

ISOC 서비스 포털 자동 프로비저닝 기능 구현

한성근, 김종백, 신영호
 한국과학기술정보연구원
 e-mail:{sgghan, oprx3, shinyh}@kisti.re.kr

Implementation of Auto-provisioning for ISOC Service Portal

Sung-Geun Han, Joong-Baek Kim, Young-Ho Shin
 Korea Institute of Science and Technology Information

요 약

클라우드 컴퓨팅의 가상화된 인프라 기반 위에서 IT 자원 및 애플리케이션 서비스를 관리하고 배포하는 작업이 클라우드 프로비저닝이다. 본 논문에서는 IT 자원을 할당함에 있어 전문적인 지식을 가진 관리자의 개입을 최소화하기 위해 일련의 자원 할당 작업을 자동화하였다. 자동화된 자원 프로비저닝을 통해 관리자는 특정 전문 지식 없이도 자원 할당 및 배포가 가능해져 이용자의 요구를 보다 신속하게 처리할 수 있게 되었다.

1. 서론

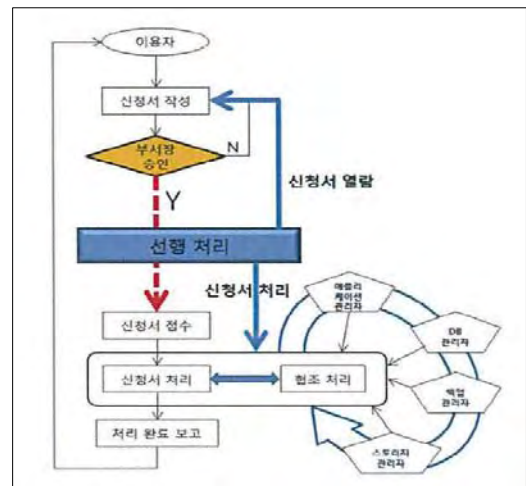
전기나 수도처럼 필요한 자원을 빌려 쓰고 사용한 만큼의 비용을 지불하듯이 서버, 스토리지, 네트워크, 응용 프로그램 등의 IT 자원도 빌려 쓰는 시대가 도래했다. 클라우드 컴퓨팅[1] 기술이 이러한 요구를 충족시키고 있다. 이용자들은 각자가 IT 자원을 직접 소유하거나 관리할 필요 없이 하드웨어 자원을 마치 소프트웨어처럼 서비스 받을 수 있는 체계가 마련된 것이다.

클라우드 컴퓨팅은 서비스 유형에 따라 서버, 스토리지, 네트워크 등 하드웨어 자원을 서비스하는 IaaS(Infrastructure as a Service), 미들웨어나 소프트웨어 개발에 필요한 플랫폼을 서비스하는 PaaS(Platform as a Service), 이용자에게 필요한 다양한 소프트웨어를 서비스하는 SaaS(Software as a Service)로 나눈다. 또한 서비스를 제공하는 형태에 따라 회사 내부에서 외부에 데이터를 공유하지 않고 내부 이용자들에게만 제공하는 프라이빗 클라우드(Private Cloud), 아마존 AWS[2], 구글의 앱 엔진[3] 등과 같이 불특정 다수에게 제공하는 퍼블릭 클라우드(Public Cloud), 프라이빗 클라우드와 퍼블릭 클라우드가 혼합된 하이브리드 클라우드(Hybrid Cloud)로 나눈다.

최근 KISTI에서는 기존의 물리서버들을 가상화 서버들로 이전 및 전환하였고, 가상화 인프라를 바탕으로 프라이빗 클라우드를 구축하였다. 또한 이용자들에게 구축된 자원을 서비스하기 위하여 ISOC 서비스 포털 홈페이지를 만들었다. ISOC 서비스 포털을 통해 원내 이용자들은 구축된 프라이빗 클라우드에서 제공하는 다양한 자원을 서비스 받을 수 있게 된 것이다.

클라우드 컴퓨팅의 가상화된 인프라(IaaS) 기반 위에서 IT 자원 및 애플리케이션 서비스를 관리하고 배포하는 작업이 클라우드 프로비저닝이다.[4] ISOC 서비스 포털 또한 이러한 클라우드 프로비저닝을 지원하기 위해 일부 자원 할당에 대해 자동화 기능을 구축하였다. 본 논문에서는 기존 ISOC 프로세스를 살펴보고, 클라우드 기반 하에서 구현된 자동 프로비저닝 기능에 대해 다룬다. 또한, 이용자가 홈페이지 구축에 필요한 자원을 요구했을 때 자동 프로비저닝을 위해 필요한 스크립트 및 처리 프로세스에 대해서 다룬다.

2. 기존 ISOC 프로세스



(그림 1) ISOC 프로세스

(그림 1)은 기존의 ISOC 프로세스를 나타낸다. 이용자

들은 ISOC 홈페이지를 통해 사용하고자 하는 IT 자원을 다양한 신청서를 통해 요구한다[5]. 신청서가 들어오면 전산 관리자들이 해당 요구사항에 맞게 IT 자원을 준비하여 제공한다. 신청서의 내용에 따라 전문 지식을 갖춘 다양한 관리자들이 필요하며 이러한 관리자들의 협업으로 다양한 요청이 처리된다. 즉, 전문 관리자가 없이는 IT 자원도 제대로 준비될 수 없는 문제가 있다.

3. 자동 프로비저닝 구현

이용자가 원하는 IT 자원을 제대로 제공하기 위해서는 시스템이 자동으로 해당 기능을 수행하여 어느 특정 전문가의 개입을 최소화하는 과정이 필요하다. 클라우드 체계가 이를 가능하게 하였으며, 기존의 ISOC 프로세스를 클라우드 체계로 개선하여 ISOC 서비스 포털을 만들었고, 일부 자원 할당 기능을 자동으로 제공할 수 있는 자동 프로비저닝(Auto-provisioning) 기능을 구현하였다.

본 논문에서는 ISOC 서비스 포털의 자동 프로비저닝 기능 중에서 가장 요구가 빈번한 홈페이지 제공용 IT 자원 할당 요구에 대한 자동 프로비저닝 방법을 다룬다.

3.1 홈페이지 자원 요구 내용

하나의 홈페이지 서비스를 제공하기 위해서는 다양한 요소의 IT 자원이 필요하다. 서버, 스토리지, 네트워크와 같은 하드웨어 인프라는 이미 클라우드를 통해 구축되어 있다고 가정하고, 필요한 소프트웨어를 설치하는 과정을 설명한다.

<표 1>은 홈페이지 자원 신청에 필요한 요구 내용을 나타낸다.

<표 1> 사용자 요구 내용

요구 내용	설 명
URL	홈페이지 서비스용 URL
서버 및 DB 계정	소스 및 개발용 서버 웹 계정, DB 계정 정보
개발 언어	홈페이지 개발용 언어 (C, Java, PHP 등)
웹 서버	Apache, WebToB 등
WAS	Tomcat, JEUS 등
DB	MySQL, MS-SQL, Oracle 등
스토리지	서비스용 디스크 및 DB용 디스크 용량
기타	DB charset 등

이용자들의 요구 내용에 따라 자동 할당에 필요한 패키지 구성 내용이 달라진다.

<표 2>는 자동 프로비저닝을 제공하기에 앞서 필요한 서비스 플랫폼용 IT 자원을 나타낸다. 표에서 나타내고 있는 IT 자원 구성요소가 미리 구축되어 있어야 자동 프로비저닝을 수행할 수 있다. 서비스용으로 미리 구성된

IT 자원에 대해 다양한 API를 사용하여 개발한 패키지가 바로 자동 프로비저닝 패키지다. 자동 프로비저닝을 구현하기 위해서는 자동화를 지원할 수 있는 서비스 플랫폼 구축이 선행되어야 한다.

<표 2> 서비스 플랫폼용 IT 자원

IT 자원	설 명
LDAP	서버 웹 계정 관리를 위한 디렉토리 서비스
웹 서버 애플리케이션	Apache, WebToB, Tomcat, Jeus 등 웹 서비스를 위한 서버 프로그램
DBMS	DB 계정 추가용 DBMS (MySQL, MS-SQL, Oracle 등)
공용 스토리지	서버 이중화 및 데이터 저장용 스토리지
IP Pool	IP 할당을 위한 네트워크 자원

3.2 자동 할당용 스크립트

ISOC 서비스 포털 자동 프로비저닝은 이용자가 제출한 신청서를 분석하여 어떠한 IT 자원을 프로비저닝할 지 결정하여 관리자용 작업 리스트를 생성한다. 관리자는 생성된 작업 리스트에 따라 프로비저닝을 수행하게 된다. 관리자가 수행하는 프로비저닝에는 애플리케이션 소스를 직접 컴파일하거나 바이너리 패키지를 이용하여 구성하는 패키지 설치 방법과 단순히 스크립트를 이용하여 일부 환경과 일을 수정하여 서비스를 구성하는 스크립트 설치 방법이 있다.

<표 3> 패키지 설치용 애플리케이션

구분	애플리케이션	설치방법
WEB	Apache	소스 컴파일
	WebToB	바이너리
WAS	Tomcat	바이너리
	Jeus	바이너리

<표 3>은 패키지 설치용 애플리케이션을 나타낸다. Apache는 배포되는 소스 파일들을 버전별로 관리하여 해당하는 OS에 맞게 컴파일하여 설치한다. 따라서, 프로비저닝시 컴파일에 소요되는 시간만큼 배포 시간이 길어진다. 그 외 다른 웹 애플리케이션들은 바이너리 패키지 형태로 제공되며 프로비저닝시에 지정한 위치에 복사되는 형태로 설치된다.

<표 4>는 특정 애플리케이션을 설치하는 것이 아니라 설치된 애플리케이션의 환경 파일을 수정하여 이용자가 원하는 서비스를 제공하기 위한 스크립트 작업을 나타낸다. 서비스 연속성을 지원하기 위해 서버들은 이중화로 구성된다. 이중화를 지원하기 위해 각 애플리케이션의 환경 설정 파일들이 수정되어야 하며 변경된 서버 IP 정보들이

반영되어야 한다.

<표 4 > 프로비저닝 스크립트

스크립트	설 명
Apache WebToB IIS	<ul style="list-style-type: none"> • 웹 서버용 환경 파일 관련 스크립트 • 관련 파일: httpd.conf ssl.conf http.m
Tomcat Jeus	<ul style="list-style-type: none"> • WAS용 환경 파일 관련 스크립트 • 수정 파일: catalina.sh server.xml JEUSMain.xml WEBMain.xml
MySQL MS-SQL Oracle	<ul style="list-style-type: none"> • DBMS 계정 추가용 스크립트 SQL을 사용하여 구성
IP Alias	<ul style="list-style-type: none"> • 웹 서버용 IP alias 사용을 위한 스크립트 • 수정 파일: network-scripts/ifcfg-*
Apache+Tomcat Apache+Jeus WebToB+Jeus	<ul style="list-style-type: none"> • WEB-WAS 연동용 스크립트 • 수정 파일: JkMount 모듈 (mod_jk.so) workers.properties WEBMain.xml
Tomcat+MySQL Tomcat+MS-SQL Tomcat+Oracle Jeus+MySQL Jeus+MS-SQL Jeus+Oracle	<ul style="list-style-type: none"> • WAS-DB JNDI 연동 (OCI, Thin) • 수정 파일: ROOT.xml JEUSMain.xml
LDAP	<ul style="list-style-type: none"> • 서버 쉘 계정 관리 스크립트
NAS Mount	<ul style="list-style-type: none"> • 공용 스토리지용 스크립트

3.3 자동 할당 처리 프로세스

3.2절의 스크립트를 통해 관리자는 특정 애플리케이션에 대한 전문 지식이 없어도 필요한 자원을 할당할 수 있게 되었다. 다음 내용은 이용자가 홈페이지를 개설하기 위해 요청한 내용에 대해 자동 프로비저닝으로 행해지는 일련의 작업을 나열한 것이다.

- LDAP (계정 추가) : 이용자의 소스 코드 업로드, 서비스용 데이터 업로드, 홈페이지용 홈디렉토리 등에 필요한 사용자 계정 추가
- NAS Mount : 데이터 저장 공간을 할당하고 할당된 계정으로 마운트하여 해당 계정이 사용 가능하게

한다.

- 웹 서버 설치 : Apache, WebToB 등 선택한 웹 서버 애플리케이션을 설치, URL 설정, 홈디렉토리를 설정한다
- WAS 설치 : Tomcat, Jeus 등을 설치, IP Pool을 이용한 IP 및 서비스 포트를 설정한다
- DB 계정 추가 : 선택한 DBMS 상에 이용자 계정을 추가하고 테이블스페이스를 할당한다
- WAS-DB 연동 : WAS와 DB 간에 JNDI를 설정하여 사용가능하게 한다.
- WEB-WAS, WAS-WEB 연동 : WEB, WAS 간 JkMount 설치 및 연동 서비스 포트를 설정하여 연동한다.
- 샘플파일 설치 : index.html, index.jsp, indexdb.jsp 등의 샘플 파일들을 설치하여, 요청한 홈페이지가 제대로 호출되는지 확인 할 수 있도록 한다.

최종적으로 이용자는 위에 설치한 서버 정보 및 환경 정보들을 피드백 받게 되고, DNS에 해당 URL을 등록함으로써 홈페이지 서비스를 받을 수 있게 된다.

4. 결론

클라우드 시스템에서 효율적인 IT 자원 할당을 위해서는 클라우드 프로비저닝이 필수적이다. 본 논문에서는 클라우드 기반 하에서 동작하는 자동 프로비저닝을 구현하였다. 이용자는 ISOC 서비스 포털을 통해 IT 자원을 요청하고, 관리자는 구현된 자동 프로비저닝을 통해 이용자에게 신속하게 IT 자원을 제공할 수 있는 체계를 마련하게 되었다.

참고문헌

- [1] Armbrust, Michael, et al. "A view of cloud computing." Communications of the ACM 53.4 (2010): 50-58.
- [2] Amazon Web Services: Elastic Load Balancing <http://aws.amazon.com/elasticloadbalancing/>
- [3] Zahariev, Alexander. "Google app engine." Helsinki University of Technology(2009).
- [4] Varia, Jinesh. "Best practices in architecting cloud applications in the AWS cloud." Cloud Computing: Principles and Paradigms (2011): 459-490.
- [5] 한성근, "정보서비스 요구 지원 시스템 개발", 2010 한국컴퓨터종합학술대회 논문집, Vol.37, No.1(B), pp.401-405, 2010