적합성 검증 시나리오

과제명 : 클라우드 엣지 기반
도시교통 브레인 핵심기술 개발

2022.08.10

㈜이노그리드

개정 이력

버전	개정일자	개정 내역	작성자	검토자	승인자
0.1	22.04.25	개정	박형탁	김바울	구원본
0.5	22.06.08	업데이트	박형탁	김바울	구원본
1.0	22.08.10	업데이트	박형탁	김바울	구원본

목차

1. 개 요

o 이 장에서는 도시교통 브레인 시스템의 전체 시스템 구성과 사업 개요, 사용자 구분 등에 대하여 설명한다.

1.1. 전체 시스템 구성

o 도시교통 브레인 시스템은 대도시의 교통소통 최적화를 위해, 클라우드-엣지 기반 실시간 교통상황 분석 및 대규모 교통 시뮬레이션 분산처리를 통한 교통제어 지능을 제공하는 도시교통 브레인 시스템 개발을 목적으로 한다.

그림 1. 클라우드 엣지 기반 도시교통 브레인 개념도

1.2. 사용자 구분

o 도시교통 브레인 사용자는 다음과 같이 구분한다.

① 도시교통 브레인 시스템 관리자

- 도시교통 최적화에 활용할 수 있도록 교통 데이터 수집, 도시교통 시뮬레이션, 교통 흐름 예측 등의 도시교통 브레인 서비스를 제공하는 사람
- 도시교통 데이터 관리, 시뮬레이터 관리, 도시교통 브레인 사용자 관리, 인프라 자원 관리 등의 도시교통 브레인에 대한 전반적인 관리를 수행하는 사람

② 도시교통 브레인 시스템 사용자
- 도시교통 최적화와 관련한 이해 관계가 있는 최종 사용자로 도시교통 혼잡을 완화하기 위해 도시교통 정책, 신호 체계 등을 검증하려는 사람
- 도시 교통 브레인을 활용한 교통 혼잡 예측, 교통 수요 예측 등을 통해 도시 교통계획을 수립하려는 사람
- 예, 교통 전문가, 교통 정책 연구자 등

③ 도시교통 브레인 서비스 개발자
- 도시 교통망을 이용하는 사용자에게 편의를 제공하기 위해 도시교통 브레인을 활용하여 응용 서비스를 개발하는 사용자

1.3. 3차년도 사업 개요

o 3차년도 사업 목표는 클라우드 엣지 관리 플랫폼 요소기술 개발이며 [표 1]는 세부 사항을 나타내며, 내용은 다음과 같다.

○ 도시교통 브레인 클라우드 엣지 관리 플랫폼 기술 개발 ✓ 클라우드 엣지 인프라 및 엣지 단말 모니터링 기술 개발 ✓ 클라우드 엣지 인프라 통합 관리를 위한 대시보드 기술 개발
○ PaaS 개발자를 위한 인터페이스 및 대시보드 개발 ✓ PaaS 개발자 편의성을 지원하는 서비스 포털 UI/UX 개발 ✓ PaaS 서비스를 이용량 모니터링 및 시각화
○ 도시교통 브레인 클라우드 엣지 테스트베드 안정화 및 고도화 ✓ 안정적인 서비스 운영을 위한 클라우드 엣지 관리 플랫폼 이중화 설계 및 개발 ✓ 테스트베드 성능향상 및 고도화를 위한 운영 결과 분석 및 시스템 설계 확장 ✓ 네트워크 이중화 기반 안정적인 도시교통 브레인 인프라 환경 제공을 위한 설계 및 구축
○ 도시교통 브레인 클라우드 엣지 관리 플랫폼 서비스 적합성 검증 ✓ 도시교통 브레인 인프라 관리 서비스 이용 시나리오 도출 ✓ 실제 서비스 적용을 위한 도시교통 브레인 인프라 관련 추가 요구사항 조사 및 분석 ✓ 주요 시나리오별 도시교통 브레인 클라우드 엣지 관리 플랫폼 서비스 적합성 검증
○ 주요 개발 결과물 공개화 추진 ✓ 도시교통 브레인 인프라 관리 플랫폼 ✓ 도시교통 브레인 시스템 자원 제공을 위한 테스트베드 ✓ 도시교통 브레인 인프라 서비스 적합성 검증 보고서

표 1. 3차년도 사업 목표

1.4. 검증 목표

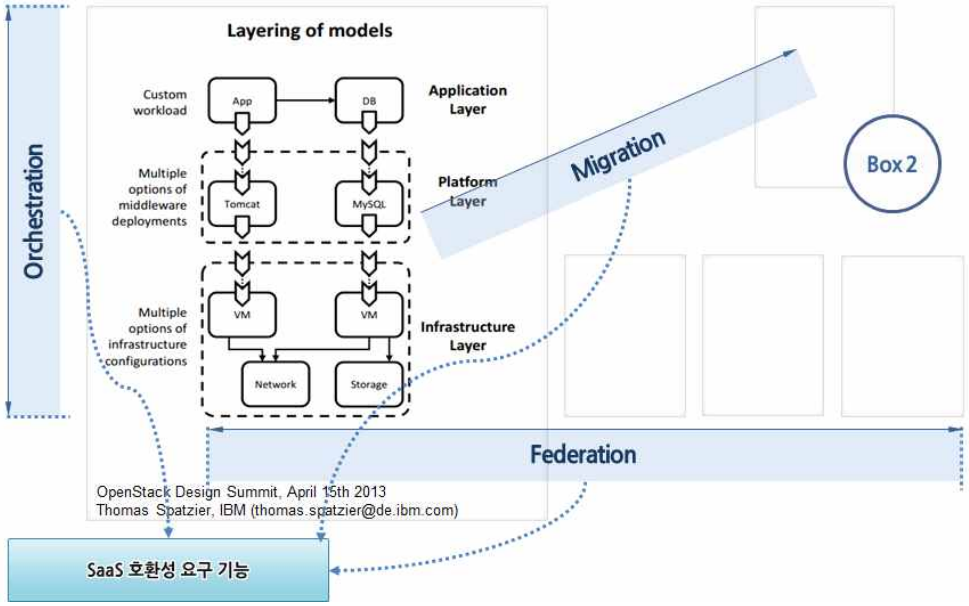
Cloud Edge Mgmt는 이종 다수 클라우드 인프라 플랫폼이 제공하는 다양한 형식의 전산 자원(P: 물리서버, V: 가상서버, C: 콘테이너)을 운용함에 따라 상호운용성(Interoperabililty), 호환성(Compatibility), 이식성(Portability) 문제를 해결해야 한다. 따라서 Cloud Edge Mgmt의 주요 기능을 상호운용성, 호환성, 이식성 관점에서 검증하는 방법 및 절차를 정의한다.

[표 2] Cloud Edge Mgmt 검증 항목

No.	검증항목	설명
1	상호 운용성 (Interoperability)	상호 운용성은 인터페이스가 어떤 제한된 접근이나 수행 없이, 다른 제품이나 시스템과 일할 수 있도록 완전히 이해되는 제품 또는 시스템의 속성
2	호환성 (Compatibility)	호환성(Compatibility)은 만약 모델 중의 하나를 작동하는 어떤 소프트웨어가 다른 모든 모델 집합 또한 작동할 수 있다면 컴퓨터 모델 집합끼리는 호환되는 것
3	이식성 (Portability)	이식성(Portability)은 원시 프로그램을 다른 기종으로 옮기기가 얼마나 쉬운가를 나타내는 정도로 기술/사업자 종속(lock-in)이 없음을 의미

1.5. 제약사항 및 적용방법

Cloud Edge Mgmt 기술의 검증은 이종 클라우드 및 엣지 클라우드 환경에서 호환성 보장에 요구되는 핵심기능에 대해 검증을 정의할 수 있지만 현재 이에 대한 공인된 테스트방법론 또는 테스트 사례 등은 보고되지 않은 실정이다. 이에 OASIS TOSCA (Topology and Orchestration Specification for Cloud Application) 사양의 개념모델을 중심으로, Cloud Edge Mgmt 구현 기능을 검증하는 절차 및 방법을 정의한다.



[그림 2] SaaS 호환성 요구기능

Cloud Edge Mgmt는 미리 정의된 자동화 절차(Work Flow)에 따라 소프트웨어를 전개하는 Orchestration, 이종 전산자원으로 논리적 클러스터를 구성하는 Federation, 운영 중인 소프트웨어 또는 데이터를 다른 박스로 옮기는 Migration을 지원한다. 기술된 기능은 각각 상호운영성, 호환성, 이식성을 통해 검증할 수 있다.

[그림 3] 주요기능 별 검증 항목

1.6. 검증시나리오

일반적으로 클라우드 컴퓨팅의 상호운용성 또는 호환성의 보장을 위하여 해결해야 하는 기술적 이슈는

1) 인프라 구성의 기반기술 요소로써 서버 가상화를 위한 하이퍼바이저의 활용,
2) 클라우드 내 보관자료 및 수행중 작업의 이동 또는 전환,
3) 이기종 클라우드 플랫폼에 대한 통합관리

등을 들 수 있다. Cloud Edge Mgmt는 DevOps Tower에서 자동화된 절차에 따

라 위 기술된 이슈를 해결하고자 함에 따라 일반적 소프트웨어 테스트 방법론 상의 점검방법을 그대로 적용하기에 무리가 있다. 이에 아래와 같은 검증 시나리오를 정의하고 장기적 관점에서 이를 통해 Cloud Edge Mgmt의 기능을 검증하고자 한다.

[표 3] Cloud Edge Mgmt 검증 시나리오

No.	검증항목	Cloud Edge Mgmt 기능	시나리오
1	상호운용성	Orchestration	Workflow 정의 절차에 따라 SaaS 응용이 정상적으로 전개되는가?
2	호환성	Federation	소프트웨어가 이종 다수 클라우드 환경에서 동시에 운영이 되는가?
3	이식성	Migration	현재 운영중인 소프트웨어 또는 데이터가 다른 Box로의 이관이 자동화 절차에 따라 수행되는가?
4	기능	Operation	이행되는 소프트웨어 스택 별 제공되는 모니터링 정보가 적절한가?

1.7. 검증시나리오 별 테스트케이스 정의

Cloud Edge Mgmt의 주요 점검 기능으로써 Orchestration, Federation, Migration 기능을 제공하는 소프트웨어 및 관련 기술 사양을 중심으로 향후 점검 항목으로써 테스트케이스를 도출한다. 따라서 기술 참조 사례를 포함하여 검증 시나리오의 이해를 돕는다. 본 문서에서 정의되는 검증시나리오 및 테스트케이스는 분석 단계의 초안으로써 향후 과제 진행에 따라 변경될 수 있다.

2. 검증시나리오 이행 방안

서비스 적합성 검증을 위하여 Cloud Edge Mgmt의 구현 기능을 검증하기 위한 시나리오 별 Test Case를 정의한다.

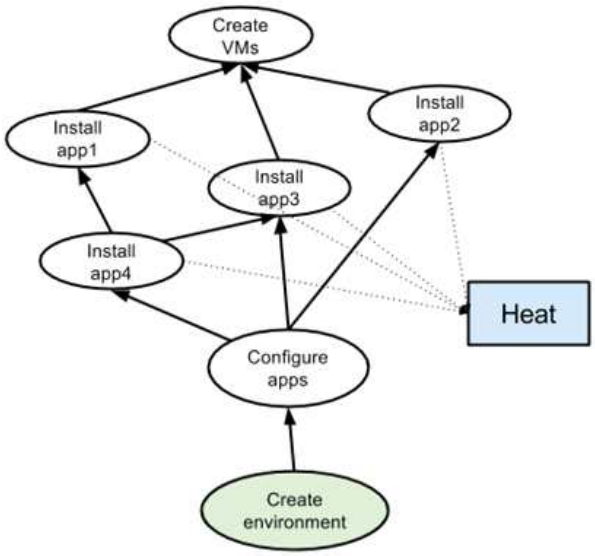
2.1. 시나리오 1 - Workflow 정의 절차에 따른 SaaS 응용 전개 여부

소프트웨어 전개에 요구되는 절차 및 태스크를 정의하고, 정해진 절차에 따라 소프트웨어가 정상적으로 전개되는지 점검한다. Cloud Edge Mgmt는 SaaS 전개에 요구되는 개별 소프트웨어의 종속관계(Dependency), 설치 순서(Flow) 등을 정의하는 환경을 제공하고, 여기에서 작성된 그래프 또는 스크립트에 따라 SaaS 응용이 전개되는지 점검한다.

[표 4] Test Case : Work Flow와 상호운용성

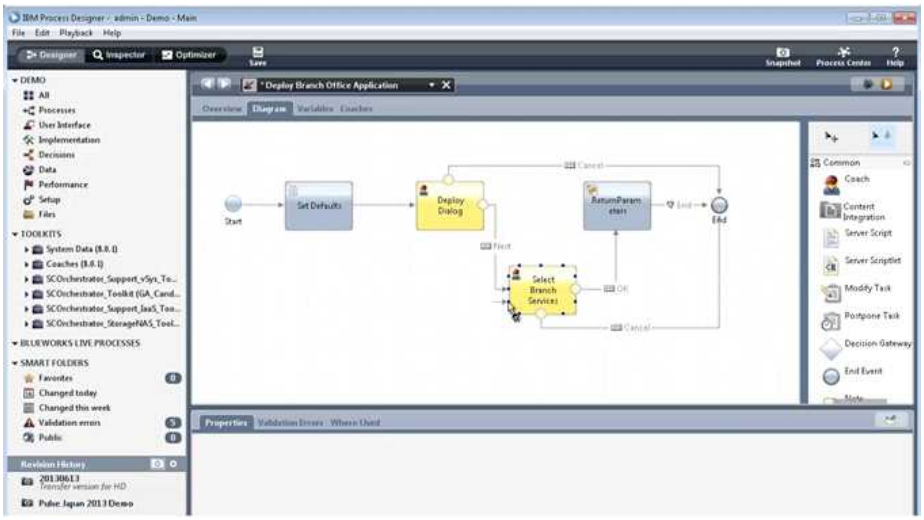
No.	Test Case	설명	확인
1	Workflow 작성	논리적 클러스터 위에 전개해야 하는 소프트웨어의 의존성 관계 및 설치 순서를 스크립트 또는 그래픽 UI를 통해 지정	
2	Workflow 유효성 확인	미리 정의된 소프트웨어의 의존성 관계 등에 문제가 없는지, 스크립트의 구문 상에 오류가 없는지 설치 이전 사전 점검	
3	SaaS 응용 전개	미리 정의된 작업절차 스크립트를 실행하여 소프트웨어가 지정된 VM 위에 정의된 설정이 제대로 반영되어 설치되었는지 점검	

(사례: Openstack Mistral) OpenStack의 Mistral은 Orchestration을 위한 Workflow 관리 도구로써, 기존 Heat에서 정의하지 못했던 Dependency Management 기능을 제공한다.



[그림 4] Cloud 환경 전개 정의

(사례: IBM Cloud Orchestrator) IBM® Cloud Orchestrator는 OASIS TOSCA(Topology and Orchestration Specification for Cloud Applications)에 따라 서비스 템플리트 가져오기, 배치 및 내보내기를 지원한다. IBM SoftLayer, 기존 Openstack 플랫폼, PowerVM, VMWare 또는 Amazon EC2로 서비스를 제공한다.



[그림 5] IBM Process Designer

2.2. 시나리오 2 - 이종 다수 클라우드 환경에서 소프트웨어 운영

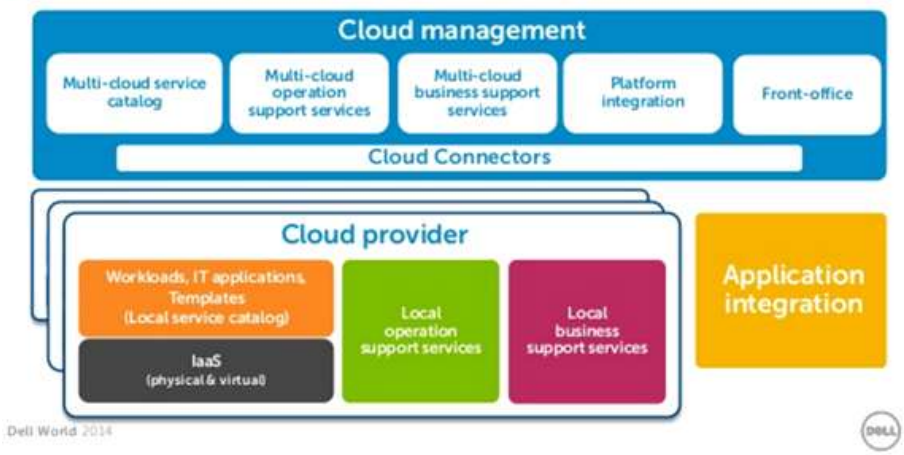
하나의 소프트웨어 스택이 둘 이상의 클라우드 환경에서 제공되는 전산자원 위에서 정상적으로 실행되는지 여부를 점검한다.

IaaS 영역인 UnderCloud를 통합하고 UnderCloud의 주요 기능은 둘 이상의 클라우드 서비스 제공자가 제공하는 전산자원의 상호연동, 프라이빗 클라우드와 퍼블릭 클라우드가 제공하는 전산 자원의 상호 연동을 통해 논리적으로 클러스터를 구성하는 것이다.

[표 5] Test Case : Federation 및 UnderCloud Composable Node

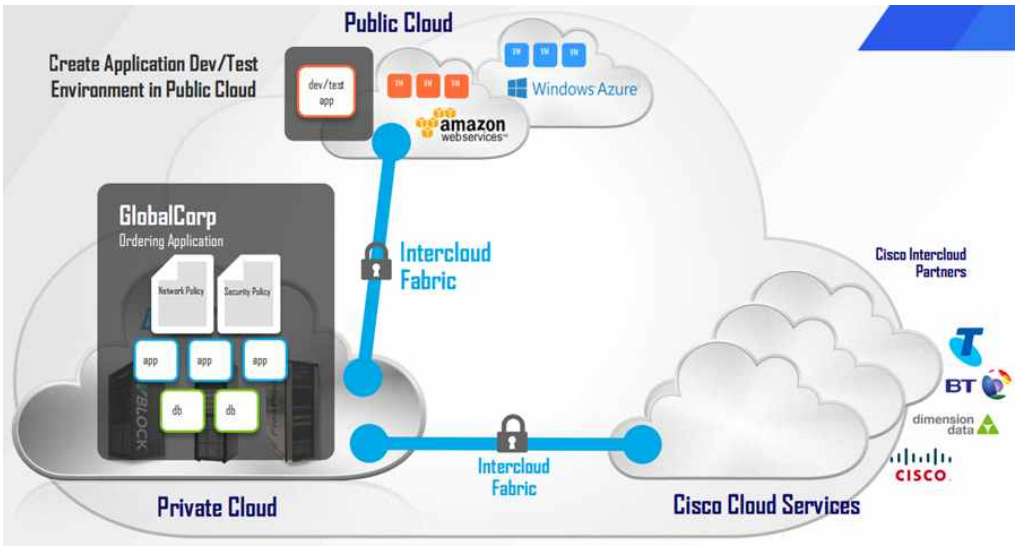
No.	Test Case	설명	확인
1	단일 클라우드 환경에서 논리 클러스터 구성	단일 클라우드 환경에서 자동화 절차에 따라 소프트웨어가 정상적으로 설치되어 배포되는지 여부 점검	
2	멀티 클라우드 환경에서 논리 클러스터 구성	이종 클라우드(하이브리드, 멀티클라우드) 환경에서 자동화 절차에 따라 소프트웨어가 정상적으로 설치되어 배포되는지 여부 점검	

(사례: 멀티클라우드) 멀티클라우드란 다양한 클라우드서비스를 사용하는 기술을 의미한다. 둘 이상의 Public IaaS 및 CSP 가 제공하는 On-Demand 관리와 서비스를 이용하는 것이며, 이종 다수 클라우드를 동시에 사용함에 있어 DevOps Tower를 통해 상호운용성 또는 호환성 문제를 해결한다. DevOps Tower는 전산자원의 구성을 위하여 내부적으로 UnderCloud를 통합한다.



[그림 6] MultiCloud Architecture (Dell World 2014)

(사례: 인터클라우드) 시스코와 시스코 파트너들은 독립적인 클라우드들을 상호 연결하기 위해 전 세계적으로 연결된 클라우드 네트워크인 인터클라우드를 구축하고 있다. 단일 클라우드서비스가 제공하는 실제 물리적 가용능력은 무한하지 않고, 지역적으로 모든 곳에 서비스를 제공할 수 없는 한계를 가진다. 이런 문제를 해결하기 위한 전제조건으로써 클라우드서비스 간의 호환성이 있어야 되며, 클라우드 간의 표준화된 인터페이스 및 네트워크 프로토콜에 대한 표준화 등의 작업이 요구된다. 시스코는 인터클라우드 패브릭을 통해 위 문제를 해결하고자 한다.



[그림 7] 인터클라우드 패브릭 아키텍처
(Cisco Connect, Seoul, Korea, 2015)

2.3. 시나리오 3 – 이종 클라우드 간 서비스 및 데이터 이관

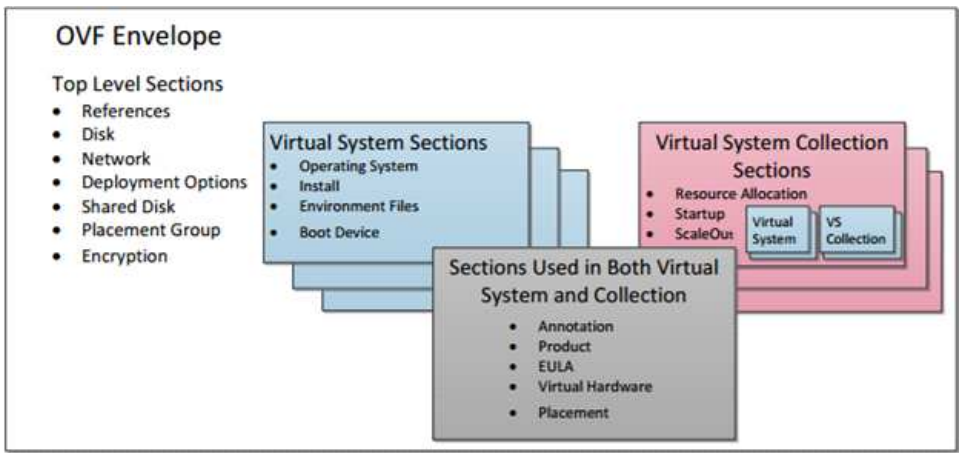
운영 중인 소프트웨어 및 데이터가 다른 클라우드 환경으로 정상적으로 이관되는지 여부의 점검한다. 특히 UnderCloud의 경우 고속 마이그레이션 기능과 초고속 대용량 데이터 교환 기능을 연계한 통합된 Composable UnderCloud 관리 기술 확보를 목표로 하고 있다. 이에 일반 상황에서의 마이그레이션 및 DTN 환경에서의 Migration이 정상적으로 이행되어 운영중인 소프트웨어의 Portability가 확인되는지 여부를 점검한다.

[표 6] Test Case : Migration

No.	Test Case	설명	확인
1	시스템 이관	논리적 클러스터에 포함되어 있는	

		전산 자원 (물리/가상/컨테이너)을 다른 클라우드 환경으로 이관	
2	데이터 이관	운영 중인 소프트웨어(예:Wordpress)가 생성한 운영계 데이터, 운영로그 등 데이터가 이종 클라우드로 정상적으로 이관되는지 여부를 점검	
3	DTN 노드 이용	DTN이 허용하는 용량 범위 내에서 1, 2번 항목이 정상적으로 수행되는지 여부 점검	

(사례: OVF) DMTF(Distributed Management Task Force)는 가상화 플랫폼 간 가상머신의 배포 및 이동성을 보장하기 위한 표준화된 메타데이터 모델을 정의해 나가고 있다. OVF(Open Virtualization Format)는 가상 어플라이언스의 패키징 및 배포를 위한 개방형 포맷으로 플랫폼 독립적이며, 확장가능하게 정의되어 있다. OVF는 가상 어플라이언스를 위한 메타데이터 표현 형식으로 XML을 사용한다.



[그림 8] OVF package structure

(사례: KREONET) 이기종간 다양한 그리드 어플리케이션 데이터의 전송/저장 및 활용을 위한 DTN (Data Transfer Node) 기술이 사용되고 있다. KREONET (Korea Research Environment Open NETwork : 국가 과학기술연구망)은 한국과학기술정보연구원이 관리·운영하는 국가 R&D 연구망이며, 이를 활용하여 Under Cloud 기반의 효과적인 초고속 데이터 교환을 위한 기술을 적용한다.

2.4. 시나리오 4 - 소프트웨어 유형 별 모니터링 정보 제공

이종 클라우드 환경이 제공하는 전산자원을 활용하여 논리적으로 클러스터를 구성한 후, 다양한 유형의 소프트웨어 스택을 이행한다. 이행되는 소프트웨어의 종류에 따라 운영상 요구되는 모니터링 항목은 달라질 수 있으며, 이들 모니터링 항목을 적절하게 제공하고 있는지 여부를 점검한다.

[표 7] Test Case : Migration

No.	Test Case	설명	확인
1	logical Cluster 구성 정보 제공	이종 클라우드가 제공하는 전산자원을 종류에 상관없이 조합하여 논리적으로 정의되는 클러스터를 구성하는지 여부 점검	
2	소프트웨어 운영 정보 제공	이종 클라우드 환경에서 다양한 소프트웨어 스택의 전개를 지원하는 DevOps Tower로써 소프트웨어 스택에 상관없이 동일한 운영정보 조회를 지원하는지 여부를 점검	

(사례: ELK) ELK Stack은 Elasticsearch, Logstash, Kibana의 약자로서 Elasticsearch는 Apache의 Lucene을 바탕으로 개발한 실시간 분산 검색 엔진이며, Logstash는 각종 로그를 가져와 JSON형태로 만들어 Elasticsearch로 전송하고 Kibana는 Elasticsearch에 저장된 Data를 사용자에게 Dashboard 형태로 보여주는 Solution이다.

ElasticSearch는 Apache Lucene을 기반으로 하는 분산 검색엔진이지만, 그 자체로 파일을 저장하는 NoSQL 형식의 데이터베이스이다. 소프트웨어 스택 운영 중 기록되는 다양한 로그 또는 상태정보를 일관된 방법으로 관리하도록 하는 운영 편의성을 제공한다.

```
filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
%{SYSLOGHOST:syslog_hostname}
%{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd
HH:mm:ss" ]
    }
  }
}
```

[그림 9] logstash의 syslog 메시지 추출 필터 설정

3. 클라우드를 통해 생성되는 3-Tier 소프트웨어의 가용성 검증

클라우드를 통해 생성되는 3-Tier 소프트웨어의 가용성 검증을 위한 도구, 검증 환경 및 검증 방법을 설명한다. 정보통신산업진흥원의 “클라우드컴퓨팅서비스 품질·성능 안내서”는 클라우드 서비스가 일정 수준의 품질 기준을 만족하고 있는지 확인하기 위한 절차 및 방법을 제공한다. 향후 다양한 분야(예: 3-Tier Web Service, IoT, HPC 등)에 적용될 수 있는 공통의 검증 방법에 대한 정의를 위하여 안내서가 제공하는 품질항목 중 “가용성” 부문에 대한 점검을 수행하고, 도출된 결과를 향후 추진계획 수립을 위한 기초자료로써 활용한다.

3.1. 가용성 검증 방법 및 도구

3-Tier SaaS 서비스에 대한 가용성 검증 측정도구는 Linux OS에서 실행 가능한 ping, nc(netcat), cURL을 주기적으로 SaaS Application에 대한 접근성 분석을 수행하여 이를 통해 가용성을 확인한다.

[표 8] 가용성 검증 방법

No.	검증항목	도구	설명
1	시스템 가용성 체크	ping	시스템의 접근성을 확인하는 가장 일반적이며 오버헤드가 없는 실행 방법은 ICMP(Internet Control Message Protocol)통신규약(RFC792 표준)을 따르는 ping 프로그램을 통해 지속적으로 대상 시스템에 접근을 시도하여 응답 결과에 오류가 있으면 시스템 장애로 판단함. ※ 검증 대상의 ICMP 프로토콜을 허용해 주어야 정상적인 측정이 가능함
2	서비스 포트 가용성 체크	nc (netcat)	클라우드서비스 상에서 이용자의 요청을 받아들이기 위해 대기하고 있는 서비스 포트로 접근을 시도함으로써 인터넷 프로토콜의 전송계층(Layer 4)을 통해 대상 서비스가 이용자 요청을 받아들일 수 있는 상태인지 확인할 수 있음. 포트 접속이 안되거나 응답에 오류가 있으면 서비스 장애로 판단함
3	웹사이트(URL) 가용성 체크	cURL	클라우드서비스 이용자가 서비스를 제공받기 위해 접근하는 웹사이트에 오류 없이 접근이 가능한지를 지속적으로 측정하여 가용성을 확인할 수

			있음. 웹사이트로 접근이 안되거나 HTTP 상태코드가 오류코드를 포함하는 경우를 서비스 장애로 판단함

3.2. 가용성 검증 시뮬레이션 환경 구성

이노그리드의 Cloudit 과 Amazon EC2가 제공하는 VM을 각각 1식씩 사용하여 가용성 검증 환경을 구성한다.

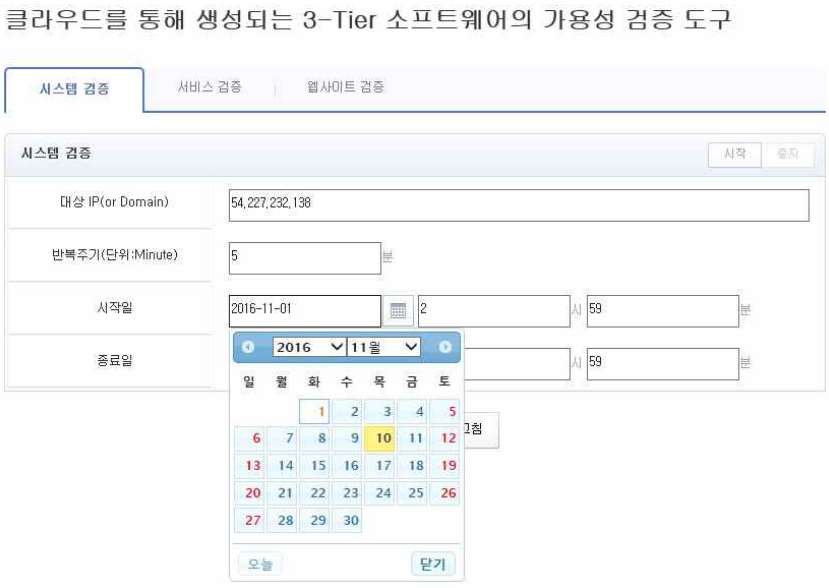
[표 9] Test Case : Federation 및 UnderCloud Composable Node

No.	용도	CSP	사양	설치 프로그램
1	검증 서버	이노그리드	1Processor 2GB Memory 100GB Storage	Apache Tomcat7 검증도구 nc(netcat) cURL
2	3-Tier SW	Amazon	1Processor 2GB Memory 100GB Storage	Wordpress

3.3. 가용성 검증 도구의 특징 및 주요 기능

(특징) 별도의 프로그램 설치 없이 웹브라우저로 손쉽게 접근하여 시스템 가용성, 서비스 포트 가용성, 웹사이트 가용성 등의 검증 스케쥴 관리가 가능하다. 검증 결과 로그파일은 /home/tomcat/log/ 디렉토리에 Verification.[yyyy-MM-dd].log 형식으로 날짜별로 구분하여 저장한다.

(주요기능) 시작, 중지 기능을 제공하여 시스템 검증 및 서비스 검증, 웹사이트 검증을 동시 혹은 부분 동작으로 선택 가능하다. 시작일과 종료일 설정으로 원하는 시간에 검증을 수행하는 예약기능을 제공한다. 반복 주기(분 단위) 설정을 통해 측정 빈도수를 조절할 수 있다.

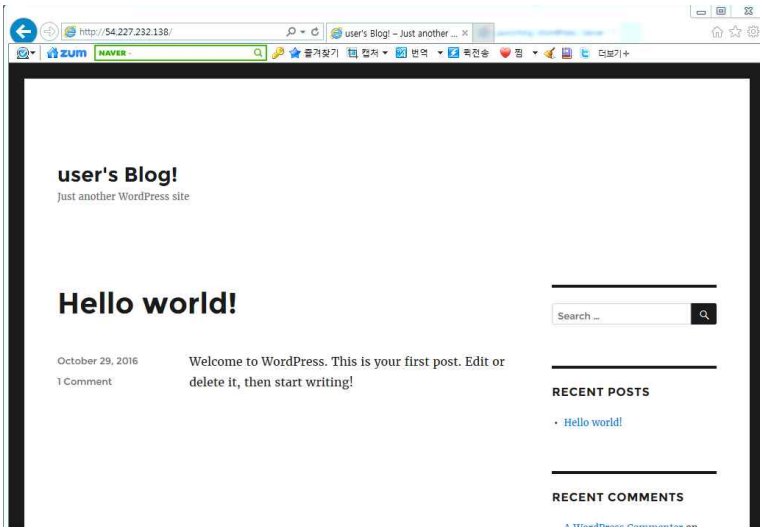


[그림 10] 가용성 검증도구 스케쥴 관리

3.4. 가용성 검증 도구 사용법

가용성 검증도구는 웹 어플리케이션으로써, 내부적으로 사용자 UI를 통해 가용성 측정 대상 소프트웨어 및 측정 주기 등의 설정을 등록한다.

1) 3-Tier 환경으로 생성된 Wordpress 서비스의 IP를 확인한다.



[그림 11] Wordpress 실행 화면

2) 웹 브라우저의 주소 창에 검증도구 URL을 입력하여 접근한다.
(URL - http://[IP or Domain]:8080/Verification)
3) 3종류(시스템, 서비스, 웹사이트)의 가용성 검증 항목 중 원하는 메뉴를

선택한다.

[그림 12] 시스템 검증 입력 양식

[그림 13] 서비스 포트 검증 입력 양식

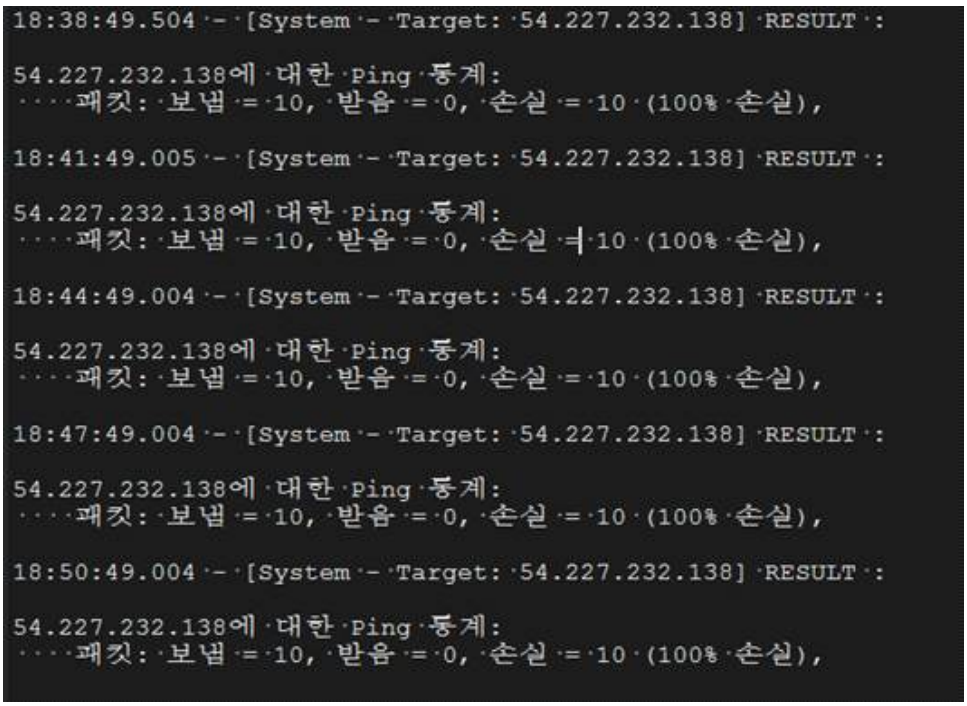
[그림 14] 웹 사이트 검증 입력 양식

4) 시스템 검증 및 서비스 검증, 웹사이트 검증의 각 입력폼을 채우고 저장 버튼을 클릭하여, 저장된 검증 스케쥴 중 시작을 원하는 서비스의 시작 버튼을 클릭한다.

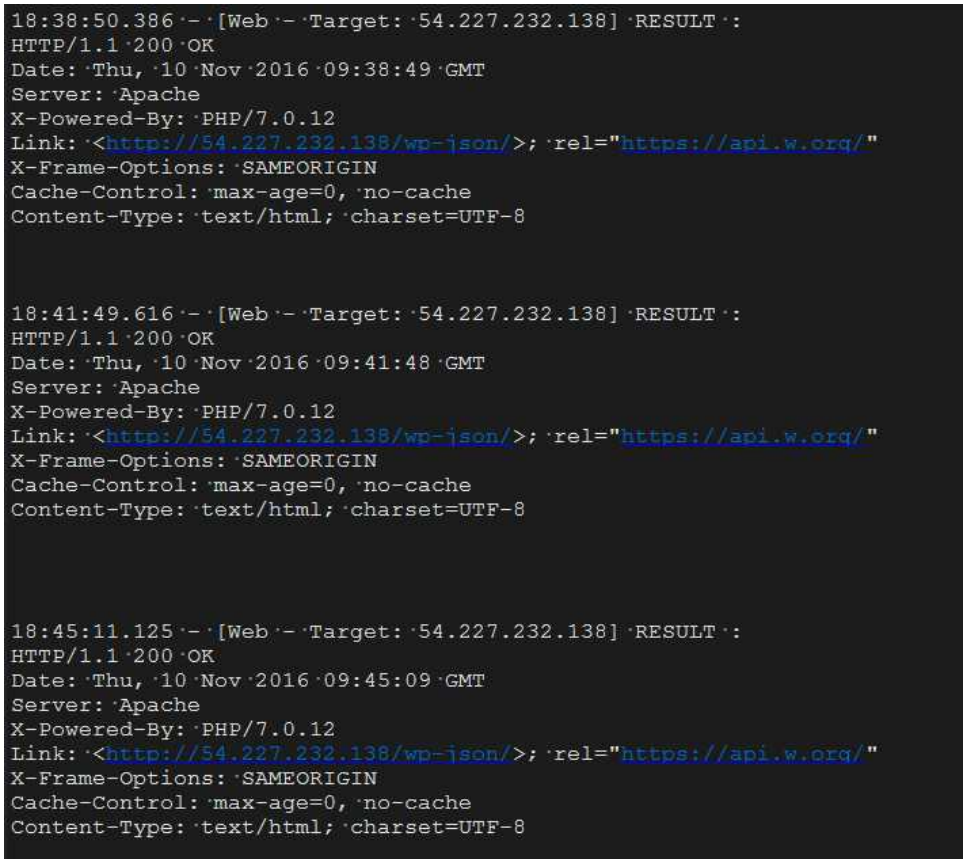
5) 측정 완료 후 검증결과 파일을 확인한다.

3.5. 가용성 검증 결과 분석

/home/tomcat/log 디렉토리에 저장된 Verification.log 파일로 직접 확인하거나 ELK(ElasticSearch, Logstash, Kibana)와 같은 로그분석 플랫폼으로 파일을 가공하여 확인 할 수 있다.



[그림 15] 시스템 가용성 검증 결과 로그파일 내용



[그림 16] 웹사이트 가용성 검증 결과 로그파일 내용