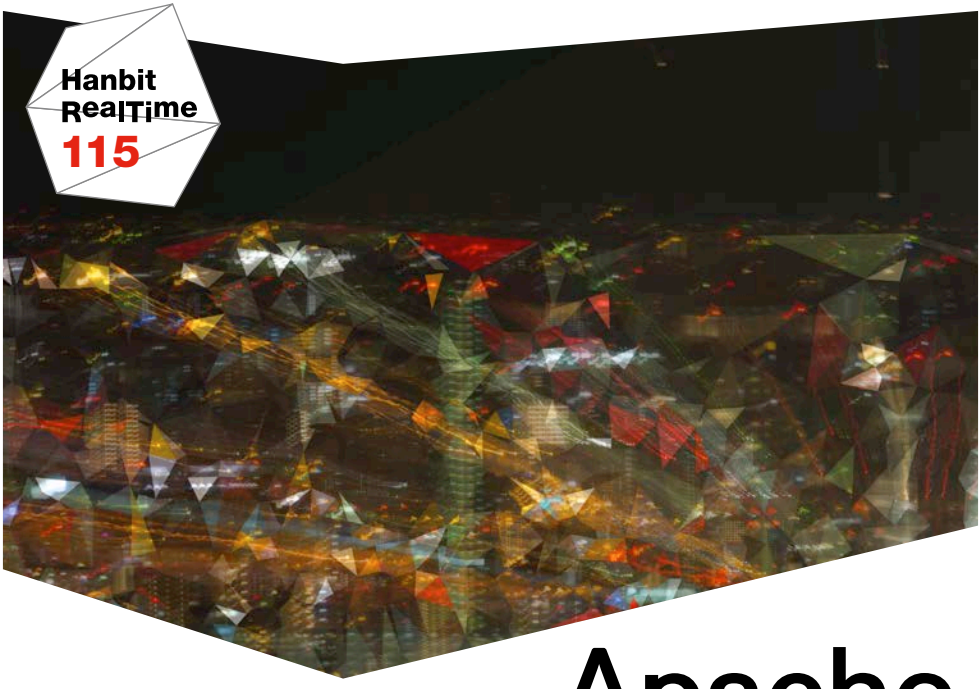
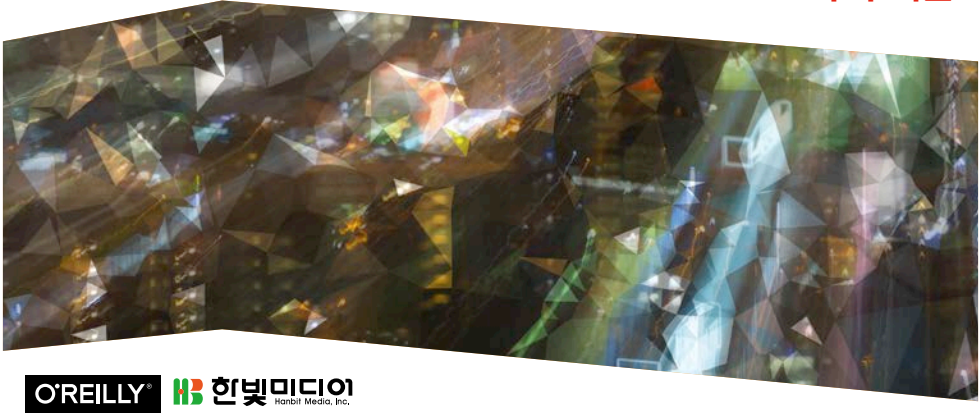
Hanbit RealTime 115



Apache CloudStack

세바스찬 고아스구엔 지음 /
최민석 옮김

클라우드 컴퓨팅을 관통하는 클라우드스택의 60가지 제안



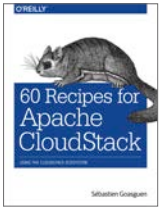
O'REILLY® 한빛미디어 Hanbit Media, Inc.

Hanbit RealTime 115

Apache CloudStack

세바스찬 고아스구엔 지음 /
최민석 옮김

클라우드 컴퓨팅을 관통하는 클라우드스택의 60가지 제안



이 도서는
Developing a React Edge(BLEEDING EDGE PRESS)의
번역서입니다

O'REILLY 한빛미디어 Hanbit Media, Inc.

표지 사진 **성진우**

이 책의 표지는 성진우님이 보내 주신 풍경사진을 담았습니다.
리얼타임은 독자의 시선을 담은 풍경사진을 책 표지로 보여주고자 합니다.

사진 보내기 **ebookwriter@hanbit.co.kr**

Apache CloudStack 클라우드 컴퓨팅을 관통하는 클라우드스택의 60가지 제안

초판발행 2015년 11월 30일

지은이 세바스찬 고아스구엔 / **옮긴이** 최민석 / **펴낸이** 김태헌
펴낸곳 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부
전화 02-325-5544 / **팩스** 02-336-7124
등록 1999년 6월 24일 제10-1779호
ISBN 978-89-6848-776-7 15000 / **정가** 13,000원

총괄 배용석 / **책임편집** 김창수 / **기획·편집** 김상민
디자인 표지/내지 여동일, 조판 김경수
마케팅 박상용, 송경석 / **영업** 김형진, 김진불, 조유미

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오.
한빛미디어 홈페이지 www.hanbit.co.kr / **이메일** ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Authorized Korean translation of the English edition of *60 Recipes for Apache CloudStack, ISBN 9781491911396*

이 책은 클라우드스택 일반 사용자와 클라우드스택 개발자, 그리고 데브옵스DevOps 지향 개발자와 일반 애플리케이션 개발자를 모두 염두에 두고 구성됐다. 또한 이 책은 아파치 클라우드스택의 생태계를 설명하고 다양한 설정에 사용되는 도구를 함께 소개한다. 예를 들어 셰프Chef, 앤서블Ansible, 베이그런트Vagrant와 같은 도구는 물론, 하둡Hadoop과 같은 애플리케이션과 RiakCS와 같은 저장소 솔루션도 소개한다. 즉, 클라우드스택에 대해서만 다루는 책은 아니다.

또한 이 책은 단일 주제에 대한 레시피만 다루는 표준 쿡북이 아니며, 이보다는 다양한 도구를 소개하고 각각의 도구를 간단하게 설명하는 책이다. 즉, 특정한 도구나 애플리케이션에 대한 간단한 자습서가 필요할 때마다 찾아볼 수 있는 자습서로 활용하도록 구성됐다. 이러한 도구는 개발자와 시스템 관리자, 그리고 설계자가 클라우드스택과 함께 사용하는 필수 기술로 자리 잡고 있으며, IT 전문가가 안정된 클라우드를 기반으로 더 많은 일을 원활하게 할 수 있게 도와준다.

조금 이상하게 생각될 수 있지만 필자는 IT 환경의 혁명이라고 할 만큼 놀라운 클라우드 서비스를 제공해준 아마존 웹 서비스 팀에게 감사 인사를 전하고 싶다. 아마존은 컴퓨팅을 공공 서비스로 제공하는 비전을 가장 먼저 제시한 기업이었으며 컴퓨트 리소스와 상호 작용하는 방법을 혁신하는 데 주도적 역할을 해왔다. 또한 아파치 소프트웨어 재단 클라우드스택 커뮤니티 전체, 특히 클라우드스택을 개발하고 출시하는 데 최선의 노력을 다한 분들에게 감사 인사를 전하고 싶다. 건강한 커뮤니티 없이는 생태계가 존재할 수 없다. 그리고 각자의 귀중한 시간을 내서 이 책을 검토하고 피드백을 제시해 준 마이크 투코스키, 제프 무디, 피에르-이브 리차드에게도 감사 인사를 전한다. 마지막으로 이 책을 집필할 시간을 허락해준 마크 힝클, 그리고 필자의 전화를 받아주고 필자와 함께 이 책에 가장 적합한 형식을 결정하는 브레인스토밍을 해준 브라이언 앤더슨에게도 감사 인사를 전한다.

지은이_ **세바스찬 고아스구엔**

세바스찬 고아스구엔Sebastien Goasguen은 1990년대 후반 박사 과정을 진행하면서 처음 컴퓨터 클러스터(당시에는 베오울프 클러스터라고 불림)를 구축했으며 이후로도 계속 컴퓨팅 유틸리티를 만드는 일을 하고 있다. 그리드 컴퓨팅과 고성능 컴퓨팅과 관련된 연구를 했으며, 가상화가 등장한 이후 2000년 중반부터는 클라우드 컴퓨팅을 집중적으로 연구하고 있다. 현재는 Citrix에서 선임 오픈소스 솔루션 설계자로 일하고 있으며, 주로 아파치 클라우드스택 프로젝트에 참여하면서 클라우드스택의 생태계를 지원하고 있다.

세바스찬은 클라우드스택과 아파치 리브클라우드의 PMC(프로젝트 관리 위원회) 회원이며 아파치 소프트웨어 재단에서도 회원으로 활동하고 있다. 현재는 클라우드 생태계에 집중하고 있으며 다양한 오픈소스 프로젝트에 기여했지만 그중에서도 나이프-클라우드스택, Eutester, 앤서블 등의 프로젝트에 기여했다. 그는 또한 Transifex와 ReadTheDocs를 사용해 클라우드스택 설명서를 현지화하는 작업도 주도하고 있다.

역자 소개

옮긴이_ **최민석**

2005년부터 번역회사에서 언어전문가로 일하다 독립한 후 현재는 IT 전문번역가로 일하고 있다. 10여년 동안 수백 건의 소프트웨어 현지화와 개발자 웹 사이트 한글화 프로젝트를 진행했으며 최근에는 이전부터 주의 깊게 지켜보던 IT 전문 서적을 번역하는 일에 집중하고 있다. 번역한 책으로는 『엔터프라이즈 애플리케이션 아키텍처 패턴』, 『유니티와 C#으로 배우는 게임 개발 교과서』, 『자바 웹 개발 완벽 가이드』, 『시작하세요! 스프링 4 프로그래밍』, 『게임 디자인 워크숍』 등이 있다.

이 책을 집필한 이유

필자는 2002년 즈음부터 가상화virtualization와 지금은 클라우드라고 불리는 기술과 관련된 일을 해왔다. 이제 클라우드를 구축할 때는 유칼립투스의 CEO 마틴 믹코스Marten Mickos가 네 자매라고 부르는 클라우드스택, 유칼립투스(https://www.eucalyptus.com/), 오픈네뷸라(http://opennebula.org/), 그리고 오픈스택(http://www.openstack.org/)의 네 가지 오픈소스 솔루션 중에서 원하는 솔루션을 선택할 수 있다. 이제 전 세계 모든 곳에서 수많은 사설 및 공용 클라우드가 성공적으로 운영되고 있다. 다시 말해 클라우드를 구축하는 일은 해결 방법이 마련된 문제로 받아들여지고 있다. 이러한 각 클라우드의 기능과 확장성, 그리고 세부적인 네트워킹이나 저장소 기능은 분명히 차이가 있지만 이들 솔루션은 모두 실무 현장에서 아주 잘 작동하고 있다. 필자가 클라우드 솔루션의 설치 방법을 설명하는 것보다는 이러한 클라우드 솔루션과 관련된 소프트웨어 생태계를 살펴보고, 클라우드의 사용법과 이를 개발과 운영 프로세스로 통합하여 클라우드를 바탕으로 보다 수준 높은 서비스를 제공하고 투자 효율을 높이는 방법을 알아보는 것이 중요하다고 생각하는 것도 바로 이런 이유에서다.

필자는 2012년부터 아파치 클라우드스택 커뮤니티에 참여하기 시작한 이후 지금까지 IT 개발자와 시스템 관리자가 유용하게 활용할 수 있는 다양한 도구를 테스트하고 이러한 도구의 클라우드스택 드라이버를 개발하는 일을 해왔다. 그러면서 일반 사용자도 이러한 도구를 직접 활용할 수 있다는 생각을 하게 됐다. 그리고 더 나아가 이러한 도구를 테스트하면서 축적한 경험을 공유하고, 구축한 클라우드를 활용하는 방법에 대한 질문을 답하며, 무엇보다 안정적인 애플리케이션 호스팅, 분산 애플리케이션 배포, 그리고 데이터 처리라는 핵심적인 문제를 집중적으로 알아보는 책을 써야겠다는 생각을 했다.

이 책은 이제 성숙 단계에 이른 클라우드 기술을 최대한 활용하여 구현 세부 사항에 대해 고민할 필요 없이 애플리케이션을 활용하는 데만 집중하도록 도와줄 것이다.

클라우드 컴퓨팅에서 클라우드스택의 현재 상황

클라우드 컴퓨팅은 상당히 모호한 용어일 수 있다. 어떤 사람은 온라인 애플리케이션을 떠올리지만, 가상화 시스템을 떠올리는 사람도 있다. 클라우드의 개념을 정확하게 이해하는 데는 미국 국립표준기술연구소(NIST)에서 제시한 정의가 도움이 된다. NIST는 2011년 보고서에서 클라우드 컴퓨팅의 의미를 다음과 같이 정의했다.

> 클라우드 컴퓨팅이란 최소한의 관리 노력으로 신속하게 공급하고 회수할 수 있는 구성 가능한 컴퓨팅 리소스(예: 네트워크, 서버, 저장소, 애플리케이션, 서비스)의 공유된 풀을 네트워크를 통해 편리하게 접근하도록 지원하는 주문식 유비쿼터스 모델이다…

NIST는 이어 클라우드의 핵심 특성(즉, 주문식, 네트워크 접근, 다중 테넌시, 탄력성, 계량)을 정의한 후, 클라우드의 세 가지 서비스 모델인 SaaS[software as a service], PaaS[platform as a service], 그리고 IaaS[infrastructure as a service]를 정의한다. 그다음에는 네 가지 배포 모델인 사설 클라우드, 공용 클라우드, 하이브리드 클라우드, 그리고 커뮤니티 클라우드를 정의한다(그중 커뮤니티 클라우드는 많이 알려지지 않았으며 현재는 거의 사용되지 않는다).

SaaS-IaaS 모델은 과거의 ISO 모델과 대응된다. SaaS는 애플리케이션 전달을 처리하고, IaaS는 인프라 관리를 처리하며, PaaS는 그 중간의 모든 사항을 처리한다. 아주 간소화된 관점이지만 핵심을 정확하게 이해할 수 있다. SaaS는 온라인 애플리케이션 호스팅을 의미한다. 사용자는 인터넷을 통해 애플리케이션 인터페이스에 접근하며 애플리케이션을 제공하고 확장하기 위해 백그라운드로 수행되는 모든 세부 작업은 사용자로부터 완전히 숨겨진다. 지메일(그리고 카렌다와 독스를 포함한 거의 모든 구글 서비스)은 전형적인 SaaS의 예다. PaaS는 그동안 흔히 미들웨어[middleware]라고 하

던 기술을 의미하며 최종 사용자 애플리케이션과 이 애플리케이션이 실행되는 기본 인프라 사이를 연결하는 역할을 한다. PaaS 솔루션은 인프라에 대한 걱정을 덜고자 하는 개발자를 위한 것이다. PaaS는 현재 아주 빠르게 발전하고 있는 영역으로서 그 중에서도 Openshift(http://www.openshift.com/), CloudFoundry(http://cloudfoundry.org/), Cloudify(http://cloudifysource.org/)가 많은 관심을 모으고 있으며, 제품의 개발 속도도 아주 빠르다. IaaS은 서버 공급, 네트워크 관리, 저장소 할당을 포함해 시스템 관리자가 애플리케이션을 호스팅하기 위해 수행하는 작업을 조율하는 인프라 계층이다.

아파치 클라우드스택은 IaaS 소프트웨어 솔루션으로서, 대규모의 가상 머신 인스턴스를 가용성과 확장성이 아주 우수한 방법으로 관리할 수 있도록 설계됐다. 아파치 클라우드스택은 탄력적인 주문식 다중 테넌트 컴퓨트, 저장소 및 네트워크 서비스를 제공하는 공용 또는 사설 클라우드를 구축하는 데 적합하다. 그리고 앞서 언급한 것처럼, 아마존 EC2 클론을 구축하는 데 사용되는 오픈 소스 클라우드 컴퓨팅의 네 자매 중 하나로 알려져 있다.

클라우드스택은 2008년 VMOps라는 회사의 프로젝트 중 하나로 시작됐다. 이 회사는 2010년 Cloud.com으로 사명을 바꾸었고 이후 Citrix Systems에 인수됐다. 그리고 2012년 Citrix Systems는 클라우드스택을 ASF(아파치 소프트웨어 재단)에 기부했다. 클라우드스택은 이후 아파치 인큐베이터 프로젝트로 선정되고 ASF의 트레이드마크로 자리잡았으며, 2013년에는 HTTPD나 하둡과 같은 다른 오픈소스 프로젝트와 함께 최상위 ASF 프로젝트 중 하나가 됐다.

이 책의 구성 방식

이 책은 아파치 클라우드스택의 생태계에 쉽고 빠르게 접근할 수 있도록, 각각 두 개의 장을 포함하는 세 개의 부로 구성됐다.

1부에서는 소스와 바이너리를 사용한 설치 단계를 설명한다.

- **1장 소스를 통한 설치** – 개발자에게 유용한 기본 단계를 설명한다. 전체 설치 단계에 대한 자세한 내용은 클라우드스택 설명서(http://docs.cloudstack.apache.org/)에서 볼 수 있으므로 여기에서는 이러한 세부 사항을 반복하기보다 클라우드스택을 소개하고 생태계 개발에 도움이 되는 몇 가지 기능(예: 시뮬레이터와 클라우드스택 샌드박스인 데브클라우드)을 집중적으로 설명한다.
- **2장 패키지를 통한 설치** – 우분투 14.04에서 KVM을 사용한 단계별 패키지 설치 가이드를 제공한다. 이 가이드는 VMware 퓨전을 사용해 로컬 시스템(KVM으로 중첩 가상화를 활용)에서 따라하거나 실제 하드웨어에서 따라할 수 있다. 2장은 소스를 컴파일할 필요가 없는 일반 사용자를 위한 내용이다.

2부에서는 API 클라이언트와 래퍼를 다룬다.

- **3장 API 클라이언트** – API 요청을 서명하는 방법을 설명하고 클라우드몽키(공식 클라우드스택 명령줄 인터페이스), 아파치 리브클라우드(클라우드 공급자 API 간의 차이를 추상화하는 파이썬 모듈), 아파치 제이클라우즈(리브클라우드와 비슷한 목적을 가진 자바 라이브러리), 클로스택(클라우드스택용 클로저 기반 클라이언트)을 포함한 몇 가지 클라이언트를 소개한다. 3장에서 각자 선호하는 언어에 맞는 클라이언트를 찾을 수 있을 것이다. 3장은 클라우드스택 API에 기반을 둔 애플리케이션을 개발하려는 개발자에게 유용하다.
- **4장 API 인터페이스** – 클라우드스택 API 전면에서 다른 API를 제공하는 세 가지 애플리케이션을 소개한다. 이러한 애플리케이션은 API 브리지API bridge 또는 래퍼wrapper라고도 하며 사용자의 시스템에서 또는 클라우드 공급자의 인프라에서 서버로서 실행되어 다른 API

를 제공하는 역할을 한다. 예를 들어 EC2Stack은 EC2 호환 인터페이스를 제공하고, gstack은 GCE 호환 인터페이스를 제공하며, rOCCI는 표준화된 인터페이스를 제공한다. 또한 4장에서는 유칼립투스 팀이 개발한 두 가지 파이썬 모듈인 보토Boto와 Eutester를 소개한다. 보토는 AWS (아마존 웹 서비스) 클라이언트이며, Eutester는 테스트 프레임워크다. 클라우드스택 사용자는 이러한 모듈을 EC2Stack과 함께 사용할 수 있다.

3부에서는 구성 관리와 몇 가지 고급 레시피를 소개한다.

- **5장 구성 관리 –** 5장에서는 먼저 비위Veewee와 팩커Packer를 소개한다. 그다음에는 로컬에서 구성을 테스트하고 이를 클라우드에 반복 가능한 방법으로 배포하도록 도와주는 소프트웨어 개발 도구인 베이그런트Vagrant 관련 레시피 몇 가지를 설명한다. 베이그런트에 대한 내용을 제외한 5장의 나머지 부분은 구성 관리 솔루션인 앤서블과 셰프를 소개하는 데 할애했다. 이들 솔루션은 클라우드에 애플리케이션을 배포하는 데 도움이 되는 클라우드스택 플러그인을 지원한다. 5장은 데브옵스 커뮤니티에서 관심을 많이 가질 것이다.
- **6장 고급 레시피 –** 6장에서는 몇 가지 수준 높은 주제를 다룬다. 클라우드 인프라의 두 가지 중요한 측면인 모니터링과 저장소에 대해 알아보고, RiakCS를 소개한 다음 이를 이미지 카탈로그로 사용하는 방법을 설명한다. 또한 플루언트Fluent를 엘라스틱서치Elasticsearch 및 몽고DBMongoDB와 함께 사용해 로그를 집계하는 방법도 알아본다. 마지막으로 제이클라우즈에 기반을 둔 애플리케이션 조정자이며 하둡과 같은 분산 시스템을 배포 및 실행하는 데 사용할 수 있는 아파치 웜Apache Whirr을 소개한다.

마지막으로 4부에서는 책의 내용을 요약하고 다른 서적이나 자료에 대한 몇 가지 정보를 제공한다.

이해가 필요한 기술

이 책은 중간 수준의 사용자를 위한 책으로서, 제대로 활용하려면 개발과 시스템 관리 개념에 대한 최소한의 이해가 필요하다. 본격적인 내용으로 들어가기 전에 다음 사항을 먼저 확인하는 것이 좋다.

배시(bash, 유닉스 셸)

배시bash는 리눅스와 OS X의 기본 유닉스 셸이다. 유닉스 셸을 사용한 파일 편집, 파일 사용 권한 설정, 파일 시스템 안에서 파일 이동, 사용자 권한 및 기본 셸 프로그래밍과 같은 기본적인 작업 방법을 알고 있으면 아주 유용하다. 리눅스 셸에 대한 전반적인 내용을 배우려면 Cameron Newham의 Learning the Bash Shell이나 Carl Albing, JP Vossen, Cameron Newham의 bash Cookbook과 같은 책을 추천한다.

패키지 관리

이 책에서 소개하는 도구는 다른 패키지를 설치해 여러 의존성을 충족해야 제대로 작동하는 경우가 많다. 따라서 자신의 시스템에서 패키지를 관리할 수 있는 기본 지식이 필요하다. 패키지 관리를 위한 도구로는 우분투/데비안 시스템에서는 앱트apt, CentOS/RHEL 시스템에서는 윰yum, OS X에서는 포트port나 브루brew를 사용할 수 있다. 사용하는 도구에 관계없이 패키지를 설치, 업그레이드, 제거하는 방법을 숙지해야 한다.

깃(Git)

깃Git은 분산 버전 제어를 위한 표준으로 자리 잡았다. CVS과 SVN에 익숙하면서도 아직 깃 사용법을 모른다면 꼭 사용해보기를 권한다. 초보자에게는 Jon Loeliger와

Matthew McCullough의 Version Control with Git (O'Reilly)을 추천한다. 깃과 함께 깃허브 웹 사이트는 각자 보유한 리포지토리를 호스팅할 수 있는 훌륭한 자원이다. 깃허브 사용법을 배우려는 독자에게는 http://training.github.com과 관련 대화식 자습서를 추천한다.

파이썬

C/C++이나 자바를 사용한 프로그래밍도 중요하지만, 필자는 항상 학생들에게 스크립팅 언어 하나를 선택해서 배우도록 추천한다. 스크립트 언어로는 과거에는 펄이 많이 사용됐지만, 현재는 루비나 고Go가 더 많이 사용된다. 필자는 개인적으로 파이썬을 사용한다. 이 책의 예제는 대부분 파이썬으로 작성됐고, 일부 예제는 루비와 클로저로 작성됐다. Bill Lubanovic의 Introducing Python, Mark Lutz의 Programming Python, 그리고 David Beazley와 Brian K. Jones의 Python Cookbook을 비롯해 파이썬에 대한 좋은 책이 많이 있다.

이와 같이 셸, 패키지 관리자, 깃허브 계정, 그리고 파이썬이 바로 앞으로 여러분이 사용할 주요 도구다. 아직 잘 모르더라도(특히 파이썬) 걱정할 필요는 없다. 루비와 클로저 프로그래머를 위한 레시피도 있으며, 진행하는 동안 필요한 내용을 배워도 된다.

온라인 자료

이 책에서 소개하는 여러 도구에 대한 자체 관리 학습을 원한다면 온라인 자습서(http://codac.co)를 확인해보자. 이 자습서에서는 클라우드스택 기반 공용 클라우드인 exoscale을 활용한다. exoscale에는 무료로 등록할 수 있으며 기본 제공되는 무료 크레딧으로 자습서를 완료할 수 있다.

이 책의 내용은 거의 대부분을 exoscale에서 테스트했지만, 이 밖에도 이러한 도구를 테스트하고 실무에 활용할 수 있는 다른 공용 클라우드스택 클라우드가 많이 있다. 이러한 클라우드 중 일부 또는 전부에서 계정을 등록하는 것을 고려해볼 수 있다.

- **exoscale** – http://exoscale.ch/
- **iKoula** – http://express.ikoula.com/cloud-public
- **British Telecom Cloud** – https://www.btcloud.bt.com/
- **GreenQloud** – https://www.greenqloud.com/
- **Leaseweb** – http://www.leaseweb.com/en/cloud
- **Cloud-n VERIO** – http://www.verio.com/cloud-computing/
- **PCExtreme** – https://www.pcextreme.nl/en/aurora/
- **Kuomo** – http://www.kumo.com.co/
- **Interoute Virtual Data Center (VDC)** – http://bit.ly/Interoute_VDC

이 책의 표기법

이 아이콘은 일반적인 주석을 의미한다.

이 아이콘은 팁이나 제안을 의미한다.

이 아이콘은 경고나 주의를 의미한다.

한빛 리얼타임

한빛 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1 eBook First – 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 보다 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2 무료로 업데이트되는 전자책 전용 서비스입니다

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3 독자의 편의를 위해 DRM-Free로 제공합니다

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

Part 2 클라이언트와 API 래퍼 67

chapter 3 API 클라이언트 71

chapter 4 API 인터페이스 115

Part 1

설치

이 책은 클라우드스택을 설치하고 구성하는 방법을 설명하는 책은 아니지만, 설치 프로세스에 대한 기본 사항을 어느 정도 다루는 것도 필요하다고 생각한다. 클라우드스택 설치와 구성 방법은 공식 클라우드스택 설명서(http://docs.cloudstack.apache.org/)에 아주 자세하게 나와 있다. 물론 클라우드스택 기반 공용 클라우드에서 클라이언트와 도구를 테스트하는 데만 이 책을 활용한다면 설치에 대해서는 신경 쓰지 않아도 된다. 그러나 원하는 배포 방식을 사용해 로컬에서 클라이언트와 도구를 테스트하고 싶을 수도 있다. 1부에서는 우분투 14.04와 CentOS 6.5에서 소스를 통한 기본 설치 과정을 안내하는 몇 가지 레시피를 소개하고, 클라우드스택 시뮬레이터와 데브클라우드(클라우드스택 샌드박스)를 활용해 개발자에게 유용한 환경을 만드는 방법을 설명한다. 클라우드스택을 컴파일하는 과정은 아주 쉽게 따라할 수 있으며, 두어 시간 정도면 작동하는 개발 환경을 구축할 수 있다. 또한 KVM 하이퍼바이저를 사용해 패키지를 통한 클라우드스택 기본 영역 설치 방법도 설명한다. 이 설치 방법은 소스 릴리스를 사용하고 싶지 않은 처음 사용자에게 적합하다.

chapter 1

소스를 통한 설치

우선 코드를 빌드할 필요가 있는 클라우드스택 개발자를 대상으로 하는 레시피를 소개한다. 소개하는 레시피의 설명은 우분투 14.04와 CentOS 6.5 시스템에 적용되며 아파치 클라우드스택 4.4 버전대의 제품으로 테스트했다. 다른 운영체제나 클라우드스택의 이전 또는 이후 버전을 사용하는 경우에는 환경에 맞게 조정이 필요할 수 있다. 이 레시피에서는 다음과 같은 내용을 다룬다.

- 필수 구성요소 설치
- 소스 컴파일 및 설치
- 클라우드스택 시뮬레이터 사용
- 클라우드스택 샌드박스: 데브클라우드를 사용한 테스트
- 자신의 패키지 생성

이 책을 집필하는 동안에는 4.4 릴리즈가 아직 발표되지 않은 상태였다. 따라서 소스 테스트는 릴리스 브랜치 4.4에서 수행했고 4.3.0 패키지를 사용했다.

1.1 우분투 14.04에서 필수 구성요소 설치

문제

클라우드스택도 다른 소프트웨어와 마찬가지로 여러 의존성이 필요하다. 이 레시

피에서는 아파치 클라우드스택을 개발하는 데 필요한 의존성을 설치하는 방법을 설명한다. 여기에서 소개하는 패키지 설치 과정으로 우분투 14.04에서 클라우드스택을 빌드할 수 있는 시스템을 설정할 수 있다.

해결책

먼저 시스템을 업데이트한다. 이 책에서는 리눅스를 주로 사용하므로 가장 적합한 OpenJDK를 설치한다. 그다음 MySQL을 설치하고, 클라우드스택 소스 코드를 복제하기 위한 깃을 설치한다. 마지막으로 나중에 클라우드스택을 빌드하는 데 사용할 메이븐을 설치한다.

설명

다음과 같은 패키지 설치 과정으로 우분투/데비안 기반 시스템에서 필수 구성요소를 설치할 수 있다(우분투 14.04 시스템에서 테스트).

```
# apt-get -y update
# apt-get -y install openjdk-7-jdk
# apt-get -y install mysql-server
# apt-get install git
# apt-get install maven
```

MySQL도 실행 중이어야 하며 다음과 같이 상태를 확인할 수 있다.

```
# service mysql status
```

메이븐 3.0.5도 설치돼 있어야 하며 mvn --version으로 버전을 확인할 수 있다.

레시피 1.4에서 시뮬레이터에 대해 알아볼 때 설명하겠지만, 파이썬도 약간 사용한다. 따라서 파이썬 패키지 관리 도구를 설치한다.

```
# apt-get install python-pip
```

1.2 CentOS 6.5에서 필수 구성요소 설치

문제

클라우드스택도 다른 소프트웨어와 마찬가지로 여러 의존성이 필요하다. 이 레시피에서는 아파치 클라우드스택 개발에 필요한 의존성을 설치하는 방법을 설명한다. CentOS 6.5에서 클라우드스택을 빌드할 수 있는 시스템을 설정하려면 다음에 나오는 패키지 설치 과정을 따라하면 된다.

해결책

먼저 시스템을 업데이트한다. 이 책에서는 리눅스를 주로 사용하므로 가장 적합한 OpenJDK를 설치한다. 아직 시스템에 설치하지 않은 경우 MySQL을 설치한다. 클라우드스택 소스 코드를 복제하기 위한 깃을 설치한다. 마지막으로 genisoimage 패키지를 통해 mkisofs를 설치한다.

설명

다음과 같은 패키지 설치 과정으로 CentOS/RHEL 기반 시스템에서 필수 구성요소를 대부분 설치할 수 있다(CentOS 6.5 시스템에서 테스트).

```
# yum -y update
# yum -y install java-1.7.0-openjdk
# yum -y install java-1.7.0-openjdk-devel
# yum -y install mysql-server
# yum -y install git
# yum -y install genisoimage
```

MySQL은 중지해야 한다. service mysqld status로 상태를 확인하고 service mysqld start로 시작할 수 있다.

다음은 클라우드스택을 빌드하는 데 필요한 메이븐을 설치해야 한다. 이 과정은 우분투 14.04보다 약간 복잡하다. 먼저 메이븐 웹 사이트(http://maven.

apache.org/download.cgi)에서 3.0.5 릴리스를 받는다.

```
# wget http://mirror.cc.columbia.edu/pub/software/apache/maven/maven-3/ \
3.0.5/binaries/apache-maven-3.0.5-bin.tar.gz
# tar xzf apache-maven-3.0.5-bin.tar.gz -C /usr/local
# cd /usr/local
# ln -s apache-maven-3.0.5 maven
```

다음과 같은 내용을 포함하는 /etc/profile.d/maven.sh 파일을 작성하면 시스템 수준에서 메이븐을 설정할 수 있다.

```
# export M2_HOME=/usr/local/maven
# export PATH=${M2_HOME}/bin:${PATH}
```

다음 명령을 실행하면 경로에 메이븐이 추가된다(앞서 수행한 단계로 메이븐 3.0을 설치할 수 있다. 메이븐의 버전은 mvn --version으로 확인할 수 있다).

```
# source /etc/profile.d/maven.sh
# mvn —version
```

파이썬을 약간 사용하게 될 수 있으므로(레시피 1.4 참고) 파이썬 패키지 인덱스 유틸리티(pip)를 설치한다.

```
# yum -y install python-pip
```

CentOS 6.5에서는 파이썬 2.6.6을 사용하는데, 시뮬레이터와 클라우드스택 마빈(CloudStack Marvin) 패키지를 사용하려면 파이썬 2.7 이상이 필요하다. 파이썬 2.7을 설정하는 과정은 이 레시피에서 다루지 않는다.

1.3 소스를 통한 설치

문제

필수 구성요소를 설치(레시피 1.1 또는 레시피 1.2)한 다음 소스를 사용해 클라우드스택을 빌드하고 로컬에서 관리 서버를 실행할 수 있다.

해결책

클라우드스택의 깃 리포지토리를 복제하고, 여러 메이븐 프로파일을 사용해 소스를 빌드한 후, 데이터베이스를 설정하고, 제티Jetty로 관리 서버를 실행하면 된다.

설명

우분투나 CentOS에서 필수 구성요소를 설치하는 단계를 완료한 후 빌드 단계는 두 시스템에서 동일하다. 클라우드스택은 소스 버전 제어를 위해 깃을 사용한다. 아직 깃에 익숙하지 않다면 깃허브 자습서(http://try.github.io/)를 추천한다.

이 절에서 설명하는 단계는 아파치 클라우드스택을 빌드하는 데 필요한 최소 단계다. 하이퍼바이저를 설정하고, 저장소를 설정하며, 비-디버그 및 프로덕션 모드로 클라우드스택 관리 서버를 실행하려면 추가 패키지가 필요하다.

시스템에서 깃을 설정한 후 다음 명령으로 아파치 리포지토리에서 소스를 가져온다.

```
git clone -b 4.4 https://git-wip-us.apache.org/repos/asf/cloudstack.git
```

깃허브에서도 아파치 클라우드스택 리포지토리를 미러링하고 있다. 여기에서 복제하는 것도 가능하다.

```
git clone -b 4.4 https://github.com/apache/cloudstack.git
```

필자는 아파치를 응원하는 입장이지만, 아파치 깃 리포지토리는 다소 느린 것이 사실이다. 깃허브 미러를 사용하는 편이 더 빠르다.

필수 구성요소를 설치하고 소스 코드를 복제하면 4.4 릴리스를 빌드할 준비가 된다. 이 책이 출간되는 즈음에는 안정된 최신 릴리스가 나왔을 것이다. 클라우드스택을 컴파일하려면 클라우드스택 소스 폴더에서 다음 메이븐 명령을 실행한다(단위 테스트를 생략하려면 이 명령에 -DskipTests를 추가하면 된다).

```
cd cloudstack
mvn -Pdeveloper,systemvm clean install
```

그리고 다음 명령으로 필요한 MySQL 테이블을 생성한다.

```
mvn -Pdeveloper -pl developer -Ddeploydb
```

어떤 테이블이 생성되었는지 확인하려면 대화식 MySQL 셸을 열고 cloud 데이터베이스를 살펴보면 된다. 아파치 클라우드스택 관리 서버는 명령 하나로 실행할 수 있다. 테스트에는 제티를 사용한다. 시스템에 톰캣을 설치한 경우 포트 8080에서 실행 중일 것이므로 제티를 사용하기 전에 톰캣을 중지해야 한다.

```
mvn -pl :cloud-client-ui jetty:run
```

제티는 포그라운드로 실행되며 해당하는 로그는 stdout에서 볼 수 있다. 대시보드를 사용하려면 웹 브라우저를 열고 http://localhost:8080/client로 이동한다(localhost 부분은 필요에 따라 관리 서버의 IP로 바꿈).

iptables를 활성화한 경우 클라우드스택에서 사용하는 포트(즉, 8080, 8250, 9090)를 열어야 할 수 있다. 또한 경우에 따라 테스트를 위해 방화벽을 비활성화해야 할 수 있지만(우분투에서 ufw disable 사용, CentOS에서 service iptables stop 사용), 해당 시스템을 공용 인터넷에 연결하기 전에 방화벽을 다시 활성화하는 것을 잊지 않아야 한다.

이제 대시보드를 조작하고 API를 사용해볼 수 있다. 물론 아직은 인프라를 설정

하지 않았고 저장소나 하이퍼바이저, 네트워크도 정의하지 않았기 때문에 아직은 인스턴스를 시작할 수 없지만, 조금 뒤에 시뮬레이터를 사용하는 방법을 알아본다. 다음 레시피에서는 실제 인프라를 설정할 필요 없이 로컬 시스템에서 시뮬레이터를 사용한 가상 데이터 센터로 개발과 테스트를 하는 방법을 보여준다.

1.4 클라우드스택 시뮬레이터 사용

문제

클라우드스택 관리 서버의 몇 가지 기능을 테스트를 해야 하지만, 완전한 클라우드스택 클라우드를 실제 인프라에 배포하기가 어려운 상황이다.

해결책

클라우드스택 시뮬레이터를 사용해 가상 데이터 센터를 설정한다. simulator 프로파일로 클라우드스택을 구성하고 다른 메이븐 프로파일로 몇 가지 특정 데이터베이스 테이블을 설정한다. 그다음은 클라우드스택용 파이썬 기반 테스트 프레임워크인 마빈(Marvin)을 설치한다. 관리 서버를 다시 시작하면 마빈 스크립트(deployDataCenter.py)를 사용해 시뮬레이트된 인프라를 구성할 수 있다.

설명

클라우드스택에는 마빈이라는 파이썬 바인딩 기반의 시뮬레이터가 포함돼 있다. 마빈은 클라우드스택 소스 코드 또는 PyPI에서 얻을 수 있다. 마빈을 사용하면 영역/포드/클러스터/호스트의 수, 저장소 유형 등을 정의하는 구성 파일을 클라우드스택에 제공하고 데이터 센터 인프라를 시뮬레이션할 수 있다. 그러면 시뮬레이터에서 프로덕션 인프라를 관리하는 것처럼 클라우드스택 관리 서버를 개발 및 테스트할 수 있다. 시뮬레이터를 사용하려면 빌드 단계 중 일부를 변경하고 마빈을 사용해 시뮬레이트되는 데이터 센터를 구성해야 한다.

먼저 클린 빌드를 하고 simulator 프로필을 추가한다.

```
mvn -Pdeveloper -Dsimulator -DskipTests clean install
```

그런 다음 데이터베이스를 배포하고 다음과 같이 시뮬레이터 관련 테이블과 구성을 설정한다.

```
mvn -Pdeveloper -pl developer -Ddeploydb
mvn -Pdeveloper -pl developer -Ddeploydb-simulator
```

앞서 언급한 것처럼 데이터 센터를 구성하는 데 사용되는 마빈 패키지에는 파이썬 2.7 이상이 필요하다. CentOS 6.5를 사용하는 경우 파이썬을 설치해야 하는데, 이를 위해서는 소스를 통해 파이썬을 빌드해야 하므로 이 단계는 다루지 않는다.

그 다음은 마빈을 설치한다. (앞서 필수 구성요소 단계에서 pip 설치를 완료해야 한다. paramiko 모듈을 설치하려면 python-dev 패키지가 필요하며, --allow-external 플래그를 사용하면 mysql-connector-python 패키지를 설치할 수 있다.)

```
sudo apt-get -y install python-dev
sudo pip install —allow-external mysql-connector-python mysql-connector-
python
sudo pip install tools/marvin/dist/Marvin-0.1.0.tar.gz
```

제티가 실행 중인 경우 중지하고 시뮬레이터 프로파일로 새 관리 서버를 시작한다.

```
mvn -pl :cloud-client-ui jetty:stop
mvn -pl client jetty:run -Dsimulator
```

관리 서버가 포그라운드로 실행 중인 상태에서 별도의 셸을 열고 마빈으로 기본 영역(basic zone) 하나를 설정한다.

```
python ./tools/marvin/marvin/deployDatacenter.py -i setup/dev/basic.cfg
```

이제 http://localhost:8080/client에서 admin/password 자격증명을 사용해 클라우드스택 관리 서버로 로그인한다. 그러면 완전하게 구성된 기본 영역 인프라를 볼 수 있다. 고급 영역을 시뮬레이트하려면 basic.cfg를 advanced.cfg로 바꾸면 된다. 이제 실무 시스템에서 할 수 있는 거의 모든 작업이 가능한 것은 물론 시뮬레이션되는 인스턴스를 시작하고 스냅샷을 만들 수도 있다. 시뮬레이터는 새로운 기능과 API 클라이언트를 테스트하고 통합 테스트를 하는 좋은 방법이다.

1.5 클라우드스택 샌드박스: 데브클라우드 사용

문제

시뮬레이터로는 테스트하거나 시연하기 어려운 기능이 있다. 이러한 경우 중첩 가상화를 활용해 Xen 기반 하이퍼바이저를 로컬에서 실행하고 클라우드스택을 사용해 이 로컬 하이퍼바이저 내에서 가상 머신을 시작하기를 원할 수 있다.

해결책

버추얼박스VirtualBox 이미지인 데브클라우드를 사용해 로컬 시스템에서 Xen 하이퍼바이저를 실행한다. 메이븐을 사용해 데브클라우드 적용 사례에 적합하게 데이터베이스를 설정하고 마빈을 실행해 데브클라우드에 맞게 클라우드스택을 구성한다. 그러면 영역, 포드, 클러스터, 호스트가 각기 하나인 데브클라우드라는 클라우드가 생성된다.

설명

소스를 통한 설치(레시피 1.3)로 관리 서버를 실행할 수 있지만, 하이퍼바이저는

제공되지 않는다. 시뮬레이터를 사용(레시피 1.4)하면 시뮬레이트되는 데이터 센터를 테스트에 이용할 수 있다. 데브클라우드를 사용하면 하이퍼바이저 하나를 사용할 수 있게 되며 실제 시스템과 마찬가지로 이를 관리 서버로 추가할 수 있다.

데브클라우드는 클라우드스택 샌드박스다. 버추얼박스 기반 이미지가 표준 버전이지만, KVM 기반 이미지도 있다. 여기에서는 버추얼박스 이미지에 대한 단계만 살펴본다. KVM 버전에 대한 내용은 클라우드스택 위키(http://bit.ly/DevCloud-KVM)에 잘 정리돼 있다.

데브클라우드를 활용하면 관리 서버를 로컬 시스템에서 실행하고 Xen 하이퍼바이저를 데브클라우드 버추얼박스 이미지에서 실행할 수 있다. 데브클라우드와 로컬 호스트는 버추얼박스에서 사용 가능한 호스트 전용 인터페이스를 통해 연결된다. 데브클라우드에도 공용 인터넷에 접근하기 위한 NAT 인터페이스가 있다.

관리 서버를 데브클라우드 안에서 실행할 수도 있다. 이렇게 하면 소스를 통해 클라우드스택을 컴파일하기 위한 로컬 환경을 구성할 필요가 없게 된다.

데브클라우드 필수 구성요소

시작하기 전에 몇 가지 필수 구성요소를 설치해야 한다.

1. 시스템에 버추얼박스(http://www.virtualbox.org/)를 설치한다.
2. 버추얼박스를 실행하고 환경 설정에서 DHCP 서버를 비활성화한 호스트 전용 인터페이스를 생성한다.
3. 데브클라우드 이미지(http://bit.ly/DevCloud_image)를 다운로드한다.
4. 버추얼박스에서 파일 → 가상 시스템 가져오기를 선택하고 데브클라우드 이미지를 가져온다.
5. 설정으로 들어가서 프로세서 메뉴의 "PAE 사용하기" 옵션을 선택한다.
6. VM이 부팅되면 자격 증명 root/password와 ssh root@192.168.56.10 명령을 사용해 VM에 대해 ssh를 시도해본다.

그 다음에는 관리 서버를 실행하는 시스템에서 영역, 포드, 클러스터, 하이퍼바이저가 각각 하나인 데브클라우드라는 기본 영역을 구성할 수 있다.

데브클라우드를 하이퍼바이저로서 추가

관리 서버를 실행하려면 클린 빌드를 해야 하며, 데이터베이스를 설정할 때 데브클라우드 전용 메이븐 프로파일을 사용해 이 특별한 설정에 맞는 값으로 여러 테이블을 채워야 한다.

```
mvn -Pdeveloper,systemvm clean install
mvn -Pdeveloper -pl developer,tools/devcloud -Ddeploydb
```

다음은 레시피 1.4와 마찬가지로 마빈을 설치한다.

```
pip install tools/marvin/dist/Marvin-0.1.0.tar.gz
```

제티로 관리 서버를 시작한다.

```
mvn -pl client jetty:run
```

클라우드스택 관리 서버가 실행되지만, 아직은 빈 인프라 하나만 있다. 이제 클라우드스택을 구성하고 단일 하이퍼바이저를 포함하는 데이터 센터를 정의해야 한다. 이를 위해 Marvin 디렉터리에 있는 deployDataCenter.py라는 파이썬 스크립트를 사용한다. 이 스크립트는 JSON 파일 하나를 받고 클라우드스택 API를 호출해 해당하는 단일 하이퍼바이저 데이터 센터의 영역, 포드, 클러스터, 호스트, 저장소 구성요소를 생성한다.

```
cd tools/devcloud
python ../marvin/marvin/deployDataCenter.py -i devcloud.cfg
```

내용이 궁금하다면 devcloud.cfg 파일을 열어서 JSON 형식으로 데이터 센터를 정의하는 방법을 확인해볼 수 있다.

이제 http://localhost:8080/client에서 관리 서버로 로그인하고 UI를 사용해 볼 수 있다. 구성이 정상적으로 완료됐다면 대시보드의 Infrastructure 탭에서 정의된 인프라를 볼 수 있다. 영역을 활성화하고 모든 구성요소를 설정한 후에는 Instances 탭에서 인스턴스를 시작할 수 있다. 기본 템플릿은 ttylinux 템플릿이며 여기로 SSH를 실행할 수 있다.

이 메시지에서 관리 서버는 로컬 시스템에서 실행되며 데브클라우드는 하이퍼바이저로만 사용됐다. 관리 서버를 데브클라우드 안에서 실행하는 것도 가능하며 메모리가 충분하다면 데브클라우드를 여러 개 실행할 수도 있다.

데브클라우드는 클라우드스택과 실제 인스턴스를 사용해볼 수 있는 좋은 방법이며, API에 대해 배우고 클라우드스택 네트워킹 설정을 테스트하는 데 적합하다.

이 책을 집필하는 동안에는 데브클라우드를 아직 출시되지 않은 클라우드스택 4.4와 함께 사용하는 데 문제가 있었다. 클라우드스택 4.4가 출시되면 데브클라우드의 문제도 해결될 것이다. GSoC Google Summer of Code 2014 프로젝트에서 개발 중인 솔루션을 대안으로 사용할 수도 있다(레시피 1.6).

1.6 베이그런트 기반 클라우드스택 테스트 배포

문제

데브클라우드 레시피 1.5도 나쁘지는 않지만, 구성 옵션이 다양하고 관리하기 편리한 베이그런트(레시피 5.4)나 셰프(레시피 5.9) 기반 솔루션을 활용해 스택클라우드 개발 환경을 시작하고 싶다.

해결책

베이그런트 기반 프로젝트는 현재 GSoC(구글 섬머 오브 코드) 프로젝트로 개발되고 있다. 이 프로젝트에서는 하이퍼바이저와 NFS, 그리고 MySQL 서버로 작동하는 여러 베이그런트 박스를 이용한다. 클라우드스택 관리 서버는 로컬 시스템에서 실행하며 여기에서 원격 데이터베이스 서버를 연결하게 된다. 데이터 센터를 구성하는 데는 데브클라우드(레시피 1.5)와 마찬가지로 마빈을 사용한다.

설명

이 레시피에서는 여러분이 베이그런트에 대해 어느 정도 알고 있다고 가정한다. 그렇지 않다면 레시피 5.4를 먼저 읽어보자. NFS와 MySQL 서버를 구성하는 작업과 NAT 라우팅을 구성하는 작업은 셰프 레시피를 통해 진행된다. 이 GSoC 프로젝트의 목표는 클라우드스택을 설치하기 위한 완전한 셰프 레시피를 제공하는 것이다.

이 레시피를 사용하려면 시스템에 버추얼박스가 설치돼 있어야 하며, 버추얼박스에서 DHCP 서버가 비활성화된 호스트 전용 인터페이스를 설정해야 한다.

이 프로젝트를 사용하려면 현재 GSoC-2014라고 하는 항목을 복제하고 베이그런트를 사용해 박스를 시작해야 한다. 이 프로젝트에는 깃 하위 모듈이 필요하므로 복제할 때 --recursive 옵션을 사용해 해당 하위 프로젝트에서 코드를 가져와야 한다.

```
git clone --recursive https://github.com/imduffy15/GSoC-2014.git
cd GSoC-2014
cd MySql_NFS_XenServer
vagrant up
```

그러면 CentOS 6.5 기반의 management 머신이 시작되고 셰프(레시피 5.9)가 사용되며, 하이퍼바이저 역할을 할 xenserver 머신도 시작된다. 그다음에는 cloudstack 디렉터리로 돌아가 로컬 시스템에서 클라우드스택 관리 서버를 빌드하고 실행한다. 그리고 마빈을 사용해 데이터 센터를 프로그래밍 방식으로 구성한다(이러한 단계에 대한 설명이 하면 레시피 1.3을 참고한다).

```
wget http://download.cloud.com.s3.amazonaws.com/tools/vhd-util \
     -P scripts/vm/hypervisor/xenserver/
chmod +x scripts/vm/hypervisor/xenserver/vhd-util
mvn -P developer,systemvm clean install -DskipTests=true
mvn -P developer -pl developer,tools/devcloud -Ddeploydb
mvn -pl :cloud-client-ui jetty:run
```

관리 서버가 실행 중일 때 http://localhost:8080/client에서 대시보드에 접근할 수 있다. 다른 셸에서 마빈을 설치하고 devcloud 구성을 배포한다.

```
pip install tools/marvin/dist/Marvin-0.1.0.tar.gz
python tools/marvin/marvin/deployDataCenter.py -i ../devcloud.cfg
```

시스템 VM의 상태를 확인(레시피 2.8)하는 데는 vagrant ssh를 사용하며, 하이퍼바이저에 로그인한 후 실행 중인 VM을 확인하는 데는 xe를 사용한다.

```
cd ../MySql_NFS_XenServer
vagrant ssh xenserver
sudo su
xe vm-list
```

시스템 VM이 정상적으로 가동되면 ttylinux 템플릿이 다운로드 및 설치되며 인스턴스를 시작할 수 있게 된다. 이제 필요한 모든 테스트를 수행하고 원하는 클라우드스택 기능을 개발할 수 있다. 마음껏 사용해보자!

1.7 클라우드스택 바이너리 패키지 빌드

문제

커뮤니티 유지관리 리포지토리에서 바이너리를 받아 사용하면 편리하지만, 소스를 통해 자신이 직접 클라우드스택 패키지를 제작하고 싶을 수 있다.

해결책

소스를 통해 빌드할 때는 필요하지 않았던 몇 가지 패키지를 추가로 설치한다. 우분투에서는 dpkg-buildpackage 명령을 사용하고 CentOS/RHEL 시스템에서는 packaging/centos63/ 디렉터리에 있는 package.sh 스크립트를 사용한다.

설명

클라우드스택 관련 개발에서는 소스를 통한 작업이 필수적이지만, 앞서 언급했듯이 이 작업은 일반 사용자에게는 필요가 없다. 그러나 특정한 용도와 인프라에 맞게 코드를 수정하기를 원할 수 있다. 보안과 관련된 이유나 네트워크 연결 제약 때문에 직접 패키지를 빌드해야 하는 경우도 있다. 이 레시피에서는 패키지를 빌드하는 방법을 간략하게 설명하며, 패키지를 제공할 리포지토리를 만드는 방법은 알고 있다고 가정한다. 자세한 설명서는 웹 사이트(http://bit.ly/build_packages)에서 볼 수 있다.

데비안 패키지를 빌드하려면 소스 컴파일의 필수 구성요소에 포함되지 않는 몇 가지 패키지를 추가로 설치해야 한다.

```
# apt-get -y install python-mysqldb
# apt-get -y install debhelper
# apt-get -y install tomcat6
```

그리고 다음과 같이 패키지를 빌드한다.

```
# dpkg-buildpackage -uc -us
```

클라우드스택 루트 디렉터리에서 한 디렉터리 위로 이동하면 다음 항목을 볼 수 있다.

```
cloudstack_4.4.0_amd64.changes
cloudstack_4.4.0.dsc
cloudstack_4.4.0.tar.gz
cloudstack-agent_4.4.0_all.deb
cloudstack-awsapi_4.4.0_all.deb
cloudstack-cli_4.4.0_all.deb
cloudstack-common_4.4.0_all.deb
cloudstack-docs_4.4.0_all.deb
cloudstack-management_4.4.0_all.deb
cloudstack-usage_4.4.0_all.deb
```

물론 커뮤니티에서도 이러한 패키지를 위한 리포지토리를 제공하고 있으므로, 커뮤니티 리포지토리를 이용하면 직접 패키지를 빌드하고 리포지토리에 넣을 필요가 없다. 커뮤니티 리포지토리를 이용하는 방법은 설치 가이드(http://bit.ly/community_repo)에 나온다.

CentOS/RHEL 시스템에서는 다음 패키지를 추가한다.

```
# yum -y install rpm-build
# yum -y install tomcat6
# yum -y install ws-commons-util
# yum -y install gcc
# yum -y install glibc-devel
# yum -y install MySQL-python
```

그 다음에는 cloudstack 디렉터리에서 ./packaging/centos63/package.sh를 실행해 패키지를 빌드할 수 있다. rpm은 ./dist/rpmbuild/RPMS/

x86_64에 있다. 클라우드스택은 아파치 라이선스와 충돌하는 소프트웨어를 사용하거나 바이너리로만 제공되는 몇 가지 플러그인(예: 네트워킹, 저장소)을 가지고 있다. 이러한 플러그인은 아파치에서 배포하지는 않지만, noredist 플래그를 사용하면 사용자가 빌드할 수 있다(CentOS의 경우 package.sh -p noredist).