicas: 1
  strategy:
    type: RollingUpdate
  minReadySeconds: 10
  revisionHistoryLimit: 1
  template:
    metadata:
      labels:
        type: lena-web
    spec:
      containers:
      - name: lena-web
        image: docker.io/lenasupport/lena-web:1.5.0.1-jdk8-openjdk
        imagePullPolicy: Always
        ports:
        - containerPort: 7180
        readinessProbe:
          httpGet:
            path: /
            port: 7180
          initialDelaySeconds: 5
          periodSeconds: 5
        livenessProbe:
          httpGet:
            path: /
            port: 7180
          initialDelaySeconds: 10
          periodSeconds: 10
        envFrom:
        - configMapRef:
            name: configmap-lena-web
      terminationGracePeriodSeconds: 0

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap-lena-web
data:
  LENA_MANAGER_ADDRESS: "lena-manager.default.svc.cluster.local:7700"
  LENA_AGENT_RUN: "Y"
  LENA_CONFIG_TEMPLATE_ID: "WEB-001:1"
  LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
```

```
  LENA_MANAGER_KEY:
"aSw7RMPSw15LeN%2FMZnrxzjgV0BzZe18iVHZbJ8CkdLlea2Ecd8AleK9oPCLXuw%3D%3D"
  LENA_LICENSE_DOWNLOAD_URL: "manager"
  LENA_CONTRACT_CODE: "dX89RRxPk6/PBPqbUuYm7w=="
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-web-deployment-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-web-deployment-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get deployments
NAME READY AGE
lena-web 2/2 10m
lena-was 2/2 10s
```

Service

다음은 Application Service를 배포하기 위한 Manifest 파일은 다음과 같다.

LENA Session Service명세 (Manifest) 파일 예시

```
---
apiVersion: v1
kind: Service
metadata:
  name: lena-web
spec:
  selector:
    type: lena-web
  ports:
  - nodePort: 31180
    port: 7180
    targetPort: 7180
  type: NodePort
```

- 배포실행
  배포는 kubectl apply 명령으로 실행한다. 파일명이 lena-web-service-sample.yaml이라면 배포 명령은 다음과 같다.

```
$ kubectl apply -f lena-web-service-sample.yaml
```

- 배포결과 확인
  배포된 Workload는 kubectl get 명령어 실행을 통해 확인할 수 있다.

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
...
lena-web NodePort 10.43.xx.xx <none> 7180:31180/TCP 13h
...
```

4.6.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **Server List 버튼** 을 클릭하면 하단에 2개의 Web Server가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다

Chapter 5. Docker Swarm 기반 배포

5.1. 배포 공통사항

5.1.1. 배포 준비

LENA 설치 사전에 Docker Swarm 실행환경이 구성되어 있어야 한다. 다음 각 항목들이 준비되어야 할 사항이다.

1. Docker가 설치되고 구동 가능한 환경 마련
2. LENA를 구동할 Docker Swarm Overlay Network 구성
3. 접근 가능한 Docker Repository (예 : ECR, Docker Hub, 내부 Docker Repository 등)
4. LENA Container License 및 계약코드

Swarm Cluster의 구성

Swarm Cluster는 복수개의 물리적 서버(Node)에 Container를 배포할수 있는 Orchestration Runtime환경으로, Manager Node와 Worker Node로 구성된다. Manager Node를 생성하는 명령어는 docker swarm init 으로 --advertise-addr 옵션에 Manager Node의 주소 (최초 Manager의 경우에는 자신의 주소)를 입력하여 실행한다.

Docker Swarm Init (Manager Node 생성)

```
$ docker swarm init --advertise-addr $(hostname -i)

Swarm initialized: current node (y8ul9r3jq0rgt9k3vbvrayeyg) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-... 192.168.0.10:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and
follow the instructions
```

Manager Node 생성시에 출력되는 docker swarm join 명령구문을 실행하여, Worker Node를 등록한다.

Docker Swarm Worker Node 생성

```
$ docker swarm join --token SWMTKN-1-... 192.168.0.10:2377
```

Manager Node에서 docker node ls 명령을 실행하면, Cluster에 속한 Node를 확인할 수 있다.

Docker Swarm Cluster Node 조회

```
$ docker node ls
ID                            HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS
ENGINE VERSION
887c4dglmszt1afbgnucnokg   SERVER02  Ready   Active
19.03.6-ce
qp9zh3zj5jlzpqhglf11v69y * SERVER01  Ready   Active        Leader
19.03.6-ce
```

Overlay Network 구성

Docker Service들이 운영될 수 있는 Container Network를 생성한다. Docker Swarm의 Overlay Network를 구성은 이후 Service 구성과 동합될 수 있도록 Stack을 구성하고, Stack에 Overlay Network을 생성한다.

Docker Swarm Network 명세 파일 Example (lena-net.yaml)

```yaml
version: "3.8"

networks:
  lena-net:
    driver: overlay
    attachable: true
```

Stack내에 Network을 생성하는 명령어는 명세파일이 lena-net.yaml이고 Stack 명이 'example'이라면 다음과 같다.

```
docker stack deploy -c lena-net.yaml exmaple
```

구성된 Docker Network을 조회하기 위해서는 docker network ls [OPTIONS] 명령어를 사용한다.

Docker Network 조회

```
$ docker network ls

NETWORK ID          NAME                DRIVER              SCOPE
x9loi84l78zu        example_lena-net    overlay             swarm
jhdd0mz8gtrh        ingress             overlay             swarm
.... (출력 생략)
```

5.1.2. 배포 실행

배포실행은 Docker의 기본 CLI 도구인 docker 명령어를 활용하는 것을 기준으로 설명한다.

Docker는 Service의 배포 방식, 환경설정정보 등을 기술한 YAML형식의 명세 파일인 Docker Compose 파일을 활용하여 배포/삭제/업데이트를 수행한다. 하기에 설명되는 ${docker-compose.yaml}은 이 명세서를 지칭한다. 본 문서에서는 배포에 필요한 최소한의 명령어 및 옵션만을 기술하므로, 상세한 내용은 공개된 Docker 설명서를 참조한다.

Docker Service 배포

Docker는 Load-balancing를 포함하는 Network 속성과 환경변수, Deployment 속성은 모두 포함한 Container군은 'Service' 단위도 관리한다. ‘docker create service’ 명령어를 통해 최초 배포를 실행하고, 서비스를 생성할 때 --replicas 옵션은 사용해서 생성할 컨테이너 개수를 지정하고, --publish 옵션은 사용하여 서비스 Port를 지정할 수 있다.

CLI기반 Docker Service 배포

```
$ docker service create --name lena-sample \
    --publish published=8180:8180/tcp \
    --replicas 2 \
    lenacloud/lena-cluster:1.5.0.1-centos7-jdk8-openjdk
```

위 명령어는 배포 뿐아니라, 동일 namespace의 동일 name을 가진 Object가 존재할 경우 해당 Resource의 명세를 업데이트 한다.

Docker Service 갱신

‘docker service scale’ 명령어도 운영 중에 리플리케이션 개수를 변경할 수 있다.

```
$ docker service scale lena-sample=3
```

‘docker service update’ 명령어도 배포된 Service를 갱신할 수 있다.

Docker Update 명령어 예시

```
$ docker service update \
--update-parallelism 1 \
--update-delay 10s \
--update-order start-first \
--image lenacloud/lena-cluster:1.5.0.1-centos7-jdk8-openjdk \
lena-sample
```

Service 갱신 관련 주요 옵션은 옵션은 다음과 같다.

- --update-parallelism : 동시에 업데이트할 컨테이너 개수를 지정한다.
- --update-delay : 업데이트 간 간격을 지정한다.
- --update-order : start-first면 새 컨테이너를 먼저 생성한 뒤에 기존 컨테이너를 삭제한다. stop-first면 기존 컨테이너를 먼저 삭제하고 그 다음에 새 컨테이너를 생성한다.
- --update-failure-action : 업데이트에 실패할 경우 이 값이 pause면 업데이트를 멈추고, continue면 업데이트를 계속하고, rollback이면 업데이트를 롤백한다.
- --update-max-failure-ratio : 실패 비율이 지정한 값 이상이면 업데이트 실패도 간주한다.

Docker Service Rollback

Docker Service 갱신(Update) 후 이전 상태도 복구(Rollback)하기 위해서는 'docker service rollback 명령어’를 사용할 수 있다.

```
$ docker service rollback [OPTIONS] SERVICE명
```

Docker Service 삭제

'docker service rm' 명령어도 배포된 Service를 삭제한다.

```
$ docker service rm [Service명...]
```

Docker Service 조회

'docker service ls' 명령어도 배포된 Service들의 목록을 조회할 수 있다.

```
$ docker service ls
```

'docker service ps 서비스명' 명령어도 배포된 Service의 Container 상태를 조회할 수 있다.

```
$ docker service ps [OPTIONS] [Service명...]
```

Compose 파일 (Service 명세서) 의 활용

실제 운영시에는 개별적인 Command도 배포를 수행하는 것보다, Service구성에 대한 명세서(Compose파일) 를 활용하는 것이 효율적이다. 다음에서는 샘플 Compose 파일을 통해 구조를 살펴본다.

```yaml
version: "3.8"

services:
  lena-was-cluster:
    image: lenacloud/lena-cluster:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - was-cluster.example.local
    ports:
    - target: 8180
      published: 8180
      protocol: tcp
      mode: ingress
    environment:
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_MANAGER_MONITORING_PORT: "16100"
      LENA_MANAGER_KEY: "C9PYwcu%3D%3D"
      LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
      LENA_CONFIG_TEMPLATE_ID: "was-cluster"
      LENA_LICENSE_DOWNLOAD_URL: "manager"
      LENA_CONTRACT_CODE: "vvW7ebNFVjDdtasuvF1utQ=="
      JAVA_DOMAIN_CACHE_TTL: "3"
    deploy:
      mode: replicated
      replicas: 3
    healthcheck:
      test: "curl -f http://localhost:8180/index.jsp"
      interval: 10s
      timeout: 5s
      retries: 3
      start_period: 20s
```

위 파일의 각 Field가 표현하는 내용은 다음과 같다.

- services : Service명으로 구분되는 Service 정의를 포함
- networks : Service가 운영되는 Container Network를 지정
- networks > ${network명} > aliases : Service의 대표 Network 주소
- ports : Service 노출 Port 및 해당 Port와 Container간의 Mapping 정보, L/B 방식 정의
- environments : Container 기동에 필요한 환경변수 정보
- deploy : Container 배포 방식 정의
- healthcheck : Healthcheck 방식 정의

Stack의 구성

Stack은 Kubernetes의 namespace와 유사한 개념으로, 분산된 Application을 통합배포/관리하기 위한 논리적 공간이다. 하기의 모든 배포관련 작업은 Stack 단위로 수행된다. Stack을 별도로 생성하는 과정을

거치지 않으며, 관련 ${docker-compose.yaml} 파일을 묶어서 하나의 Stack을 구성한다. 명령어 구분은 docker stack deploy [-c Compose파일 Path...] [OPTIONS] 이고, 아래는 Service를 구성하는 복수개의 compose 파일을 단일 Stack으로 구성한 샘플이다.

Docker stack 생성

```
docker stack deploy \
       -c lena-net.yaml \
       -c lena-manager.yaml \
       -c lena-session-cluster.yaml \
       -c lena-was-cluster.yaml \
       -c lena-web-cluster.yaml \
       example
```

Docker는 Stack에 포함된 모든 Service, Network를 일괄삭제하는 기능을 제공한다. 명령어 구문은 docker stack rm [OPTIONS] [Stack명...]' 이다.

Docker stack 삭제

```
docker stack rm example
```

5.2. Manager 배포

5.2.1. 설정 항목

기본 설정 항목

Manager Service는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

설정 관련 항목	설정 값 / 설명	비고
N/W 유형	Overlay Network - Host	-
End Point 유형	dnsrr (DNS Round Robin)	-
Replica개수	1	-
Container Port	TCP : 7700, 16100 및 UDP : 16100	-
Volume Mount	디렉토리 /usr/local/lena/repository를 외부 Volume에 매핑 (아래 예제에서는 nfs 방식으로 매핑)	-
Probe (Health Check)	HttpGetAction, '/lena' 페이지 호출	-

적용시점 결정 항목 – Service관련

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-manager.1.5.0.1-centos7-jdk8-openjdk

설정 관련 항목	설명	Sample 값
Stack 명	복수개의 응용 서비스를 통합관리하기 위한 Swarm Cluster의 논리적 그룹인 stack의 이름	example
Service 명	Service를 식별하는 이름으로, Stack내에서 유일한 값이 어야 한다.	lena-manager
network alias	Docker N/W내에서의 식별자(주소)	lena-manager.example.local

설정 가능한 환경변수는 다음과 같다.

환경변수	설명	Sample 값
LENA_JVM_HEAP_SIZE	Heap Memory 크기 지정 (Template 다운도드시 Template 값으로 교체됨)	1024m (기본)
LENA_JVM_METASPACE_SIZE	Metaspace Memory크기 (Template 다운도드시 Template 값으로 교체됨)	256m (기본)
LENA_MANAGER_DOMAIN_ENABLED	Domain Name 환성화 여부	Y
LENA_MANAGER_ADDRESS	Service의 Domain 주소 : 포트	lena-manager.example.local:7700
JAVA_DOMAIN_CACHE_TTL	Domain 주소 Cache 시간 (초)	3 (기본)

5.2.2. 명세서 (Compose 파일) 기반 배포

Docker Swarm에서 Container는 Service 단위로 배포되며, Compose 파일에 Service 명세를 작성하여 배포하는 것이 일반적이다.

다음은 Manager를 설치하기 위한 Compose 파일 샘플이다. 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Manager Service 명세 (lena-manager.yaml) 파일 샘플

```
version: "3.8"
services:
  lena-manager:
    image: lenacloud/lena-manager:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - "lena-manager.example.local"
    ports:
    - target: 7700
      published: 7700
      protocol: tcp
      mode: host
    environment:
      LENA_MANAGER_DOMAIN_ENABLED: "Y"
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_MANAGER_MONITORING_PORT: "16100"
    deploy:
      mode: replicated
      replicas: 1
      endpoint_mode: dnsrr
    volumes:
    - lena-manager-repository:/usr/local/lena/repository
    healthcheck:
      test: "curl -f http://localhost:7700/lena"
      interval: 10s
      timeout: 3s
      retries: 3
      start_period: 30s
volumes:
  lena-manager-repository:
    driver: local
    driver_opts:
      type: "nfs"
      o: "nfsvers=4.1,addr=${FILE_STORAGE_ADDRESS}"
      device: "${FILE_STORAGE_ADDRESS}:/example/lena-manager-repository"
```

- 배포실행
  docker-compose를 활용하여 개별 명세파일별로 배포를 실행 할 수 있으나, 본 문서에서는 Service를 Stack으로 관리하는 방식으로 배포방법을 설명한다. 'docker stack deploy' 명령으로

```
$ docker stack deploy -c lena-manager.yaml example
```

- 배포결과 확인
  배포된 Service는 docker service ls 명령어 실행을 통해 확인할 수 있다.

```
$ docker service ls
ID                  NAME                    MODE                REPLICAS
IMAGE                                                   PORTS
grhr01h2bgqc        example_lena-manager    replicated          0/1
lenacloud/lena-manager:1.5.0.1-centos7-jdk8-openjdk
```

5.2.3. Manager 접속

Service가 정상기동 되면 Browser에서 http://[Node IP]: [Service Port] /lena 주소로 Manager에 접속한다.

5.3. Session Server 배포

5.3.1. 배포 준비

본문서 Session Server 배포 준비 부분 설명은 참조한다.

5.3.2. 설정 항목

기본 설정 항목

Session Server Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

설정 관련 항목	설정 값 / 설명	비고
N/W 유형	Overlay Network - Host	-
End Point 유형	dnsrr (DNS Round Robin)	-
Replica개수	Replica 1 * 2 Set (Primary, Secondary 개별 배포)	-
Container Port	TCP : 5180	-
Health Check	${LENA_SERVER_HOME}/health.sh 호출	-

적용시점 결정 항목 – Service 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Docker Compose 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Stack 명	복수개의 응용 서비스를 통합관리하기 위한 Swarm Cluster의 논리적 그룹인 stack의 이름	example
Service 명	Service를 식별하는 이름으로, Stack내에서 유일한 값이 어야 한다.	• lena-session-0 (Primary 서버) • lena-session-1 (Secondary 서버)

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Session Server Image	lenacloud/lena-session:1.5.0.1-centos7-jdk8-openjdk
Network alias	Docker N/W내에서의 식별자(주소)	• session-0.example.local (Primary 서버) • session-1.example.local (Secondary 서버)

설정 가능한 환경변수는 다음과 같다.

환경변수	설명	Sample 값
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m (기본)
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	SESSION-001
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트	sample-lena-manager.example.local:7700
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	0 (기본)
LENA_SESSION_0_ADDRESS	• Primary Session 서버 주소, StatefulSet설정과 일치되어야 함	session-0.example.local:5180
LENA_SECONDARY_SESSION_NO	• Session Server가 상호 연결하는 Session 서버 번호	• Primary Session Server는 1 • Secondary Session Server는 0
LENA_SESSION_1_ADDRESS	• Secondary Session 서버 주소, StatefulSet설정과 일치되어야 함	session-1.example.local:5180
LENA_SESSION_EXPIRE_SEC	• Session 만료 시간 (초)	1800 (기본)
LENA_CONFIG_SHARE_SESSION	• Application 간 Session 공유여부	N

5.3.3. Docker Compose 기반 배포

Docker Swarm에서 Container는 Service 단위로 설치되며, Service를 기술한 YAML 파일형식의 Docker Compose 파일을 작성하여 설치하는 것이 일반적인 방식이다.

다음은 Manager를 설치하기 위한 Docker Compose 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의

결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Session Cluster Service명세(lena-session-cluster.yaml) 파일 예시

```
version: "3.8"

services:
  lena-session-0:
    image: lenacloud/lena-session:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - session-0.example.local
    environment:
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_MANAGER_MONITORING_PORT: "16100"
      LENA_CONFIG_TEMPLATE_ID: "session-cluster"
      LENA_SESSION_0_ADDRESS: "session-0.example.local:5180"
      LENA_SESSION_1_ADDRESS: "session-1.example.local:5180"
      LENA_SECONDARY_SESSION_NO: 1
    deploy:
      mode: replicated
      replicas: 1
      endpoint_mode: dnsrr
    healthcheck:
      test: "sh /usr/local/lena/servers/sessionServer/health.sh"
      interval: 5s
      timeout: 3s
      retries: 3
      start_period: 5s

  lena-session-1:
    image: lenacloud/lena-session:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - session-1.example.local
    environment:
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_MANAGER_MONITORING_PORT: "16100"
      LENA_CONFIG_TEMPLATE_ID: "session-cluster"
      LENA_SESSION_0_ADDRESS: "session-0.example.local:5180"
      LENA_SESSION_1_ADDRESS: "session-1.example.local:5180"
      LENA_SECONDARY_SESSION_NO: 0
    deploy:
      mode: replicated
      replicas: 1
      endpoint_mode: dnsrr
    healthcheck:
```

```
      test: "sh /usr/local/lena/servers/sessionServer/health.sh"
      interval: 5s
      timeout: 3s
      retries: 3
      start_period: 5s
```

- 배포실행
  배포는 docker stack deploy 명령으로 실행한다. 파일명이 lena-session-cluster.yaml이고 Stack명이 example이라면 배포 명령은 다음과 같다.

```
$ docker stack deploy \
    -c lena-net.yaml \
    -c lena-manager.yaml \
    -c lena-session-cluster.yaml \
    example
```

docker stack 명령은 구성요소를 일괄적으로 등록/삭제하므로 앞서 설명한 Network 및 Manager Service 명세를 모두 포함하여 실행하여야 한다.

- 배포결과 확인
  배포된 Workload는 docker stack services 명령어 실행을 통해 확인할 수 있다.

```
$ docker stack services example
ID                  NAME                      MODE                REPLICAS
IMAGE                                                 PORTS
.............       example_lena-session-1    replicated          1/1
lenacloud/lena-session:1.5.0.1-centos7-jdk8-openjdk
.............       example_lena-session-0    replicated          1/1
lenacloud/lena-session:1.5.0.1-centos7-jdk8-openjdk
... (출력 일부 생략)
```

5.3.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **ServerList버튼** 을 클릭하면 하단에 2개의 Session Server가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다.

5.4. WAS 배포

5.4.1. 배포 준비

본문서 WAS 배포준비 부분 설명을 참조한다.

5.4.2. 설정 항목

기본 설정 항목

WAS Container는 다음과 같은 권고 설정을 기준으로 배포 된다.

설정 관련 항목	설정 값 / 설명	비고
N/W 유형	Overlay Network - Ingress	-
End Point 유형	VIP	-
Container Port	TCP : 8180	-

적용시점 결정 항목 – Service 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Docker Compose 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-cluster:1.5.0.1-centos7-jdk8-openjdk
Stack명	Swarm Cluster내 논리적 그룹 명	example
Service 명	Service를 식별하는 이름으로, Stack내에서 유일한 값이 어야 한다.	lena-was-cluster
replica 개수	Container 개수	2
Health Check	HttpGetAction, 체크 Page, 시작 Delay 시간, 주기는 Application 특성에 맞도록 설정 필요	'/index.jsp' 호출

설정 가능한 환경변수는 다음과 같다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WAS-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.example.local:7700
LENA_MANAGER_KEY	• LENA_MANAGER_KEY : Open API로 Manager 접근시 필요한 인증도큰 • Manager의 Admin > Users 메뉴에서 확인 가능	(개별 Manager에서 확인 입력 필요)
LENA_CONFIG_TEMPLATE_DOWNLOAD	• 설정 정보 다운로드 여부	Y
LENA_CONTRACT_CODE	• 라이선스 다운로드를 위한 계약 코드 • 라이선스 발급 시 제공된 암호화된 코드 값	(개별 코드 확인 필요)

환경변수	설명	Sample 값
LENA_LICENSE_DOWNLOAD_URL	• License 다운로드 주소	manager(기본)
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	3 (기본)

5.4.3. 명세서 (Compose 파일) 기반 배포

다음은 LENA WAS를 구동하기 위한 Docker Compose 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA WAS Service 명세(lena-was-cluster.yaml) 파일 Example

```
version: "3.8"

services:
  lena-was-cluster:
    image: lenacloud/lena-cluster:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - was-cluster.example.local
    ports:
    - target: 8180
      published: 8180
      protocol: tcp
      mode: ingress
    environment:
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_MANAGER_MONITORING_PORT: "16100"
      LENA_MANAGER_KEY:
"C9PYwcu%2FDcB12IyhTrD63o2mXtnCWqV0dvnhunznedBnpRUgjVr6QeLA9zvReQqWMzbgyIHNVaj
oSvSuE0jANQ%3D%3D"
      LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
      LENA_CONFIG_TEMPLATE_ID: "was-cluster"
      LENA_LICENSE_DOWNLOAD_URL: "manager"
      LENA_CONTRACT_CODE: "vvW7ebNFVjDdtasuvF1utQ=="
      JAVA_DOMAIN_CACHE_TTL: "3"
    deploy:
      mode: replicated
      replicas: 2
    healthcheck:
      test: "curl -f http://localhost:8180/index.jsp"
      interval: 10s
      timeout: 5s
      retries: 3
      start_period: 20s
```

- 배포실행
  배포는 docker stack deploy 명령으로 실행한다. 파일명이 lena-was-cluster.yaml이라면 배포 명령은 다음과 같다.

```
$ docker stack deploy \
    -c lena-net.yaml \
    -c lena-manager.yaml \
    -c lena-session-cluster.yaml \
    -c lena-was-cluster.yaml \
    example
```

- 배포결과 확인
  배포된 Workload는 docker stack services 명령어 실행을 통해 확인할 수 있다.

```
$ docker stack services example
ID                  NAME                          MODE                REPLICAS
IMAGE                                                          PORTS
.............       example_lena-was-cluster      replicated          2/2
lenacloud/lena-cluster:1.5.0.1-centos7-jdk8-openjdk    *:8180->8180/tcp
... (출력 일부 생략)
```

5.4.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **Server List 버튼** 을 클릭하면 하단에 2개의 WAS가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다.

5.5. Embedded WAS 배포

5.5.1. 배포 준비

본문서 Embedded WAS 배포준비 부분 설명을 참조한다.

5.5.2. 설정 항목

기본 설정 항목

WAS Container는 다음과 같은 권고 설정을 기준으로 배포 된다.

설정 관련 항목	설정 값 / 설명	비고
N/W 유형	Overlay Network - Ingress	-
End Point 유형	VIP	-
Container Port	TCP : 8180	-

적용시점 결정 항목 – Service 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Docker Compose 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-embedded:1.5.0.1-centos7-jdk8-openjdk
Stack명	Swarm Cluster내 논리적 그룹 명	example
Service 명	Service를 식별하는 이름으로, Stack내에서 유일한 값이 어야 한다.	lena-was-cluster
replica 개수	Container 개수	2
Health Check	HttpGetAction, 체크 Page, 시작 Delay 시간, 주기는 Application 특성에 맞도록 설정 필요	'/index.jsp' 호출

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Embedded WAS 환경변수 부분을 참조한다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WAS-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.default.svc.cluster.local:7700
LENA_MANAGER_MONITORING_PORT	• Manager 모니터링 Port 정보	16100
LENA_APP_FILE	• Application Jar 파일 명	sample-app.jar
LENA_APP_DIR	• Application Jar 디렉토리 명	/usr/local/lena

5.5.3. 명세서 (Compose 파일) 기반 배포

다음은 LENA Embedded WAS를 구동하기 위한 Docker Compose 파일 샘플이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도록 변경하여 사용할 수 있다.

LENA Embedded WAS Service 명세(lena-was-embedded.yaml) 파일 Example

```yaml
version: "3.8"

services:
  lena-was-cluster:
    image: lenacloud/lena-embedded:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - was-cluster.example.local
    ports:
    - target: 8180
      published: 8180
      protocol: tcp
      mode: ingress
    environment:
      LENA_CONFIG_TEMPLATE_ID: "WAS-001"
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_APP_FILE: "sample-app.jar"
    deploy:
      mode: replicated
      replicas: 2
    healthcheck:
      test: "curl -f http://localhost:8180/index.jsp"
      interval: 10s
      timeout: 5s
      retries: 3
      start_period: 20s
```

- 배포실행
  배포는 docker stack deploy 명령으로 실행한다. 파일명이 lena-was-embedded.yaml이라면 배포 명령은 다음과 같다.

```
$ docker stack deploy \
    -c lena-net.yaml \
    -c lena-manager.yaml \
    -c lena-session-cluster.yaml \
    -c lena-was-embedded.yaml \
    example
```

- 배포결과 확인
  배포된 Workload는 docker stack services 명령어 실행을 통해 확인할 수 있다.

```
$ docker stack services example
ID                  NAME                          MODE                REPLICAS
IMAGE                                                                   PORTS
............        example_lena-was-embedded   replicated          2/2
lenacloud/lena-embedded:1.5.0.1-centos7-jdk8-openjdk   *:8180->8180/tcp
... (춘력 일부 생략)
```

5.5.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **Server List 버튼** 을 클릭하면 하단에 2개의 WAS가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다.

5.6. Web Server 배포

5.6.1. 배포 준비

본 문서 Web Server 배포 준비 부분의 설명을 참조한다.

5.6.2. 설정 항목

기본 설정 항목

Web Server Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

설정 관련 항목	설정 값 / 설명	비고
N/W 유형	Overlay Network - Ingress	-
End Point 유형	VIP	-
Container Port	TCP : 7180	-

적용시점 결정 항목 – Service 관련

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Docker Compose 파일에 적용된 Sample 값은 다음과 같다.

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-web:1.5.0.1-centos7-jdk8-openjdk
Stack명	Swarm Cluster내 논리적 그룹 명	example
Service 명	Service를 식별하는 이름으로, Stack내에서 유일한 값이 어야 한다.	lena-web-cluster
Network Alias	Swarm Cluster내 N/W식별자 (Domain 주소)	web-cluster.example.local

설정 관련 항목	설명	Sample 값
replica 개수	Container 개수	2
Health Check	HttpGetAction, 체크 Page, 시작 Delay 시간, 주기는 Application 득성에 맞도독 설정 필요	'/index.html' 호촏

설정 가능한 환경변수는 다음과 같다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WEB-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.example.local:7700
LENA_MANAGER_KEY	• LENA_MANAGER_KEY : Open API도 Manager 접근시 필요한 인증도큰 • Manager의 Admin > Users 메뉴에서 확인가능	(개볃 Manager에서 확인 입뎍 필요)
LENA_CONFIG_TEMPLATE_DOWNLOAD	• 설정 정보 다운도드 여부	Y
LENA_CONTRACT_CODE	• 라이선스 다운도드를 위한 계약 코드 • 라이선스 받급 시 제공된 코드 값	(개볃 코드 확인 필요)
LENA_LICENSE_DOWNLOAD_URL	• 라이선스 다운도드 위치	manager (기본)
LENA_RUN_AGENT	• Agent 실행여부	Y

5.6.3. 명세서 (Compose 파일) 기반 배포

다음은 LENA WEB Server를 기동하기 위한 Docker Compose 파일 샘픋이고, 상세 내용은 앞서 설명된 설정 항목의 결정에 따라 Project 환경에 맞도독 변경하여 사용핟 수 있다.

LENA Web Service 명세(lena-web-cluster.yaml) 파일 Example

```
version: "3.8"

services:
  lena-web-cluster:
    image: lenacloud/lena-web:1.5.0.1-centos7-jdk8-openjdk
    networks:
      lena-net:
        aliases:
        - web-cluster.example.local
    ports:
    - target: 7180
      published: 7180
      protocol: tcp
      mode: ingress
    environment:
      LENA_MANAGER_ADDRESS: "lena-manager.example.local:7700"
      LENA_MANAGER_MONITORING_PORT: "16100"
      LENA_MANAGER_KEY:
"C9PYwcu%2FDcB12IyhTrD63o2mXtnCWqV0dvnhunznedBnpRUgjVr6QeLA9zvReQqWMzbgyIHNVaj
oSvSuE0jANQ%3D%3D"
      LENA_CONFIG_TEMPLATE_DOWNLOAD: "Y"
      LENA_CONFIG_TEMPLATE_ID: "web-cluster"
      LENA_LICENSE_DOWNLOAD_URL: "manager"
      LENA_CONTRACT_CODE: "vvW7ebNFVjDdtasuvF1utQ=="
      LENA_AGENT_RUN: "Y"
    deploy:
      mode: replicated
      replicas: 2
    healthcheck:
      test: "curl -f http://localhost:7180/index.html"
      interval: 10s
      timeout: 5s
      retries: 3
      start_period: 20s
```

- 배포실행
  배포는 docker stack deploy 명령으로 실행한다. 파일명이 lena-web-cluster.yaml이라면 배포 명령은 다음과 같다.

```
$ docker stack deploy \
    -c lena-net.yaml \
    -c lena-manager.yaml \
    -c lena-session-cluster.yaml \
    -c lena-was-cluster.yaml \
    -c lena-web-cluster.yaml \
    example
```

- 배포결과 확인
  배포된 Workload는 docker stack services 명령어 실행을 통해 확인할 수 있다.

```
$ docker stack services example
ID                  NAME                        MODE                REPLICAS
IMAGE                                                   PORTS
.............       example_lena-web-cluster    replicated          2/2
lenacloud/lena-web:1.5.0.1-centos7-jdk8-openjdk    *:7180->7180/tcp
... (출력 일부 생략)
```

5.6.4. Server 등록 확인

Manager에 접속하여 'Cluster > Service Cluster' 메뉴에서 해당 Service Cluster를 선택하여 정상 등록여부를 확인한다. **Server List 버튼** 을 클릭하면 하단에 2개의 Web Server가 조회되는 것을 확인한다.

Server는 Manager의 Scheduler에 의해 등록되므로, 최대 15초 후 자동 등록된다

Chapter 6. ECS 기반 설치

본 장에서는 ECS환경에서 LENA 서버 또는 LENA 이미지 기반의 애플리케이션을 Container로 배포하는 방법을 설명한다. AWS Console을 통해서 설치하는 방법을 설명한다. 또한, 일반적인 ECS 설정에 대한 부분은 제외하고, ECS환경에서 LENA를 구동하기 위해 필요한 내용만 다룬다.

6.1. ECS 개요

ECS는 AWS에서 제공하는 Container 서비스 플랫폼으로, Docker를 이용하여 EC2 인스턴스 상에서 Container를 배포/운영하고, 동일 Service를 제공하는 Container를 Task 단위로 묶어서 Replica, Service Discovery, L/B, Auto-scale 정책 등을 관리할 수 있는 기능을 제공한다.

6.2. 설치 준비

다음과 같은 ECS 환경이 미리 준비되어 있어야 한다.

- ECS Cluster 및 ECS 작업권한
- ECS 환경에서 접근 가능한 Docker Registry (예 : ECR, Docker Hub)

추가적으로 Container 기반 Manager를 설치하기 위해서는 외부 저장소가 필요하고, 이것은 Manager Container의 생성/소멸 시에도 기존 데이터 및 파일을 지속적으로 유지하기 위해 사용된다.

- EFS 등 외부 저장소 가능 환경 확인

Container 기반 Manager 및 Session Server를 설치하기 위해서는 Service Discovery 기능이 적용되는 환경(awsvpc 기반 EC2, Fargate 및 지원 Region)인지 사전 확인이 필요하다.

- ECS Service Discovery 적용 가능 환경

6.3. Manager 배포

ECS환경에서도 Manager Instance의 영속성 보장을 하기 위해서 1) 고정주소 2) 외부 Volume 두 가지가 필요하다. 고정 주소는 ECS의 Service Discovery나 ELB 연결을 통해서 가능하며, 외부 Volume은 EFS 연결를 통해 제공 가능하다. 하기에서는 Service Discovery와 EFS를 활용한 Manager 배포에 대해 설명한다.

6.3.1. 설정 항목

기본 설정 항목

Manager Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

Table 2. ECS기반 Manager 설치 - 배포 기준

설정 관련 항목	설정 값 / 설명	비고
Service 유형	Replica	-
Replica개수	1	-
Service Discovery	Service Discovery 사용	-
Volume Mount	디렉토리 /usr/local/lena/repository를 EFS에 연결	-

적용시점 결정 항목

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 설치과정에서 사용되는 Sample 값은 다음과 같다.

Table 3. ECS기반 Manager 설치 - 적용시점 결정 항목

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-manager.1.3.1.10_3-jdk8-openjdk
Probe (Health Check)	HttpGetAction, '/lena' 페이지 호출	
Task 및 Service name	Task 및 Service 의 이름	lena-manager
Service Namespace	Service검색시 Domain 주소의 Suffix도 사용	local
Service Discovery Name	Service검색시 Domain 주소의 Prefix도 사용	lena-manager

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본 문서의 Manager 환경변수 부분 참조한다.

Table 4. ECS기반 Manager 설치 - 환경변수

환경변수	설명	Sample 값
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m (기본)
LENA_JVM_METASPACE_SIZE	• Metaspace Memory크기	256m (기본)
LENA_MANAGER_DOMAIN_ENABLED	• Domain Name 활성화 여부	Y
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트	lena-manager.local:7700
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	3 (기본)

6.3.2. Task 설정

Task의 이름, 권한등은 Project의 표준에 따라 입력하길 권고하며, 하기에서는 Container 정의 중 LENA 운영을 위해 필요한 부분만 설명한다. 하기에 설명되는 설정의 기준은 설정 항목 부분의 설명을 참조한다.

Task 정의

Container 정보를 포함하는 Task를 정의한다.

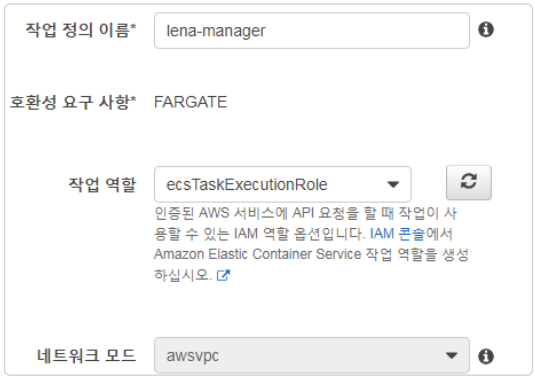


Figure 17. ECS기반 Manager 설치 - Task 정의

Volume 추가

Task 정의에서 Manager Repository를 저장할 EFS와 저장위치를 정보를 추가한다.

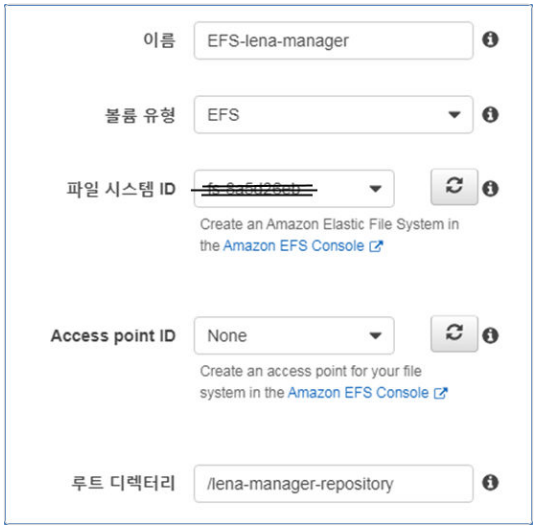


Figure 18. ECS기반 Manager 설치 - Volume 추가

Container 추가

이름, 이미지등의 기본 Container정보를 입력한다.

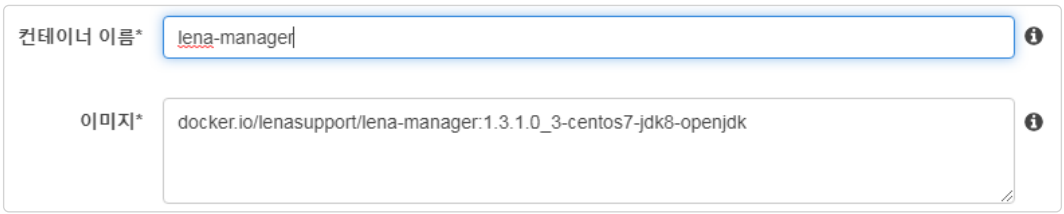


Figure 19. ECS기반 Manager 설치 - Container 추가

환경변수 설정

Container 설정중 환경변수를 추가한다. Manager 내부에서 Manager의 Service Discovery 주소를 인식하지 못하므로 이를 환경변수 LENA_MANAGER_ADDRESS로 입력을 한다.

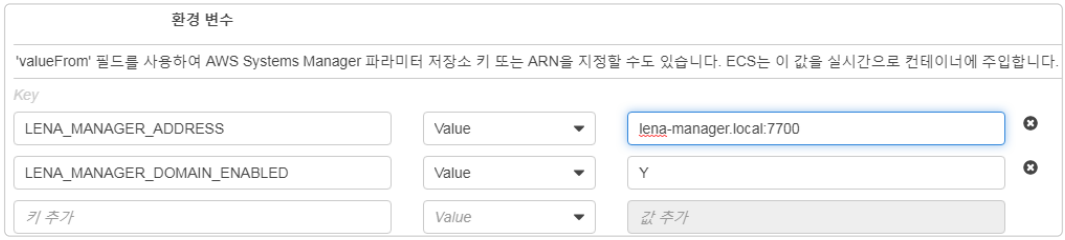


Figure 20. ECS기반 Manager 설치 - 환경변수 설정

Health Check 설정

http://localhost:7700/lena/ 페이지를 호출하는 Health Check 정보를 입력한다.

Volume 매핑

앞서 추가한 Volume을 Container내부 Directory와 매핑한다.

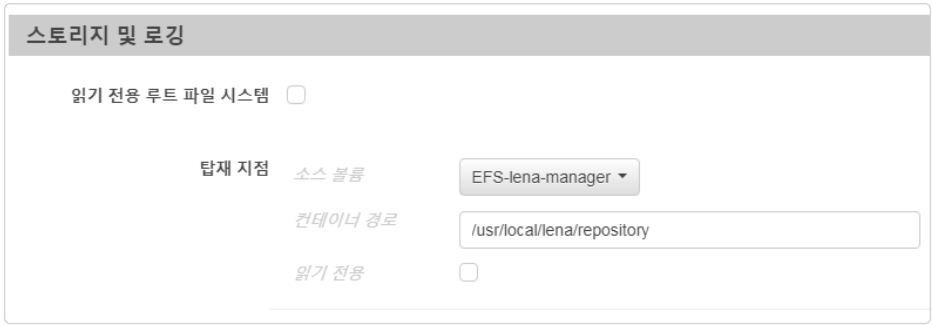


Figure 21. ECS기반 Manager 설치 - Volume 매핑

6.3.3. Service 설정

서비스 정의

앞서 정의한 Task를 실제 기동/운영하기 위한 Service를 정의한다.

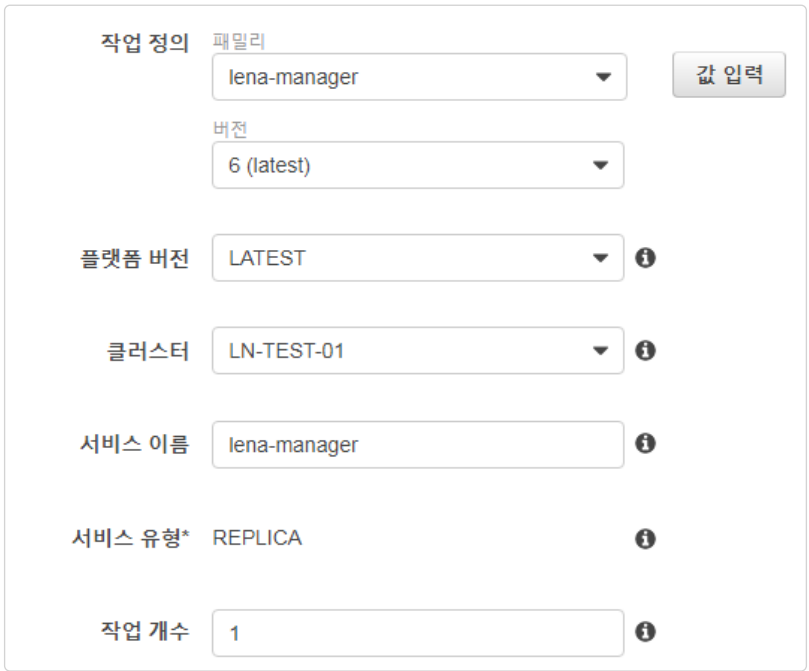


Figure 22. ECS기반 Manager 설치 - Service 정의

서비스 검색 (Service Discovery) 설정

AWS는 ECS간 서비스들은 연결하여 사용할 수 있도록 서비스 검색 (Service Discovery) 기능을 지원한다. Manager는 Service Discovery 기능을 이용하여 고정된 주소를 확보하여, 타 Server의 설정 관리 및 모니터링 기능을 제공한다.

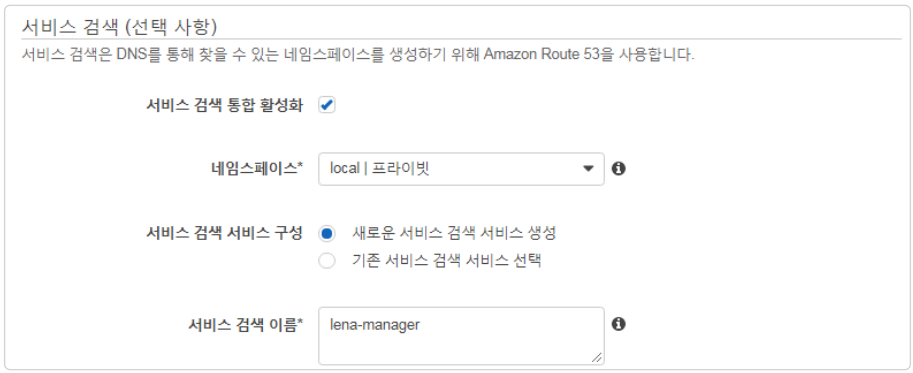


Figure 23. ECS기반 Manager 설치 - Service Discovery 설정

6.3.4. Service 기동 및 확인

Service 기동은 Service 정의를 저장하면 기동이 시작된다. ECS Cluster의 화면에서 서비스 및 작업 탭에서 운영중인 Service와 Task의 상태를 확인한다.

Service 상태 확인

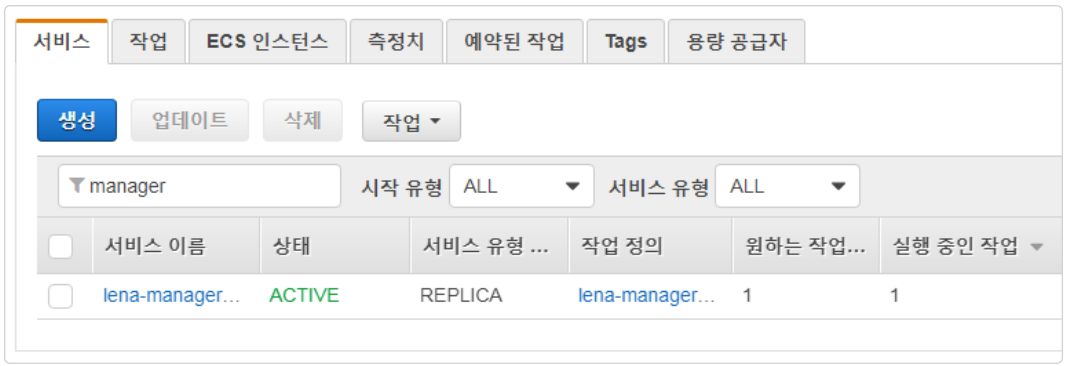


Figure 24. ECS기반 Manager 설치 - Service 상태 확인

Task 상태 확인

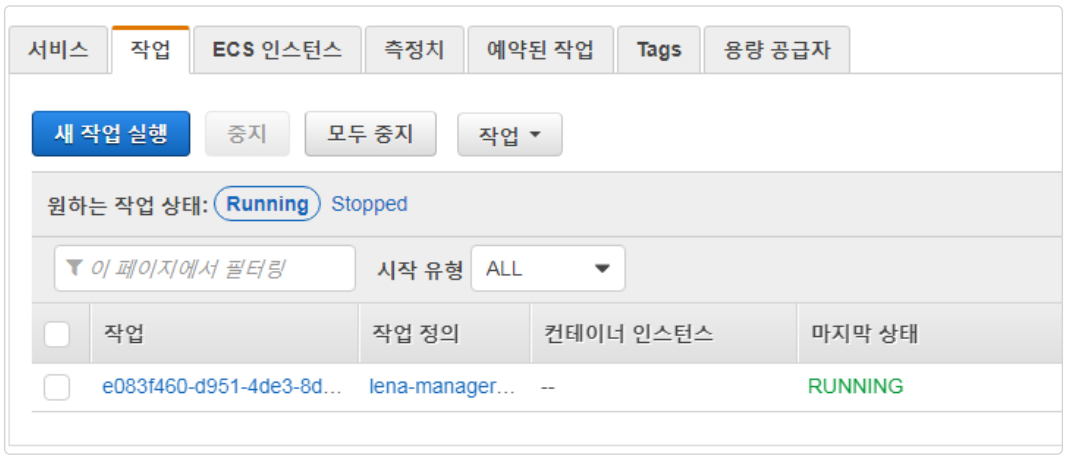


Figure 25. ECS기반 Manager 설치 - Task 상태 확인

6.4. Session Server 배포

ECS에서 환경에서도 Session Server의 서비스를 위해 고정주소가 필요하다. 고정 주소는 ECS의 Service Discovery나 ELB 연결을 통해서 가능하다. 다음에서는 Service Discovery 를 활용한 Session Server 배포에 대해 설명한다.

6.4.1. 배포 준비

본문서 Session Server 배포 준비 부분 설명을 참조한다.

6.4.2. 설정 항목

기본 설정 항목

Session Server Container는 다음과 같은 권고 설정을 기준으로 배포되어야 한다.

Table 5. ECS 기반 Session Server 배포 기준

설정 관련 항목	설정 값 / 설명	비고
Service 종류	Replica	-
Service / Replica개수	Service 2개, 각 Service 별 Replica 1개	-
Container Port	TCP : 5180	-
Probe	${LENA_SERVER_HOME}/healt h.sh 호출	-

적용시점 결정 항목

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Task 및 Service 설정에 적용된 Sample 값은 다음과 같다.

Table 6. ECS기반 Session Server 설치 - 적용시점 결정 항목

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-session:1.3.1.10-jdk8-openjdk
Task 및 Service name	Task 및 Service 의 이름	lena-session
Service Namespace	Service검색시 Domain 주소의 Suffix로 사용	local
Service Discovery Name	Service검색시 Domain 주소의 Prefix로 사용	lena-session

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 문서 Session Server 환경변수 부분 참조

Table 7. ECS기반 Session Server 설치 - 환경변수

환경변수	설명	Sample 값
LENA_JVM_HEAP_SIZE	• Heap Memory 크기 지정	1024m (기본)
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	SESSION-001
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트	lena-manager.local:7700
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	0 (기본)
LENA_SESSION_0_ADDRESS	• Primary Session 서버의 서비스 주소, Service 명 및 Service Discovery 설정에 따른 도메인 주소와 일치되어야 함	lena-session-0.local:5180
LENA_SESSION_1_ADDRESS	• Secondary Session 서버의 서비스 주소, Service 명 및 Service Discovery 설정에 따른 도메인 주소와 일치되어야 함	lena-session-1.local:5180
LENA_SESSION_EXPIRE_SEC	• Session 만료 시간 (초)	1800 (기본)
LENA_CONFIG_SHARE_SESSION	• Application 간 Session 공유여부	N

6.4.3. Task 설정

Task의 이름, 권한등은 Project의 표준에 따라 입력하기를 권고하며, 다음에서는 Container 정의 중 LENA 운영을 위해 필요한 부분만 설명한다. 하기에 설명되는 설정의 기준은 설정항목 부분의 설명을 참조한다.

Task 정의

Container 정보를 포함하는 Task를 정의한다.

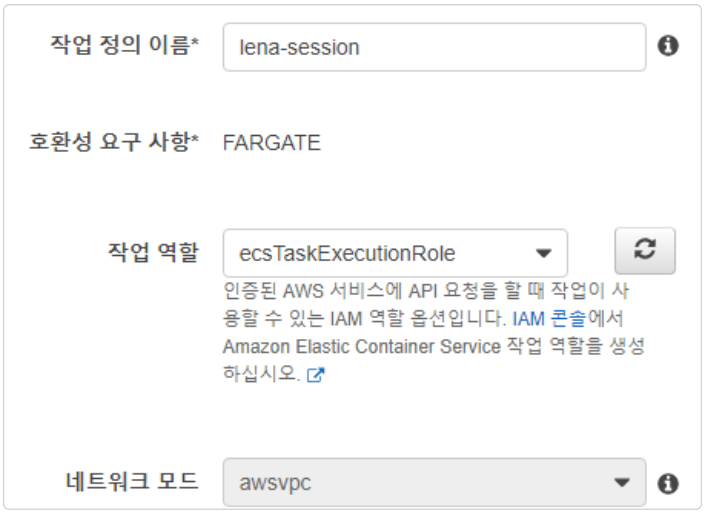


Figure 26. ECS기반 Session Server 설치 - Task 정의

Container 추가

이름, 이미지등의 기본 Container정보를 입력한다.

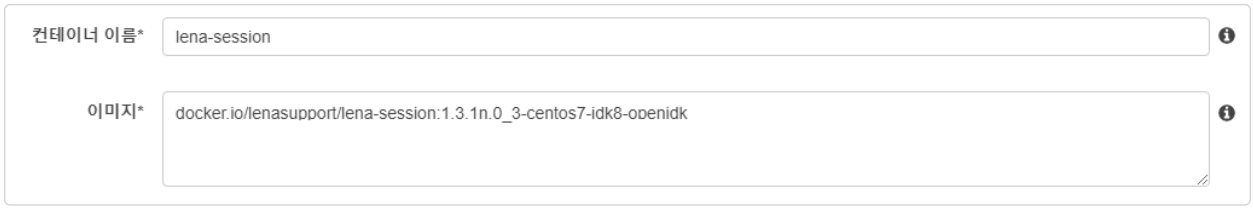


Figure 27. ECS기반 Session Server 설치 - Container 추가

환경변수 설정

Container 설정 중 환경변수를 추가한다.

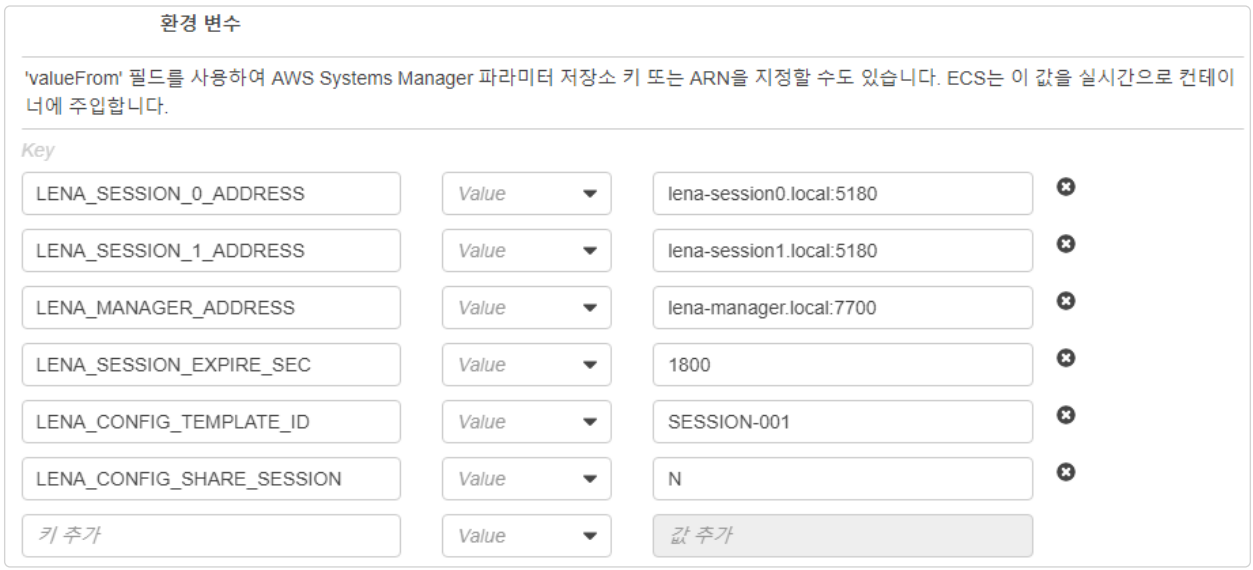


Figure 28. ECS기반 Session Server 설치 - 환경변수 설정

LENA_SESSION_1_ADDRESS 환경변수에 반드시 Secondary Session Service의 Service Discovery 주소를 입력한다.

6.4.4. Service 설정

서비스 정의

앞서 정의한 Task를 실제 기동/운영하기 위한 Service를 정의한다.

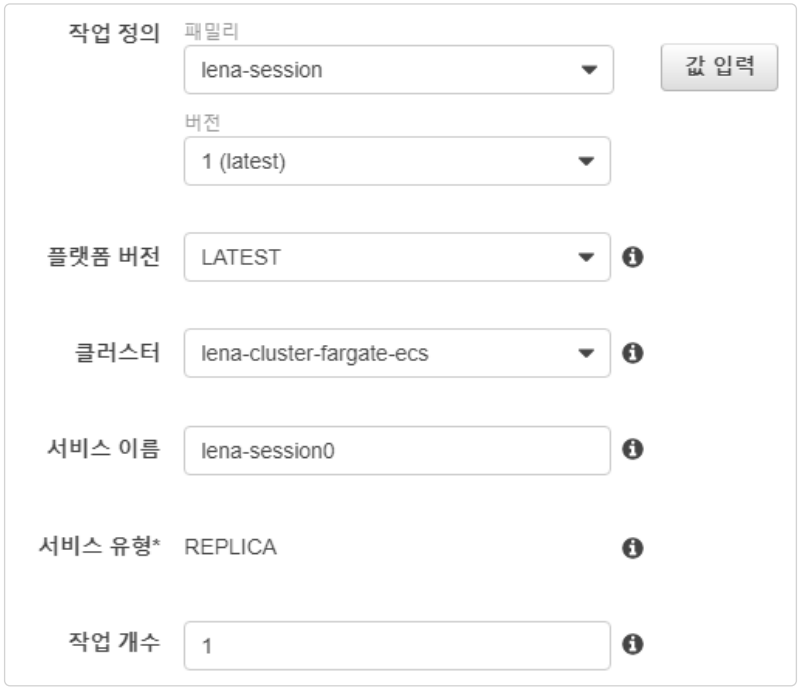


Figure 29. ECS기반 Session Server 설치 - Service 정의

서비스 검색 (Service Discovery) 설정

AWS는 ECS간 서비스들은 연결하여 사용할 수 있도록 서비스 검색 (Service Discovery) 기능은 지원한다. Manager는 Service Discovery 기능은 이용하여 고정된 주소를 확보하여, 타 Server의 설정 관리 및 모니터링 기능은 제공한다.

Figure 30. ECS기반 Session Server 설치 - Service Discovery 설정

6.4.5. Service 기동 및 확인

Service 기동은 Service 정의를 저장하면 기동이 시작된다. ECS Cluster의 화면에서 서비스 및 작업 탭에서 운영중인 Service와 Task의 상태를 확인한다

Service 상태 확인

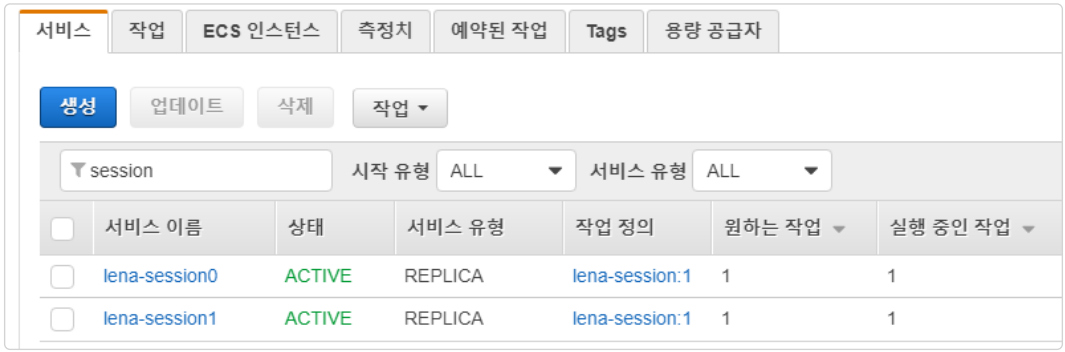


Figure 31. ECS기반 Session Server 설치 - Service 상태 확인

Task 상태 확인

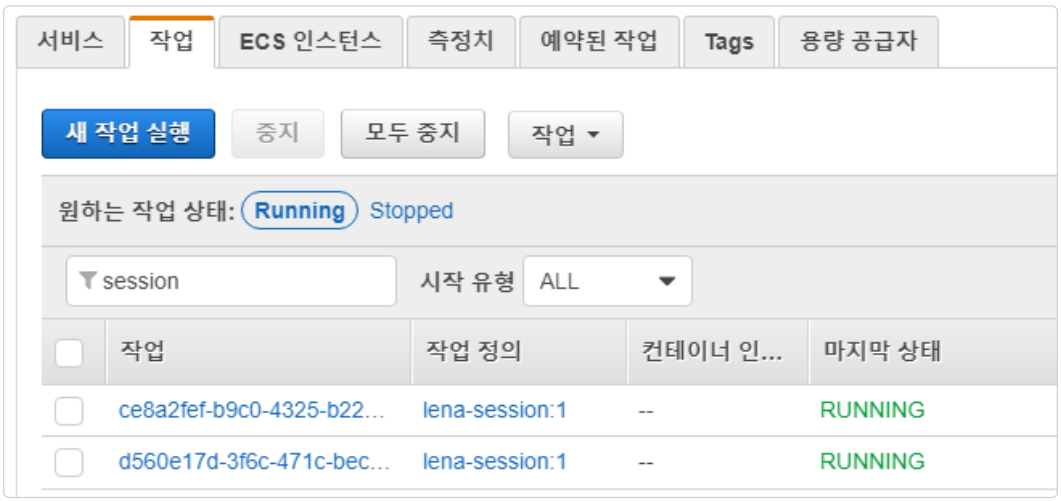


Figure 32. ECS기반 Session Server 설치 - Task 상태확인

6.5. WAS 배포

6.5.1. 배포 준비

본문서 WAS 배포준비 부분 설명은 참조한다.

6.5.2. 설정 항목

기본 설정 항목

WAS Container는 다음과 같은 권고 설정은 기준으로 배포 된다..

Table 8. ECS기반 WAS 설치 - 배포 기준

설정 관련 항목	설정 값 / 설명	비고
Service 종듀	Replica	
Container Port	TCP : 8180	-

적용시점 결정 항목

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Task 및 Service 설정에 적용된 Sample 값은 다음과 같다.

Table 9. ECS기반 WAS 설치 - 적용시점 결정 항목

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-cluster:1.5.0.1-jdk8-openjdk
Task 및 Service name	Task 및 Service 의 이름	lena-was
label	Service와의 연결, 검색에 사용되는 Label도 Key: Value쌍으로 정의된다.	type: lena-was
Service Namespace	Service검색시 Domain 주소의 Suffix도 사용	local
Service Discovery Name	Service검색시 Domain 주소의 Prefix도 사용	lena-was
Probe (Health Check)	체크 Page, 시작 Delay 시간, 주기는 Application 특성에 맞도록 설정 필요	'/' 호출

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본 문서 WAS 환경변수 부분을 참조한다.

Table 10. ECS기반 WAS 설치 - 환경변수

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WAS-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.local:7700
LENA_MANAGER_KEY	• LENA_MANAGER_KEY : Open API도 Manager 접근시 필요한 인증도큰 • Manager의 Admin > Users 메뉴에서 확인 가능	(개별 Manager의 Administration > IAM > Users 메뉴에서 확인 및 입력 필요)
LENA_CONFIG_TEMPLATE_DOWNLOAD	• 설정 정보 다운도드 여부	Y
LENA_CONTRACT_CODE	• 라이선스 다운도드를 위한 계약 코드 • 라이선스 받급 시 제공된 코드 값	(개별 코드 확인 필요)
LENA_LICENSE_DOWNLOAD_URL		manager
JAVA_DOMAIN_CACHE_TTL	• Domain 주소 Cache 시간 (초)	3 (기본)

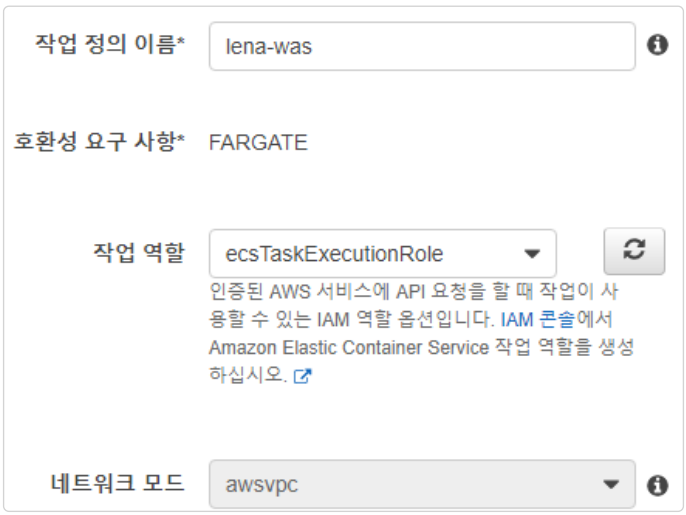
6.5.3. Task 설정

Task의 이름, 권한등은 Project의 표준에 따라 입력하기를 권고하며, 하기에서는 Container 정의 중 LENA 운영을 위해 필요한 부분만 설명한다. 하기에 설명되는 설정의 기준은 본문서 설정 항목 부분의 설명을 참조한다.

Task정의

ECS기반 WAS 설치 - Task정의

Container 정보를 포함하는 Task를 정의한다.



Container 추가

이름, 이미지등의 기본 Container정보를 입력한다.

Figure 33. ECS기반 WAS 설치 - Container 추가

환경변수 설정

Container 설정중 환경변수를 추가한다.

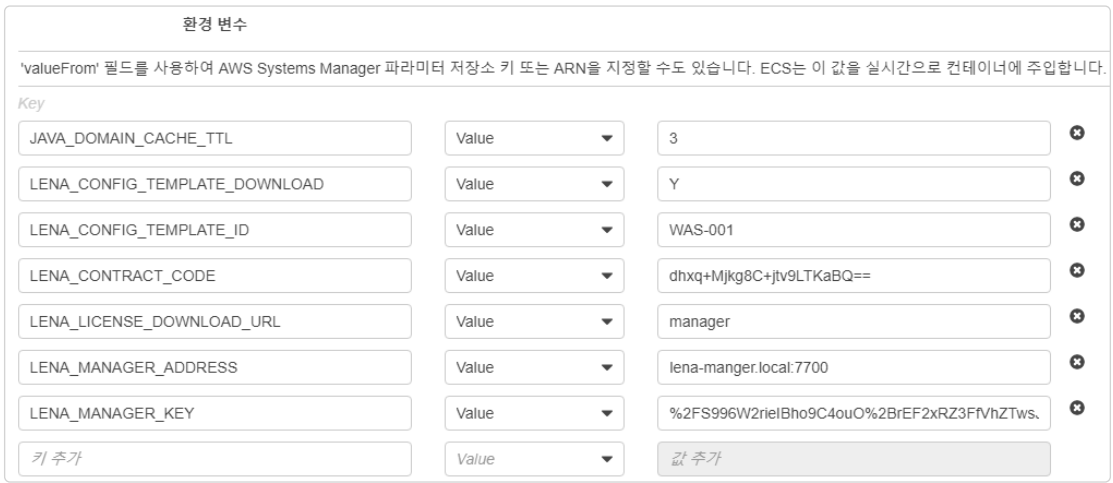


Figure 34. ECS기반 WAS 설치 - 환경변수 설정

6.5.4. Service 설정

서비스 정의

앞서 정의한 Task를 실제 기동/운영하기 위한 Service를 정의한다.

Figure 35. ECS기반 WAS 설치 - Service 정의

서비스 검색 (Service Discovery) 설정

AWS는 ECS간 서비스들을 연결하여 사용할 수 있도록 서비스 검색 (Service Discovery) 기능을 지원한다. Manager는 Service Discovery 기능을 이용하여 고정된 주소를 확보하여, 타 Server의 설정 관리 및 모니터링 기능을 제공한다.

Figure 36. ECS기반 WAS 설치 - Service Discovery 설정

6.5.5. Service 기동 및 확인

Service 기동은 Service 정의를 저장하면 기동이 시작된다. ECS Cluster의 화면에서 서비스 및 작업 탭에서 운영중인 Service와 Task의 상태를 확인한다.

Service 상태 확인

Figure 37. ECS기반 WAS 설치 - Service 상태 확인

Task 상태 확인

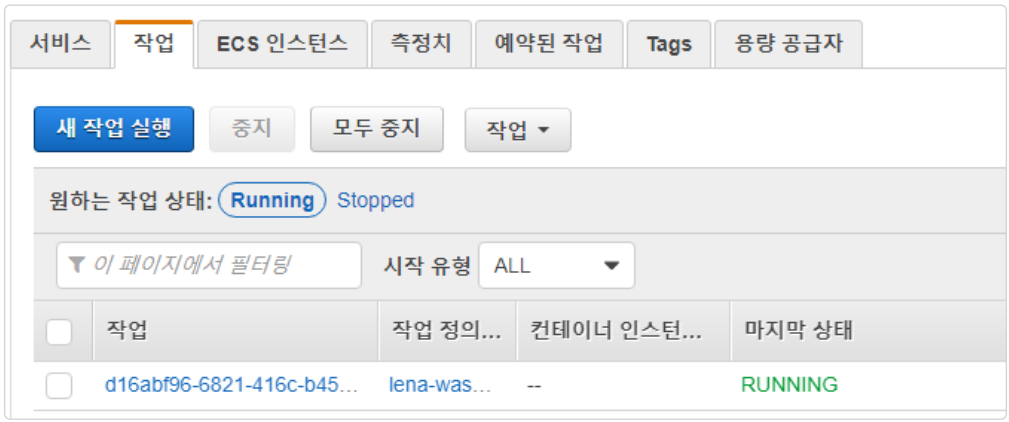


Figure 38. ECS기반 WAS 설치 - Task 상태 확인

6.6. Embedded WAS 배포

6.6.1. 배포 준비

본문서 Embedded WAS 배포준비 부분 설명은 참조한다.

6.6.2. 설정 항목

기본 설정 항목

Embedded WAS Container는 다음과 같은 권고 설정은 기준으로 배포 된다..

Table 11. ECS기반 Embedded WAS 설치 - 배포 기준

설정 관련 항목	설정 값 / 설명	비고
Service 종듀	Replica	
Container Port	TCP : 8180	-

적용시점 결정 항목

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 Task 및 Service 설정에 적용된 Sample 값은 다음과 같다.

Table 12. ECS기반 WAS 설치 - 적용시점 결정 항목

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-embedded:1.5.0.1-jdk8-openjdk
Task 및 Service name	Task 및 Service 의 이름	lena-was
label	Service와의 연결, 검색에 사용되는 Label도 Key: Value쌍으도 정의된다.	type: lena-was
Service Namespace	Service검색시 Domain 주소의 Suffix도 사용	local
Service Discovery Name	Service검색시 Domain 주소의 Prefix도 사용	lena-was
Probe (Health Check)	체크 Page, 시작 Delay 시간, 주기는 Application 득성에 맞도독 설정 필요	'/' 호춛

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Embedded WAS 환경변수 부분을 참조한다.

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No	WAS-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.default.svc.cluster.local:7700
LENA_MANAGER_MONITORING_PORT	• Manager 모니터딩 Port 정보	16100
LENA_APP_FILE	• Application Jar 파일 명	sample-app.jar
LENA_APP_DIR	• Application Jar 디덱도리 명	/usr/local/lena

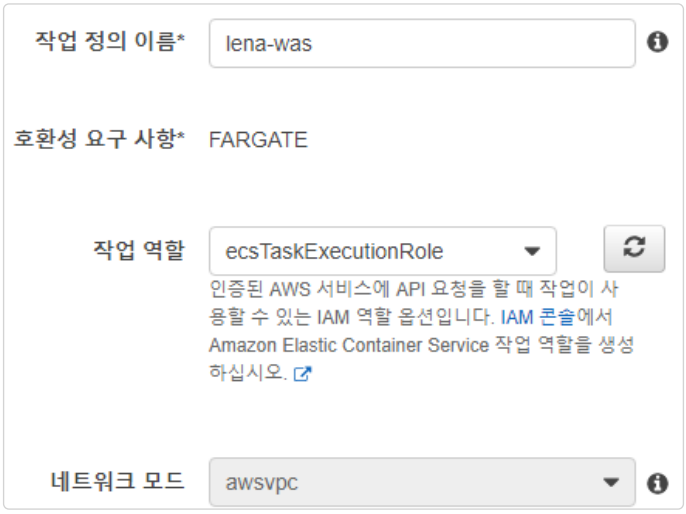
6.6.3. Task 설정

Task의 이름, 권한등은 Project의 표준에 따라 입뎍하기를 권고하며, 하기에서는 Container 정의 중 LENA 운영을 위해 필요한 부분만 설명한다. 하기에 설명되는 설정의 기준은 본문서 설정 항목 부분의 설명을 참조한다.

Task정의

ECS기반 Embedded WAS 설치 - Task정의

Container 정보를 포함하는 Task를 정의한다.



Container 추가

이름, 이미지등의 기본 Container정보를 입력한다.

Figure 39. ECS기반 Embedded WAS 설치 - Container 추가

환경변수 설정

Container 설정중 환경변수를 추가한다.

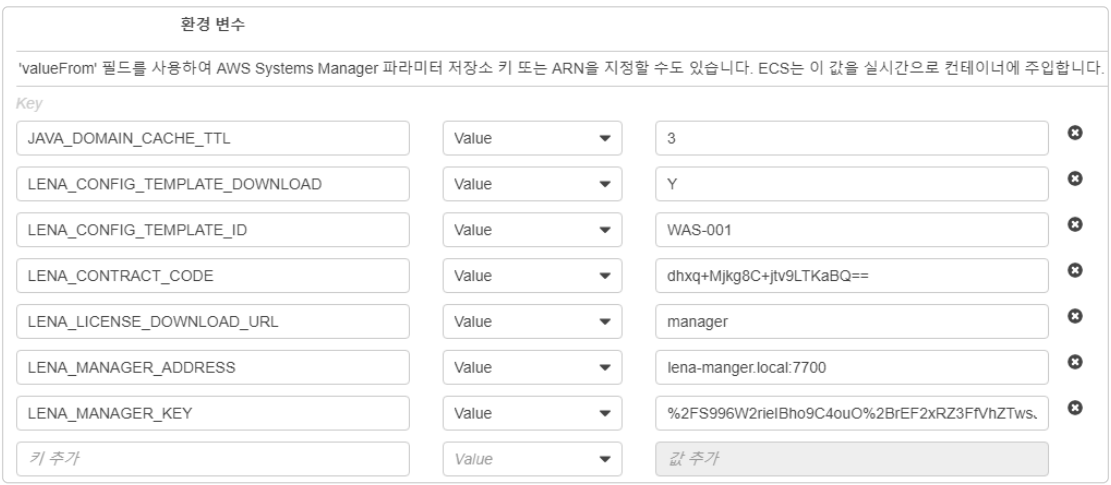


Figure 40. ECS기반 Embedded WAS 설치 - 환경변수 설정

6.6.4. Service 설정

서비스 정의

앞서 정의한 Task를 실제 기동/운영하기 위한 Service를 정의한다.

Figure 41. ECS기반 WAS 설치 - Service 정의

서비스 검색 (Service Discovery) 설정

AWS는 ECS간 서비스들은 연결하여 사용할 수 있도록 서비스 검색 (Service Discovery) 기능은 지원한다. Manager는 Service Discovery 기능은 이용하여 고정된 주소를 확보하여, 타 Server의 설정 관리 및 모니터링 기능은 제공한다.

Figure 42. ECS기반 Embedded WAS 설치 - Service Discovery 설정

6.6.5. Service 기동 및 확인

Service 기동은 Service 정의를 저장하면 기동이 시작된다. ECS Cluster의 화면에서 서비스 및 작업 탭에서 운영중인 Service와 Task의 상태를 확인한다.

Service 상태 확인

Figure 43. ECS기반 WAS 설치 - Service 상태 확인

Task 상태 확인

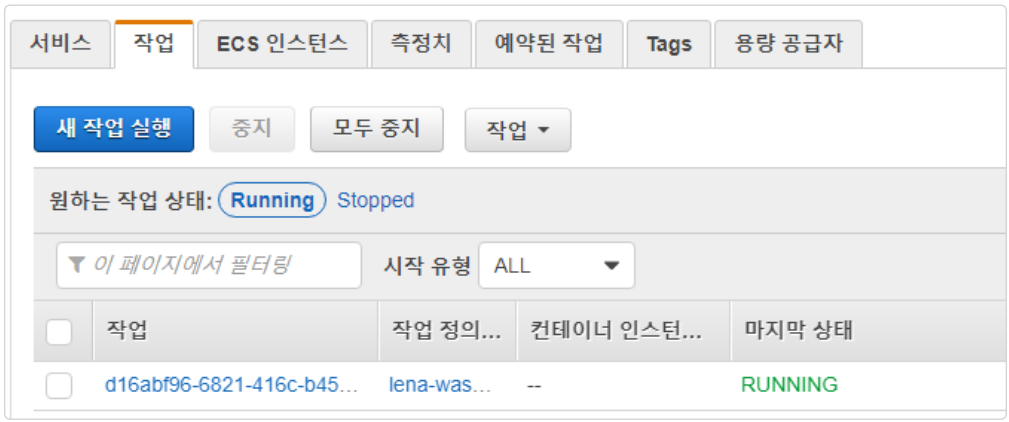


Figure 44. ECS기반 WAS 설치 - Task 상태 확인

6.7. Web Server 배포

6.7.1. 배포준비

본 문서 Web Server 배포 준비 부분의 설명은 참조한다.

6.7.2. 설정 항목

기본 설정 항목

Web Server Container는 다음과 같은 권고 설정은 기준으로 배포되어야 한다.

Table 13. ECS기반 Web Server 설치 - 배포 기준

설정 관련 항목	설정 값 / 설명	비고
Service 종듀	Replica	-
Container Port	TCP : 7180	-

적용시점 결정 항목

적용 시점 Project 환경에 따라 결정해서 적용해야 할 설정 요소와 이후 설명되는 ECS Task 및 Service 설정에 적용된 Sample 값은 다음과 같다.

Table 14. ECS기반 Web Server 설치 - 적용시점 결정 항목

설정 관련 항목	설명	Sample 값
Container Image	Project별 Architecture 결정에 따라 선별된 OS 및 JDK 버전에 맞는 LENA Manager Image	lenacloud/lena-web:1.3.1.10_3-jdk8-openjdk
Task 및 Service name	Task의 이름, 이 값은 Task 이름 / Hostname의 Prefix도 사용된다.	lena-web
replica 개수	Container (Task) 개수	2
Probe (Health Check)	시작 Delay시간, 주기는 Application 득성에 맞게 설정 필요	'/' 호춛
Service Namespace	Service검색시 Domain 주소의 Suffix도 사용	local
Service Discovery Name	Service검색시 Domain 주소의 Prefix도 사용	lena-web

설정 가능한 환경변수는 다음과 같다. 각 환경변수에 대한 상세한 설명은 본문서 Web Server 환경변수 부분을 참조한다.

Table 15. ECS기반 Web Server 설치 - 환경변수

환경변수	설명	Sample 값
LENA_CONFIG_TEMPLATE_ID	• Service Cluster 명 : Revision No • Revision No가 빈값인 경우 Default Revision을 다운도드 받음	WEB-001:1
LENA_MANAGER_ADDRESS	• Service의 Domain 주소 : 포트 (앞서 설치된 Manager의 Service 주소)	lena-manager.local:7700
LENA_MANAGER_KEY	• LENA_MANAGER_KEY : Open API도 Manager 접근시 필요한 인증도큰 • Manager의 Admin > Users 메뉴에서 확인가능	(개별 Manager에서 확인 입력 필요)
LENA_CONFIG_TEMPLATE_DOWNLOAD	• 설정 정보 다운도드 여부	Y
LENA_CONTRACT_CODE	• 라이선스 다운도드를 위한 계약 코드 • 라이선스 받급 시 제공된 코드 값	(개별 코드 확인 필요)
LENA_LICENSE_DOWNLOAD_URL	• 라이선스 다운도드 위치	manager
LENA_RUN_AGENT	• Agent 실행여부	Y

6.7.3. Task 설정

Task의 이름, 권한등은 Project의 표준에 따라 입력하고, Container 정의 중 LENA 운영을 위해 필요한 부분만 설명한다. 하기에 설명되는 설정의 기준은 본문서 설정항목 부분의 설명을 참조한다.

Task정의

Container 정보를 포함하는 Task를 정의한다.

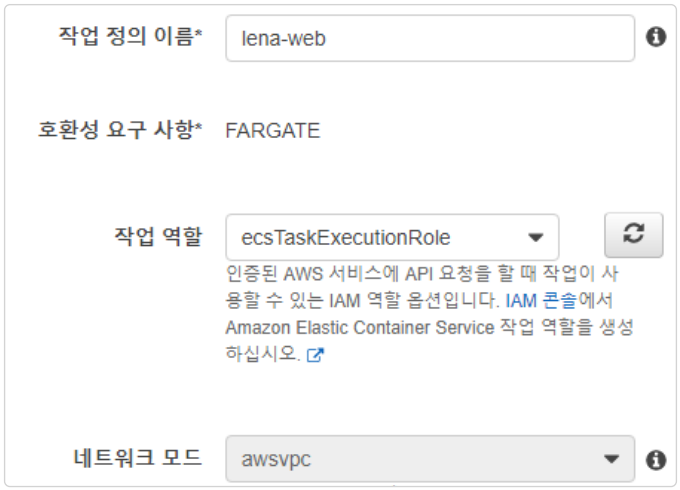


Figure 45. ECS기반 Web Server 설치 - Task 정의

Container 추가

이름, 이미지등의 기본 Container정보를 입력한다

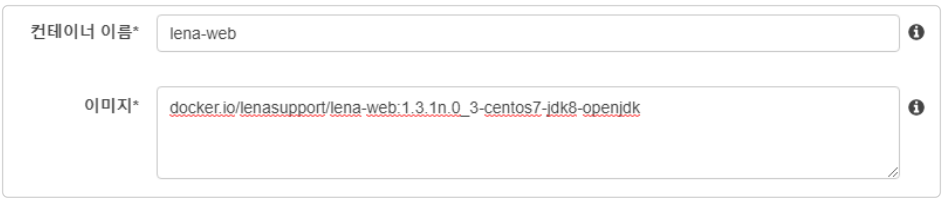


Figure 46. ECS기반 Web Server 설치 - Container 추가

환경변수 설정

Container 설정중 환경변수를 추가한다.

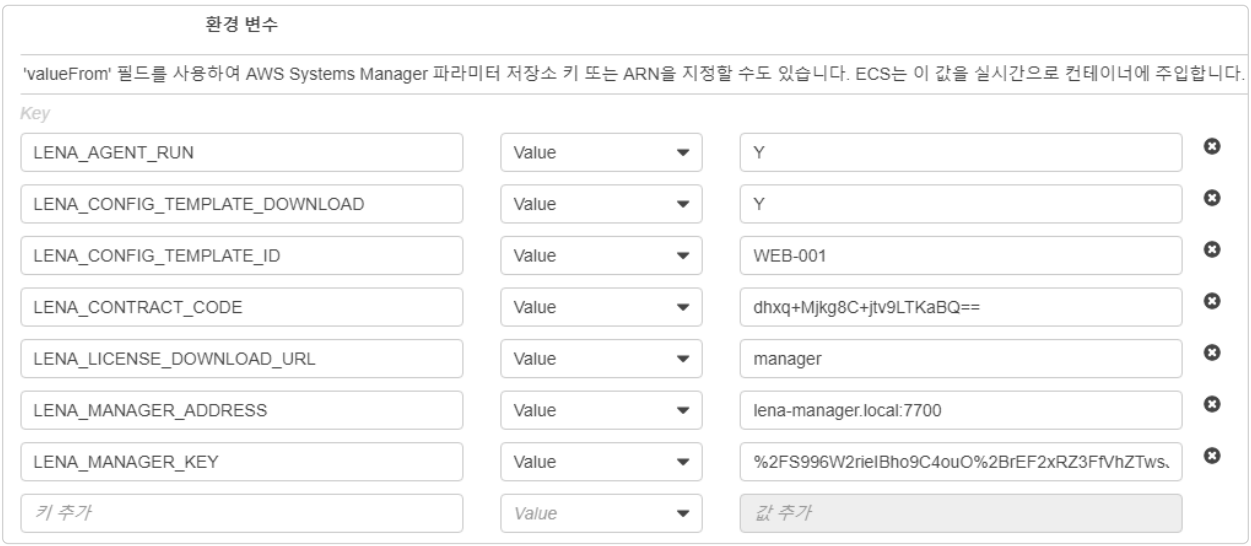


Figure 47. ECS기반 Web Server 설치 - 환경변수 설정

6.7.4. Service 설정

서비스 정의

앞서 정의한 Task를 실제 기동/운영하기 위한 Service를 정의한다.

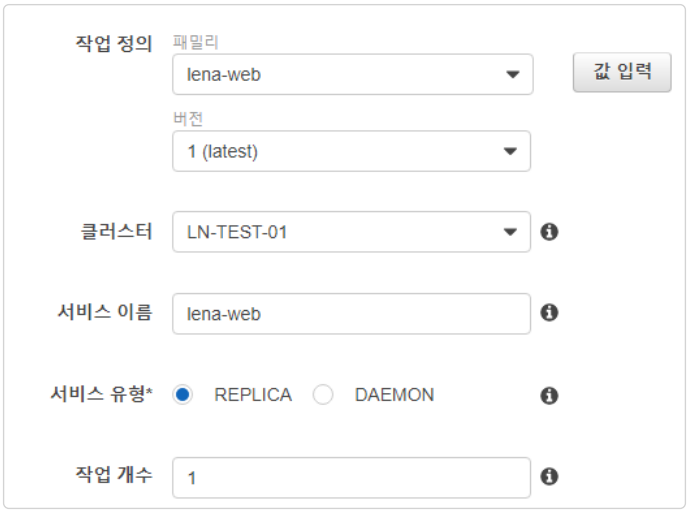


Figure 48. ECS기반 Web Server 설치 - Service 정의

서비스 검색 (Service Discovery) 설정

Web Server는 ELB 또는 Service Discovery 기능을 이용하여 고정된 주소를 확보하여 외부 또는 타 서비스에 Web 서비스를 제공할 수 있다.

Figure 49. ECS기반 Web Server 설치 - Service Discovery 설정

6.7.5. Service 기동 및 확인

Service 기동은 Service 정의를 저장하면 기동이 시작된다. ECS Cluster의 화면에서 서비스 및 작업 탭에서 운영중인 Service와 Task의 상태를 확인한다.

Service 상태 확인

Figure 50. ECS기반 Web Server 설치 - Service 상태 확인

Task 상태 확인

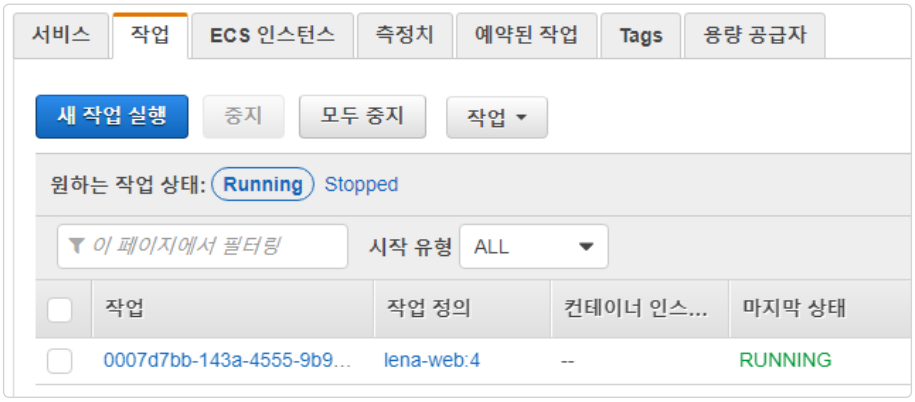


Figure 51. ECS기반 Web Server 설치 - Task 상태 확인

Chapter 7. VM/Host 기반 설치

7.1. 설치준비

설치 준비 작업으로 설치파일을 대상 서버에 업로드하고, Manager 및 Node Agent를 설치 및 실행한다. 이후 설치 작업은 Manager의 Web UI를 통해서 설치가 가능하며, 방화벽 등으로 Web UI 접속이 되지 않은 경우, 커맨드 라인으로도 동일하게 설치가 가능하다.

7.2. LENA 설치

LENA 설치파일은 gzip형식으로 제공되며, 설치 대상 서버에 업로드 후 설치 홈 디렉토리(${LENA_HOME})에 압축을 해제한다. 기본 설치 경로는 '/engn001/lena/1.3/'를 사용한다.

LENA 설치

```
[engn001]#
[engn001]# tar -xzvf lena-1.3.x.tar.gz
```

설치 모듈은 용도에 따라 다음과 같이 제공이 된다.

Table 16. LENA 설치 모듈 목록

Scripts	설명	비고
lena-[버전].tar.gz	Web/Application 통합 설치 파일로 Application Server, Web Server, Session Server 설치 모듈이 모두 포함	lena-1.3.x.tar.gz
lena-enterprise-[버전].tar.gz	Enterprise 버전의 WAS설치 모듈 Enterprise 버전에는 Session Server가 포함	lena-enterprise-1.3.x.tar.gz
lena-standard-[버전].tar.gz	Standard 버전의 WAS설치 모듈	lena-standard-1.3.x.tar.gz

7.3. 디렉토리 구성

LENA 설치를 위한 파일을 준비한다. LENA 설치 파일은 별도로 제공된다.

${LENA_HOME}의 디렉토리 구조는 아래와 같다.

Table 17. 디렉토리 구조

디렉토리	설명	비고
bin	Node Agent와 Manager의 Start/Stop scripts, install scripts 제공	
conf	Node Agent, Manager 등의 설정파일	

디렉토리	설명	비고
database	모니터링에서 생성한 일별 데이터를 저장하는 디렉토리	
depot	설치를 위한 Local Repository	
etc	기타 메타 정보 및 설정 파일	
license	License 정보를 관리하는 디렉토리	
logs	Node Agent / Manager 로그파일	
modules	실행에 필요한 모듈이 위치하는 경로 (lena-node-agent, lena-installer, lena-manager 등)	
servers	Server가 설치될 기본경로	
tmp	임시디렉토리	

제공하는 실행 Scripts 는 아래와 같다. (${LENA_HOME}/bin 에 위치)

Table 18. 제공 Script 목록

Scripts	설명	비고
install.sh	서버를 설치하기 위한 기본 script	
web-compile.sh	Web Server를 컴파일하기 위한 script	
web-package-install.sh	Web Server 컴파일 및 구동에 필요한 패키지 설치 script	Linux only, root 권한 필요
crypt.sh	Datasource에 사용하는 Password 수동 암호화 실행 (입력한 문자열을 암호화 문자열로 변환)	
env-manager.sh	Manager실행을 위한 환경변수	Manager 설치시
start-manager.sh	Manager의 실행	Manager 설치시
stop-manager.sh	Manager의 종료	Manager 설치시
ps-manager.sh	Manager의 프로세스 확인	Manager 설치시
start-agent.sh	Node Agent의 실행	
stop-agent.sh	Node Agent의 종료	
ps-agent.sh	Node Agent의 프로세스 확인	

환경설정 파일은 아래와 같다. (${LENA_HOME}/conf 에 위치)

Table 19. 환경설정 파일 목록

Config File	설명	비고
manager.conf	Manager 관련 설정	
agent.conf	Node Agent 관련 설정	

7.4. Manager 설치

제공되는 LENA는 Web Server, WAS와 Session Server, Node/Server에 설치되어 제어 및 Status를 확인하는 Agent와 관리자에게 제공되는 Manager도 구성된다.

7.4.1. Manager 설치

Manager는 install.sh을 이용하여 아래와 같은 순서도 설치한다.

1. ${LENA_HOME}/bin/install.sh create lena-manager
2. Service Port 정보를 입력한다. (default: 7700)
3. 서버 상태정보를 수신 받을 port 정보를 입력한다. 기본 설정을 사용하며, Manager를 추가도 설치하는 경우에는 port를 변경한다. (default: 16100)
4. Manager를 실행할 OS계정을 입력한다. (default: 스크립트 실행 유저)

LENA Manager 설치

```
[bin]$ ./install.sh create lena-manager
******************************
* LENA Server Install !       *
******************************
+-------------------------------------------------------------------------------
--------
| 1. SERVICE_PORT is the port number used by Manager.
|    ex : 7700
| 2. MONITORING_PORT is the port number used by Manager for monitoring.
|    ex : 16100
| 3. RUN_USER is user running Argo Manager.
|    ex : tomat
+-------------------------------------------------------------------------------
--------
Input SERVICE_PORT for installation. (q:quit)
Default value is '7700'

Input MONITORING_PORT for installation. (q:quit)
Default value is '16100'

Input RUN_USER for installation. (q:quit)
Default value is 'lena'

========================= Execution Result ========================
LENA_HOME : /engn001/lena/1.3
JAVA_HOME : /engn001/java/jdk1.8.0_191
SERVER_ID : lena-manager
SERVICE_PORT : 7700
MONITORING_PORT : 16100
INSTALL_PATH : /engn001/lena/1.3/modules/lena-manager
RESULT : Success
MESSAGE : create succeeded
===================================================================

Execution is completed.!!
[bin]$
```

여러 대의 장비도 서비스를 하는 경우, Manager는 한대의 장비에만 설치한다.

7.4.2. Manager 실행

Manager를 기동하여 정상적으로 설치되었는지 확인한다.

1. start-manager.sh 파일을 실행한다.

LENA Manager 실행

```
[bin]$ ./start-manager.sh
-------------------------------
            LENA Manager
-------------------------------
Using LENA_HOME :   /engn001/lena/1.3
Using JRE_HOME :    /engn001/java/jdk1.8.0_191
Using SERVER_PID :  /engn001/lena/1.3/modules/lena-manager/lena-
manager_solmanager.pid
Using SERVER_HOME : /engn001/lena/1.3/modules/lena-manager
Using SERVER_ID :   lena-manager
Using INSTANCE_NAME : lena-manager_solmanager
LENA started.
[bin]$
```

2. http://[Manager IP]:7700 에 접속하여 아래 페이지를 확인한다.(초기값: admin/!admin1234)

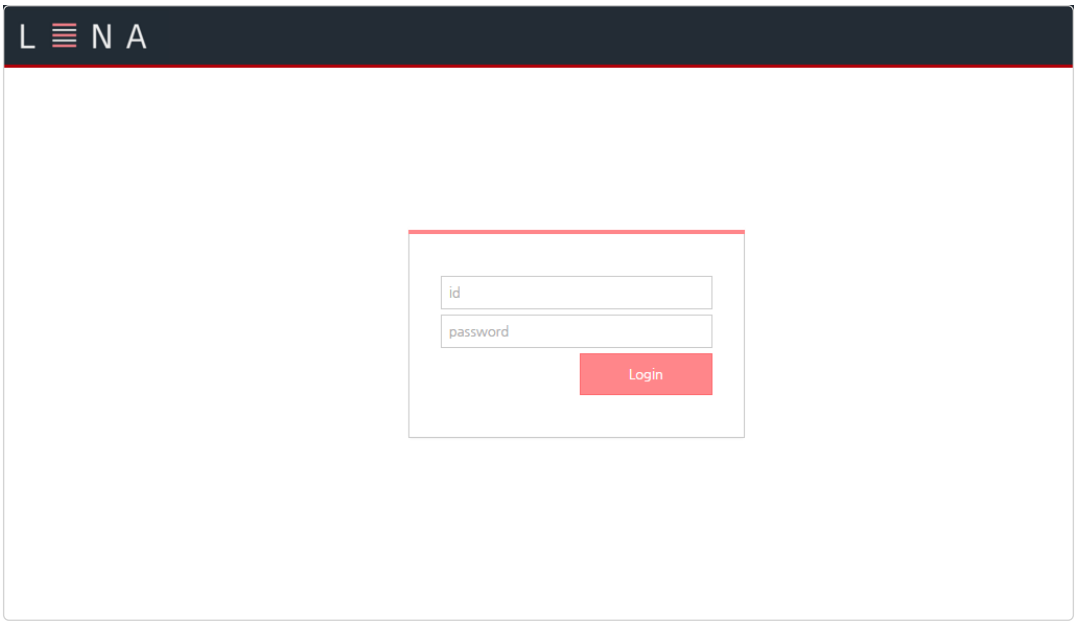


Figure 52. LENA Manager 로그인

3. stop-manager.sh 파일은 실행하여 종료 할 수 있다.

LENA Manager 종료

```
[bin]$ ./stop-manager.sh
--------------------------------
           LENA Manager
--------------------------------
Using LENA_HOME : /engn001/lena/1.3
Using JRE_HOME: /engn001/java/jdk1.8.0_191
Using SERVER_PID: /engn001/lena/1.3/modules/lena-manager/lena-
manager_solmanager.pid
Using SERVER_HOME : /engn001/lena/1.3/modules/lena-manager
Using SERVER_ID : lena-manager
Using INSTANCE_NAME : lena-manager_solmanager
LENA stopped.
##### lena-manager_solmanager successfully shut down ######
[bin]$
```

7.5. Node Agent 실행

Node Agent는 Node, Server의 제어 및 모니터링 기능을 담당하는 Agent 이다. Node Agent는 LENA설치시 기본적으로 설치가 되며, Node에 대한 정보를 가져오기 위한 Agent를 실행하여야 한다. Node Agent는 Web/Application/Session Server의 상태조회 및 시작과 종료를 수행 할 수 있다.

7.5.1. Node Agent 실행

${LENA_HOME}/bin/start-agent.sh 파일을 실행한다. JAVA_HOME이 지정되지 않은 경우, terminal에서 JAVA_HOME을 입력하라는 메시지가 나오게 된다. 이때, JAVA_HOME의 경로를 입력하면 agent가 실행된다.

```
[bin]# ./start-agent.sh
Input JAVA_HOME path for LENA. ( q: quit )
JAVA_HOME PATH :
/engn001/java/jdk1.8.0_191
Input Agent port for LENA Agent. ( q: quit )
Agent port (Default : 16800):
Input Agent user for LENA Agent. ( q: quit )
Agent user (Default : root):
root

--------------------------------
            LENA Agent
--------------------------------
Using LENA_HOME : /engn001/lena/1.3
Using JAVA_HOME : /engn001/java/jdk1.8.0_191/jre
Using CONF_FILE : /engn001/lena/1.3/conf/agent.conf
Using LOG_HOME : /engn001/lena/1.3/logs/lena-agent
Using RUN_USER : root
Using PORT : 16800
Using UUID : d03ddd60-de12-35df-9ea1-a409a3085eeb
LENA Agent is started.
[bin]#
```

7.5.2. Node Agent 동작 여부 확인

${LENA_HOME}/bin/ps-agent.sh 파일은 실행하여 아래와 같이 Process의 상태를 확인한다.

```
[bin]$ ./ps-agent.sh
lena     24208      1 62 14:00 ?      00:00:03
/engn001/java/jdk1.8.0_191/bin/java -Xms64m -Xmx256m
-Dlena.home=/engn001/lena/1.3 -Dlog.home=/engn001/lena/1.3/logs/lena-agent
-Dpatch.log.home=/engn001/lena/1.3/logs/lena-patcher
-Djava.library.path=:/engn001/lena/1.3/modules/lena-agent/lib/sigar
-Djava.net.preferIPv4Stack=true -cp .::/engn001/lena/1.3/modules/lena-
agent/lib/bcprov-jdk15on-1.55.jar:/engn001/lena/1.3/modules/lena-
agent/lib/lena-agent-1.3.0.jar:/engn001/lena/1.3/modules/lena-
agent/lib:/engn001/java/jdk1.8.0_191/lib/tools.jar
argo.node.agent.server.NodeAgentServer -start
[bin]$
```

7.5.3. Node Agent 종료

stop-agent.sh를 실행하여 종료할 수 있다.

```
[bin]$ ./stop-agent.sh
--------------------------------
            LENA Agent
--------------------------------
Using LENA_HOME : /engn001/lena/1.3
Using JAVA_HOME : /engn001/java/jdk1.8.0_191/jre
Using CONF_FILE : /engn001/lena/1.3/conf/agent.conf
Using LOG_HOME : /engn001/lena/1.3/logs/lena-agent
Using RUN_USER : lena
Using PORT : 16800
Using UUID : 0d5f6a4a-1084-4bac-ad8c-70b67bf3e495
LENA Agent is stopped normally.
[bin]$
```

7.6. Session Server 설치 (WEB UI 기반)

Session Server를 관리하기 위한 화면은 제공한다. Node에 설치한 Session Server의 등록, 수정, 삭제가 가능하며, 시작과 종료 Shell을 실행할 수 있다.

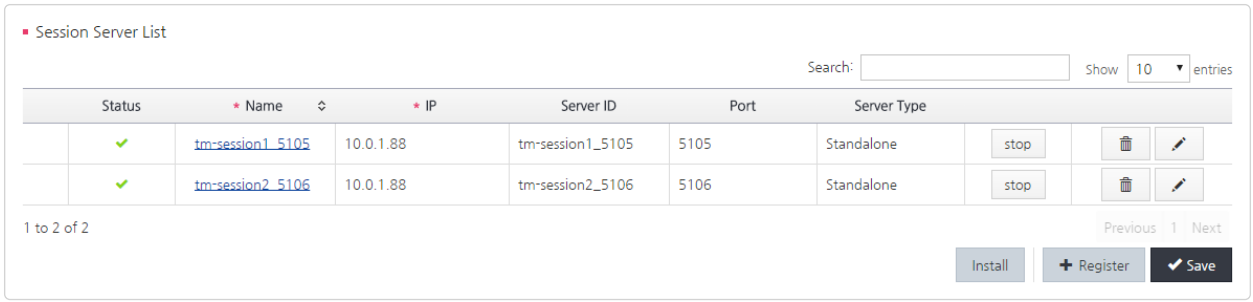


Figure 53. Session Server 목록

Session Server의 속성은 아래와 같다. (*) 는 필수값

Table 20. Session Server의 속성

항목	설명	비고
Status	Session Server의 상태	
Name(*)	Session Server의 이름	
IP(*)	Session Server의 IP주소	
Server ID	Session Server의 Identifier	
Port	Service 포트번호	
Server Type	Session Server의 유형	
Start/Stop	Server의 시작 및 종료	
+ 아이콘	**Register 버튼** 또는 **수정(연필) 버튼** 은 클릭하여 선택된 Server 정보가 변경 중임을 표시	
- 아이콘	**삭제(휴지통) 버튼** 은 클릭하여 선택된 Server정보가 삭제됨을 표시	

7.6.1. Session Server 설치

1. **Install 버튼** 을 클릭한다.

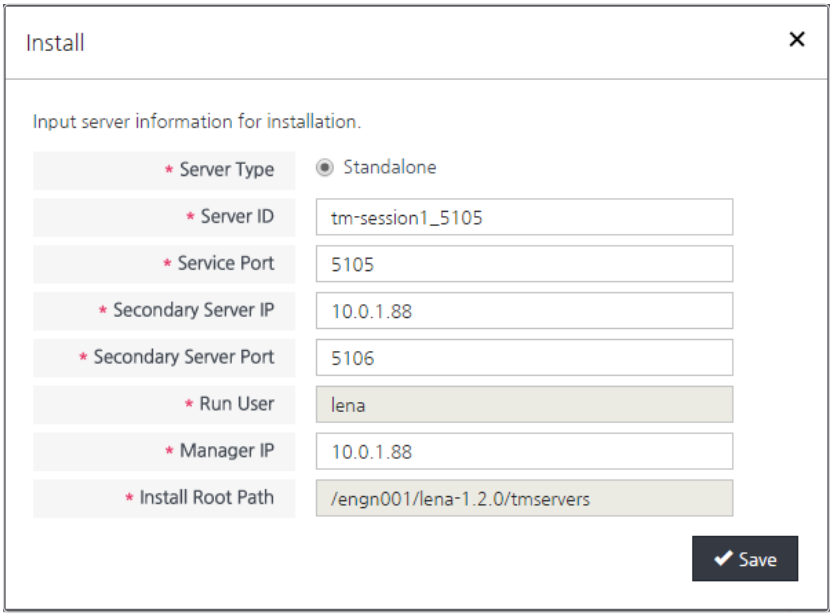


Figure 54. Session Server 설치시 입력 화면

2. Server ID와 Service Port, Secondary Server IP/Port를 입력한다.
3. 'Save' 버튼을 클릭하여 저장한다.

- Node에 실제 설치되어 있는 서버와 Manager에서 관리하는 서버의 정보에는 차이가 있을 수 있다. (console기반 설치 시)
- 서버ID중복오류가 발생하는 경우, Register기능을 이용하여 설치된 서버정보를 확인한다.
- Manager IP는 Node의 host IP로 자동 입력된다. 네트워크 구성에 따라 자동 입력된 IP가 실제 네트워크 IP와 다른 경우가 발생할 수 있다. 이때는 Manager IP를 수정하여 입력해야 한다.

7.6.2. Server 실행

1. **Stop 버튼** 을 클릭하여 Server를 종료한다.
2. **Start 버튼** 을 클릭하여 Server를 시작한다.

시작 가능한 상태일 경우에만 시작버튼이 활성화 된다.

7.6.3. Server 삭제

1. **삭제(휴지통) 버튼** 을 클릭하여 Server정보를 삭제 가능한 상태로 변경한다.
2. **Save 버튼** 을 클릭한다.
3. OK버튼을 누르면 Manager의 DB데이터와 물리적 서버를 완전히 삭제하고, Cancel버튼을 클릭하면 Manager의 DB 데이터만 삭제한다.

7.6.4. Server 등록

Console 기반으로 설치한 서버를 Manager를 통해서 관리하려면, Server 정보 등록이 필요하다.

1. **+Register 버튼** 은 클릭한다.
2. 등록할 서버를 클릭한다.

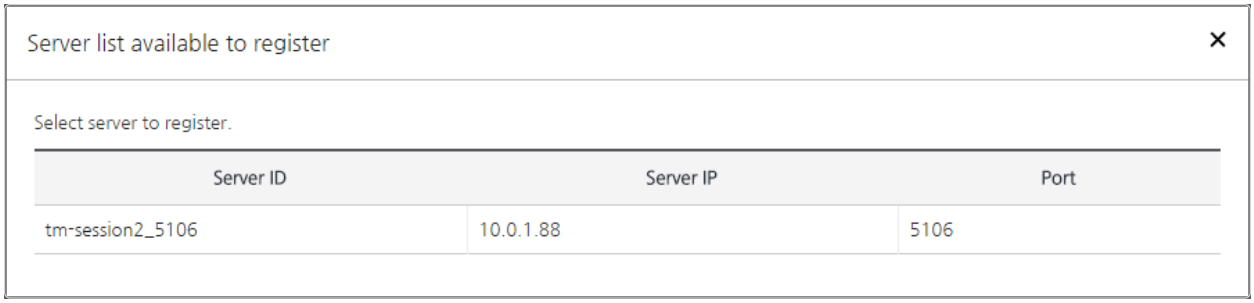


Figure 55. Session Server 등록시 Server 선택 화면

3. **Save 버튼** 은 클릭하여 저장한다..

7.7. Session Server 설치 (CLI 기반)

7.7.1. Session Server 설치

Session Server는 Embedded와 Standalone버전으로 구분된다. Embedded 버전의 경우 Application 서버 내에 포함되어 있어 별도 설치가 필요 없으며, Standalone 버전 설치 시 install.sh을 이용하여 아래와 같은 순서로 설치한다.

1. ${LENA_HOME}/bin/install.sh create lena-session

Session Server 설치

```
[bin]$ ./install.sh create lena-session
*******************************
* LENA Server Install !       *
*******************************
+-------------------------------------------------------------------------------
--------
| 1. SERVER_ID means business code of system and its number of letter is
from 3 to 5.
|    ex : tom1, tc01, svr01
| 2. SERVICE_PORT is the port number used by Session Server.
|    ex : 8080
| 3. SECONDARY_SERVER_IP is the ip number communicate with Secondary Session
Server
|    ex : 127.0.0.1
| 4. SECONDARY_SERVICE_PORT is the port number used by Secondary Session
Server.
|    ex : 8080
| 5. RUN_USER is user running Session Server
|    ex : tomat, apahe
| 6. INSTALL_ROOT_PATH is is server root directory in filesystem.
|    ex : /ssw, /sw/server, /ssw/was
+-------------------------------------------------------------------------------
--------
```

2. 입력 항목
   - 항목별로 default값이 표시되며, 변경이 필요한 경우 사용자가 직접 입력하여 변경할 수 있다.

Session Server 설치시 입력 항목 예시

```
Input SERVER_ID for installation. (q:quit)
tm-session1
Input SERVICE_PORT for installation. (q:quit)
Default value is '5000'
5005
Input SECONDARY_SERVER_IP for installation. (q:quit)
127.0.0.1
Input SECONDARY_SERVICE_PORT for installation. (q:quit)
Default value is '5001'
5006
Input RUN_USER for installation. (q:quit)
Default value is 'lena'

Input INSTALL_ROOT_PATH for installation. (q:quit)
Default value is '/engn001/lena-1.3.0/tmservers'

 ======================== Execution Result ======================
LENA_HOME : /engn001/lena/1.3
JAVA_HOME : /engn001/java/jdk1.8.0_191/jre
SERVER_ID : tm-session1
SERVICE_PORT : 5005
SECONDARY_SERVER_IP : 127.0.01
SECONDARY_SERVICE_PORT : 5006
RUN_USER : lena
INSTALL_PATH : /engn001/lena/1.2/servers/session1
RESULT : Success
MESSAGE : create succeeded
 ================================================================

create is completed.!!
[bin]$
```

Table 21. Session Server 설치시 입력 항목

항목	설명	비고
SERVER_ID	Session Server의 ID	
SERVICE_PORT	Session Server의 서비스포트	Default: “5000”
SECONDARY_SERVER_IP	Secondary Server의 IP주소	
SECONDARY_SERVICE_PORT	Secondary Server의 서비스포트	Default: “5001”

항목	설명	비고
RUN_USER	Session Server를 실행하는 실행 계정명	Default: “스크립트 실행 계정”
INSTALL_ROOT_PATH	Session Server를 설치할 상위 디렉토리	Default: “${LENA_HOME}/tmservers”

- $INSTALL_ROOT_PATH/tmservers/”SERVICE_ID” Directory생성을 확인한다.

install.sh 수행 시 하나의 Session Server가 설치되며, N 개의 서버 설치 시 install.sh은 N회 수행해야 한다.

7.7.2. Session Server 실행

Session Server를 기동하여 정상적으로 설치되었는지 확인한다.

1. Session Server 설치 위치에서 start.sh 파일을 실행한다.

Session Server 기동

```
[tm-session1]$ ./start.sh
--------------------------------
Start Session Server
--------------------------------
Using LENA_HOME : /engn001/lena/1.3
Using SERVER_HOME : /engn001/lena/1.3/servers/tm-session1
Using SERVER_ID : tm-session1
Using JAVA_HOME : /engn001/java/jdk1.8.0_191

Session Server Started..
[tm-session1]$
```

2. ps.sh 파일을 실행하여 프로세스의 상태를 확인한다.

Session Server 프로세스 상태 확인

```
[tm-session1]$ ./ps.sh

lena 16232     1     1 09:56 pts/7 00:00:00
/engn001/java/jdk1.8.0_191/bin/java -Xmx1024m -Dzodiac.name=session_5105
-Dzodiac.logdir=/engn001/lena/1.3/logs/session-server -cp
.::/engn001/lena/leesyong/1.2/servers/tm-session1/lib/lena-session-common-
1.2.0.jar:/engn001/lena/leesyong/1.2/servers/tm-session1/lib/lena-session-
server-1.2.0.jar -Dzodiac.config=session.conf zodiac.server.Main

[tm-session1]$
```

Session Server 종료

3. stop.sh 파일은 실행하여 종료할 수 있다.

```
[tm-session1]$ ./stop.sh
--------------------------------
Stop Session Server
--------------------------------
Using LENA_HOME : /engn001/lena/1.3
Using SERVER_HOME : /engn001/lena/1.3/servers/tm-session1
Using SERVER_ID : tm-session1
Using JAVA_HOME : /engn001/java/jdk1.8.0_191

Session Server Stoped..
[tm-session1]$
```

7.7.3. Session Server 삭제

기 설치된 서버는 스크립트를 이용하여 Uninstall할 수 있다.

LENA에서는 설치된 서버의 정보를 별도의 xml파일에 저장하고 있다. 따라서, directory를 직접 삭제하지 않고, install.sh 스크립트를 이용하여 Uninstall 해야 한다.

1. install.sh 스크립트 실행
    - Session Server : ${LENA_HOME}/bin/install.sh delete lena-session
    - Manager : ${LENA_HOME}/bin/install.sh delete lena-manager

Session Server 삭제

```
[lena@RNDTOMCAT1V bin]$ ./install.sh delete tm-session
******************************
* LENA Server Install !       *
******************************
+-------------------------------------------------------------------------------
--------
| 1. SERVER_ID : Server'id to delete
+-------------------------------------------------------------------------------
--------
Input SERVER_ID for installation. (q:quit)
tm-session
========================= Execution Result ======================
LENA_HOME : /engn001/lena/1.3
JAVA_HOME : /engn001/java/jdk1.8.0_191/jre
SERVER_ID : lenawas2
DELETE_PATH : /engn001/lena/1.3/servers/tm-session
RESULT : Success
MESSAGE : delete succeeded
=================================================================

delete is completed.!!

[bin]$
```

2. 입력 항목

Table 22. Session Server 삭제시 이력 항목

항목	설명	비고
SERVER_ID	Uninstall할 Server의 ID	Manager의 경우 id가 lena-manager도 자동입력 되며, 별도로 Server ID를 입력 받지 않는다.

LENA에서는 설치된 서버의 정보를 별도의 xml파일에 저장하고 있다. 따라서, directory를 직접 삭제하지 않고, install.sh 스크립트를 이용하여 Uninstall해야 한다.

Chapter 8. 별첨

8.1. LENA 지원 Spec별 버전

Table 23. LENA 지원 Spec

Specification	Version	비고
Java Development Kit (JDK)	1.8~	
Java Servlet	3.1	
Java Server Pages (JSP)	2.3	
Expression Language (EL)	2.2	
JavaServer Pages Standard Tag Library (JSTL)	1.2	
Enterprise JavaBeans (EJB)	3.2	
Java Message Service (JMS)	1.1	
Java Transaction API (JTA)	1.2	
Java API for RESTful Services (JAX-RS)	2.0	
Java API for XML Web Services (JAX-WS)	2.2	

8.2. Manager DB파일 백업

Manager의 내부데이터 관리를 위한 HSQL DB의 파일은 주기적으로(1일) 백업파일을 생성하고 있다. 생성위치는 ${LENA_HOME}/repository/backup/lena-manager/script 이다.

기본적으로 30일 이전 백업정보는 삭제하도록 되어 있는데 보관기간을 변경하고 싶은 경우, ${LENA_HOME}/conf 폴더 하위에 manager.conf 파일을 열고, dbbackup.size=보관기간 을 입력 후 Manager를 재 기동하면 보관기간을 변경할 수 있다.

8.3. Manager 의 내부이력 삭제

Manager가 내부적으로 남기는 이력은 주기적으로 삭제하도록 스케쥴링이 되어 있다. 삭제하는 정보는 Action Trace 이력과 Server History 이력이다.

기본적으로 Action Trace이력은 30일까지만 보관하고, Server History 이력은 90일까지 보관하고 있다. 이 보관기간을 변경하고 싶은 경우 ${LENA_HOME}/repository/conf 폴더 하위에 manager.conf 파일을 열고, actiontrace.size=보관기간, serverhistory.size=보관기간을 입력 후 Manager를 재 기동하면 보관기간을 변경할 수 있다.

8.4. Manager 의 admin 패스워드 초기화

Manager의 admin사용자 패스워드를 분실하거나 비밀번호 오듀횟수가 초과하였을 경우에는 console를 통하여 패스워드를 초기화해야 한다.

1. Manager가 설치된 장비에 console(telnet or ssh)로 접속한다.
2. $LENA_HOME/bin/reset_manager_pw.sh 파일을 실행한다.

3. 패스워드를 초기화 할 user인 admin을 입력한다.
4. 초기화할 패스워드를 입력한다. 단, 패스워드는 8자리이상, 알파벳/숫자/특수문자의 조합으로 입력한다. 패스워드는 보안을 위해 console에 표시되지 않는다.

Manager의 admin 패스워드 초기화

```
[bin]$ ./reset-manager-pw.sh
******************************
* LENA Server Install !       *
******************************
+-------------------------------------------------------------------------------
--
| 1. USER_ID is the user id to reset
| ex : admin
| 2. NEW_PASSWORD is the password to change
| - password rule #1 : more than 8 length
| - password rule #2 : inclusion of one or more alphabet characters
| - password rule #3 : inclusion of one or more numerical digits
| - password rule #4 : inclusion of one or more special characters
+-------------------------------------------------------------------------------
--
Input USER_ID for installation. (q:quit)
administrator
Input NEW_PASSWORD for installation. (q:quit)

The password has been changed successfuly.
Execution is completed.!!s
```

8.5. LENA 설치 권장 OS파라미터(CentOS기준)

LENA 설치 시 OS파라미터는 max user processes 값은 8192이상으로 설정하는 것은 권장한다.

Table 24. 권장 OS파라미터 (CentOS 기준)

parameter	권장값	기본값
max user processes	8192	1024
open files	8192	1024

CentOS기준으로 max user processes 설정은 다음과 같이 'ulimit –a' 명령어를 실행하여 확인을 할 수 있다.

OS 파라미터 max user processes 확인 (CentOS 기준)

```
$ ulimit -a +
core file size          (blocks, -c) 0 +
data seg size           (kbytes, -d) unlimited +
scheduling priority             (-e) 0 +
file size               (blocks, -f) 8192 +
pending signals                 (-i) 14891 +
max locked memory       (kbytes, -l) 64 +
max memory size         (kbytes, -m) unlimited +
open files                      (-n) 1024 +
pipe size            (512 bytes, -p) 8 +
POSIX message queues     (bytes, -q) 819200 +
real-time priority              (-r) 0 +
stack size              (kbytes, -s) 10240 +
cpu time               (seconds, -t) unlimited +
max user processes              (-u) 1024 +
virtual memory          (kbytes, -v) unlimited +
file locks                      (-x) unlimited
```

CentOS를 기준으로 명령어 'ulimit –u'와 'ulimit –n'로 프로세스 수와 오픈파일 개수를 설정할 수 있다. 위 변경사항을 영구적으로 반영하기 위해서는 각 유저의 profile (.profile, .bash_profile)에 ulimit 실행명령을 추가하거나, 강제 설정할 수 있다 (CentOS 기준).

OS 파라미터 설정 - 프로세스 수 및 오픈파일 개수 (CentOS 기준)

```
$ cat $HOME/.bash_profile*
.. (생략)*
ulimit -u 8192*
ulimit -n 8192*
```

또 다른 설정 방법으로는 /etc/security/limits.conf (CentOS 기준) 파일을 열어서 프로세스 최대수(nproc)와 오픈파일 최대수(nofile)를 설정한다.

OS 파라미터 설정 - 프로세스 수 및 오픈파일 개수 (CentOS 기준)

```
$ cat /etc/security/limits.conf*
.. (생략)*
*     soft nproc 8192*
*     hard nproc 8192*
*     soft nofile 8192*
*     hard nofile 8192*
```

8.6. LENA 주기적으로 증가하는 파일

Table 25. 주기적으로 증가하는 파일

항목	경로	삭제주기	월 예상 증가량	비고
Manager정기점검도깅	LENA_HOME/repository/monitoringDB	N/A	10MB ~ 120MB	
Manager모니터딩, 진단리포트	LENA_HOME/repository/monitoringDB	7일	N/A	자동삭제
Manager진단동계	LENA_HOME/repository/monitoringDB	영구	1MB 이하	
Manager백업파일	LENA_HOME/repository/backup/lena-manager	영구	300MB 이하	
Manager Server 템픈릿	LENA_HOME/repository/container	영구	10MB / Service Cluster	Service Cluster 개수에 따라 판단
Manager도그	LENA_HOME/logs/lena-manager	30일	10MB ~ 100MB	
Agent도그	LENA_HOME/logs/lena-agent	30일	N/A	자동삭제
Installer도그	LENA_HOME/logs/lena-installer	영구	1MB 이하	
서버인스턴스도그	서버인스턴스설치경도 LENA_HOME/servers/server_id/logs	영구	부하에 따라 판단	경도변경가능

8.7. WAS Image OS 참조자료

타 WAS 솔루션의 이미지는 다음과 같은 OS 륻 사용한다.

Table 26. 타 WAS 솔루션 이미지 사용 현황 (2020년 기준)

WAS Image	OS Image
jboss/**wildfly**	**centos**:7
open-**liberty**:full-java8-openj9	**debian**:buster (from adoptopenjdk/openjdk8-openj9)
store/oracle/**weblogic**:12.2.1.4	**Oracle Linux**
ibmcom/**websphere**-traditional	**ubuntu**:16.04
tomcat 기본 이미지 예) tomcat:9-jdk8	기본 Image Tag는 openJdk 기본 Tag FROM openjdk:8-jdk (FROM **debian**:buster)

연락처

- **담당부서** : LG CNS 시스템솔루션사업팀
- **주소** : 서울특별시 강서구 마곡중앙10로 10 엘지사이언스파크 E13 [07796]
- **전화** : (02) 2099-6136
- **이메일** : lena-support@lgcns.com