

CloudStack을 활용한 코딩 교육 지원 환경 개발

201645801 강다미

201324416 김동환

201724633 Kamila Makhmudova

Team: 바다

dndb44@naver.com 강다미

goddh94@gmail.com 김동환

kamila.mqa@gmail.com Kamila

정보컴퓨터공학부 부산대학교

Dept. of Computer Science & Engineering, Pusan National University

2020년 6월 12일

지도교수 : 염 근 혁 (인)

요 약

본 문서는 부산대학교 정보컴퓨터공학부 졸업과제 (CloudStack을 활용한 코딩 교육 지원 환경 개발)에 대한 착수 내용을 담고 있다. 본문의 내용은 과제의 목표, 대상 문제 및 요구조건 분석, 현실적 제약사항 분석 결과 및 대책, 설계문서, 추진 일정, 구성원간의 역할 분담을 담고 있다.

주제어: 클라우드스택, 클라우드 컴퓨팅, 코딩 교육 환경

1. 과제의 목표

1. 과제 선정 배경

최근 코딩교육에 대한 중요도가 높아지며 많은 사람들이 코딩과 코딩 교육에 관심을 가지고 있다. 코딩을 통해 프로그램을 개발하기 위해서는 프로그램이 동작할 수 있는 개발환경을 구축해야 한다. 사용자마다 다양한 컴퓨터 시스템을 가지고 있으므로 그에 적합한 개발환경을 구축하는 것이 필수적이다. 하지만 개발환경 구축을 위한 프로그램들의 요구사항이 다르고 제약사항이 있어 개발환경을 구축하는 과정이 복잡하고 시간이 많이 소모된다.

따라서 본 졸업과제는 컴퓨팅 자원의 가상화를 지원하는 클라우드 서비스(CloudStack)를 이용하여 개발환경을 구축하고 공유하는 시스템을 통해 별도의 개발환경 재구축 없이 시스템 환경에 제약을 받지 않는 웹 기반 코딩 실습 환경제공을 목표로 한다.

2. 기능적/기술적 목표

- 1) 가상화 플랫폼을 이용하여 별도의 개발환경 구축 없이 웹에서 개발이 가능한 시스템을 제공한다.
- 2) CloudStack을 통하여 가상화 서비스를 제어할 수 있다.
- 3) 다수의 사용자들이 동일한 환경에서 개발을 진행할 수 있는 클라우드 기반의 실습환경을 구축한다.
- 4) 사용자를 선생님과 학생으로 구분해서 선생님이 학생에게 코딩 교육을 지원할 수 있는 환경을 제공 한다.
- 5) 깃허브와 연동을 통해 개발 과정을 효율적으로 관리할 수 있게 한다.

2. 대상문제 및 요구조건 분석

2.1 유사시스템 분석



NeoReason사의 Repl.it은 웹에서 코드를 작성하고 컴파일할 수 있는 클라우드 기반 개발환경 툴이다. 언어마다 기본적인 컴파일러를 제공하고 사용자가 원하는 개발환경을 구축해서 시스템 환경 제약 없이 웹 브라우저에서 코딩이 가능하다.

2.2 문제점 분석

- 1) 다른 프로젝트에서 사용했던 개발환경을 재사용할 수 없기 때문에 새로운 프로젝트 생성 마다 개발환경 구축이 필수적이다.
- 2) 자신이 구축한 개발환경을 다른 사람과 공유할 수 없어 협업자들이 같은 개발환경에서 작업을 하고자 할 때 매번 개발환경을 재구축해야 한다.
- 3) Repl.it에서 제공되는 컴파일러 버전만을 사용해야하기 때문에 사용자가 원하는 버전에서의 컴파일이 불가능하다.

-예로 C언어의 경우 현재 gcc version 7.5.0을 사용하고 있는데 최근 릴리즈 버전은 9.3이다. 사용자가 원하는 컴파일러 버전을 선택할 수 없는 제약이 존재한다.

2.3 유스케이스 분석

1) 유스케이스 다이어그램

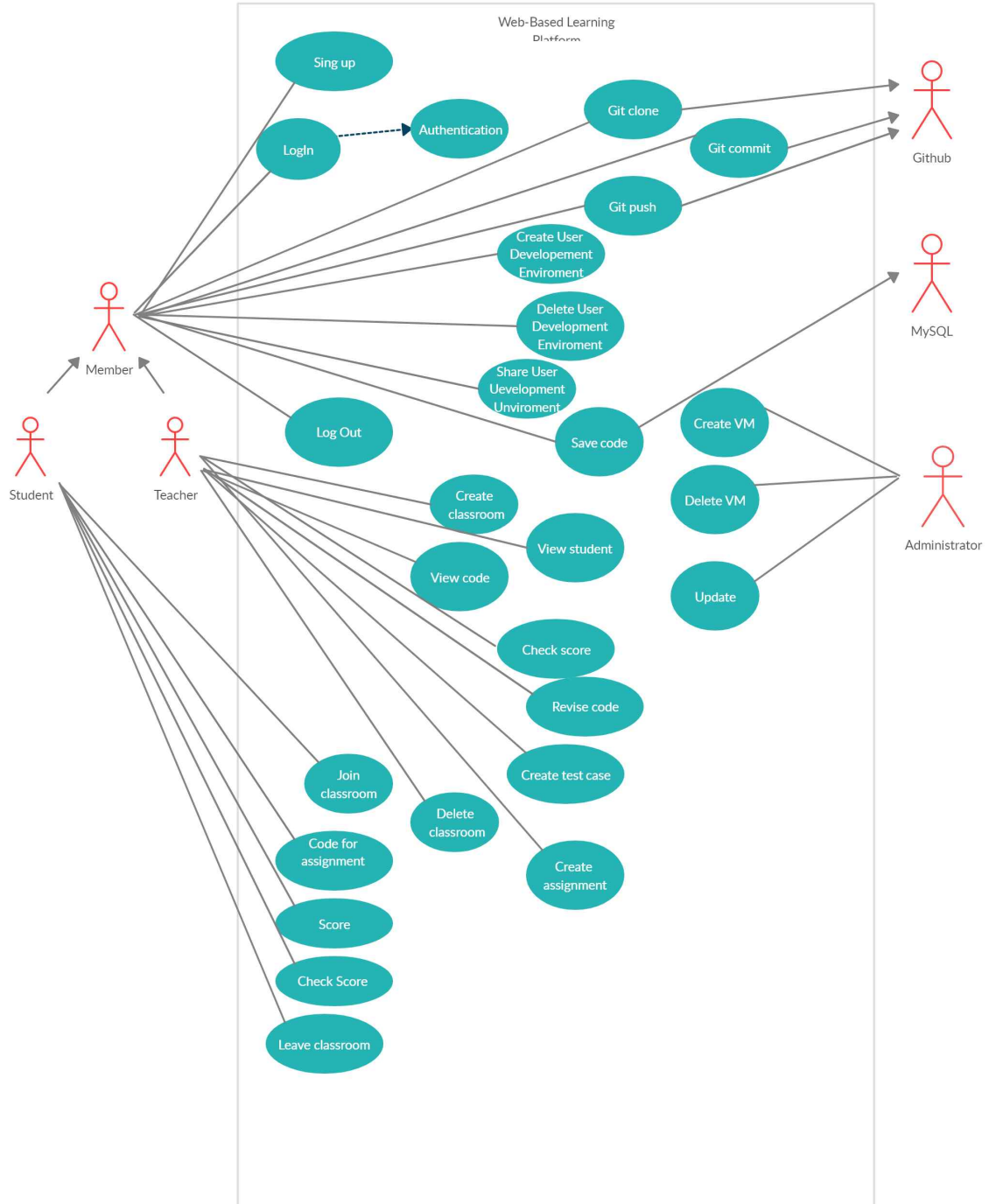


그림 1 유스케이스 다이어그램

2) 액터 및 유스케이스

[표 1] 액터 설명

Actor 명	설명
Student	학생으로 등록하고 로그인한 사용자
Teacher	선생님으로 등록하고 로그인한 사용자
Administrator	VM관리, 업데이트 작업을 수행하는 관리자
MySQL	사용자의 코드가 저장되는 데이터베이스
Github	사용자가 코드를 clone, commit, push하는 github 웹 사이트

[표 2] 유스케이스 설명

Usecase명	설명
Sign up	사용자는 학생 또는 선생님으로 시스템에 회원 가입할 수 있다.
Log in	회원은 인증 시스템을 통해 시스템에 로그인할 수 있다.
Log out	회원은 시스템에서 로그아웃할 수 있다.
Authentication	사용자의 아이디와 비밀번호로 본인을 확인한다.
Git clone	Github에 있는 소스코드를 clone할 수 있다.
Git commit	수정된 소스코드를 github로 commit할 수 있다.
Git push	Commit한 내용을 github로 push할 수 있다.
Create User Development Environment	사용자는 개발환경을 구축하고 하나의 유저 정의 개발환경으로 생성할 수 있다.
Delete User Development Environment	사용자는 자신이 생성한 개발환경을 삭제할 수 있다
Share User Development Environment	사용자는 자신이 생성한 개발환경을 공유하거나 다른 사용자가 생성한 개발환경을 사용할 수 있다.
Save code	사용자는 작성 중인 자신의 코드를 데이터베이스에 저장할 수 있다.
Create classroom	선생님은 학생들을 관리하는 클래스룸을 생성할 수 있다
Delete classroom	선생님은 클래스룸을 삭제할 수 있다.
View student	클래스룸에 입장한 학생을 조회할 수 있다
Check score	클래스룸에 있는 학생들의 과제별 성적을 조회할 수 있다.

View code	클래스룸에 있는 학생들이 작성한 코드를 조회할 수 있다.
Revise code	학생들이 작성한 코드를 수정할 수 있다.
Create assignment	학생들을 위한 문제를 생성할 수 있다.
Create test case	채점을 위해 문제마다 테스트 케이스를 생성할 수 있다.
Join classroom	학생은 선생님이 생성한 클래스룸에 입장할 수 있다.
leave classroom	학생은 클래스룸에서 퇴장할 수 있다.
Code for assignment	학생은 선생님이 생성한 문제를 풀 수 있다.
Score	학생 코드의 결과와 선생님이 생성한 테스트 케이스를 비교해서 점수를 채점한다.
Check Score	학생은 자신이 푼 문제에 대한 점수를 확인할 수 있다.
Create VM	관리자는 VM을 생성할 수 있다.
Delete VM	관리자는 VM을 삭제할 수 있다.
Update	관리자는 C언어 컴파일러 버전을 업데이트할 수 있다.

3) 요구 사항 분석

[표 3] 기능적 요구사항

기능적 요구사항	
사용자 개발환경	회원이가입을 통해 서비스에 등록되고 인증을 통해 로그인할 수 있다.
	C언어 IDE 인터페이스를 통해 코드를 작성하고 실행시킬 수 있다.
	개발환경을 구축해서 하나의 유저 개발환경으로 생성할 수 있고 생성된 개발환경을 삭제하거나 배포해서 다른 사람과 공유할 수 있다.
	선생님은 클래스룸을 생성하고 삭제할 수 있다. 클래스룸에서는 학생 조회, 성적 조회, 코드 조회와 코드 수정 기능을 제공한다.
	선생님은 문제, 문제에 대한 Test case를 생성할 수 있다.
	학생은 클래스룸에 입장하고 퇴장할 수 있다. 클래스룸에서는 문제를 풀 수 있고 Test case와 본인의 결과를 비교한 채점결과를 제공받으며 각 문제마다 성적을 확인할 수 있다.
	Github 아이디와 연동하여 github에 있는 코드를 clone할 수 있고 수정사항에 대해 commit, push 기능을 제공한다.
	사용자가 작성한 코드는 code save를 통해 클라우드 데이터베이스에 저장된다.
관리자 메뉴	관리자는 VM을 생성하고 삭제할 수 있고 C언어 컴파일러의 버전을 업데이트할 수 있다.

[표 4] 비기능적 요구사항

비기능적 요구사항	
사용성	사용자가 사용하는 서비스와 소비하는 클라우드 컴퓨터 자원의 양을 추적하고 그에 맞는 클라우드 컴퓨터 자원을 동적으로 할당해서 클라우드 시스템의 QoS(Quality of Service)를 보장한다.
제약 사항	서버환경: CloudStack
	커맨드 라인툴: CloudMonkey
	서버OS: CentOS7
	클라우드 DB: MySQL
	Web 개발 환경: Angular JS
	Rest-like API를 통한 웹서비스 연동
	Docker를 통한 이미지 배포 및 활용

3. 현실적 제약 사항 분석 결과 및 대책

문제 1. 컨테이너 구성 시 컨테이너가 호스트OS의 주요 자원에 접근 가능하도록 설정된 경우 컨테이너에서 호스트 OS의 주요자원에 대한 수정으로 인해 해당 호스트 및 호스트에서 실행되는 다른 컨테이너의 안정성과 보안에 영향을 미칠 수 있다.

대책: 주요 시스템 디렉터리 마운트 금지(각 컨테이너에 매핑된 디렉터리의 목록과 권한 확인), 컨테이너 내 불필요하게 접근 가능한 호스트 장치의 존재여부 및 필요에 의해 접근 가능한 장치에 대한 권한이 올바르게 설정되어 있는지 확인할 수 있는 방법을 찾아본다.

문제 2. 다양한 언어에 대한 컴파일러를 모두 제공하는데 한계가 있다.

대책: 초기 사용자에게 지원하는 언어를 C언어로 제한하고 추후 업데이트를 통하여 다른 언어를 추가할 수 있는 방안을 고려한다.

4. 설계문서

4.1 개발 환경

Windows 64bit

CentOs 7 (VM)

CloudStack 4.14

CloudMonkey 6.0.0

Docker

4.2 사용 기술

1) CloudStack



Apache CloudStack은 오픈 소스 형태의 IaaS(Infrastructure-as-a-service) 플랫폼으로 서버 공급, 네트워크 관리, 저장소 할당을 통한 공용 또는 사설 클라우드를 구축하는데 적합하다.

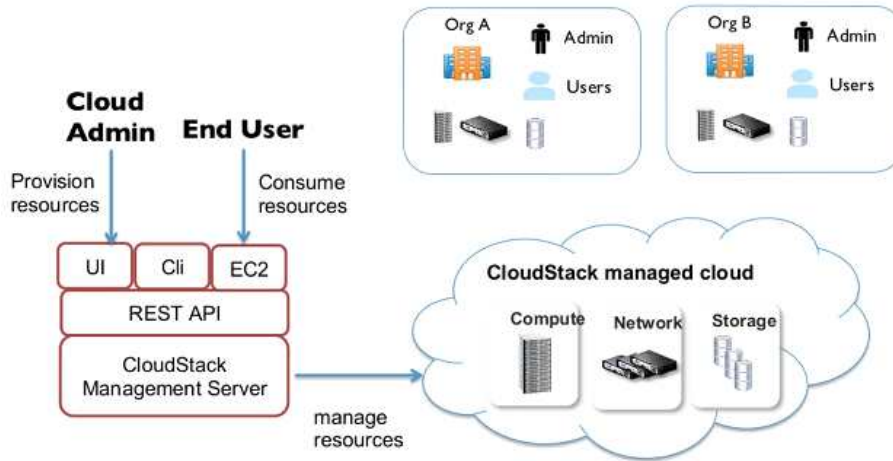


그림 2 CloudStack 환경

그림 2와 같이 IaaS 클라우드 환경을 구축할 수 있도록 하이퍼바이저 호스트에 관리서버와 에이전트를 제공하며 클라우드 관리를 위한 친숙한 웹 기반 UI를 제공한다. 또한 Command line tools, full-featured 쿼리 기반 API를 사용하여 클라우드를 관리할 수 있다.

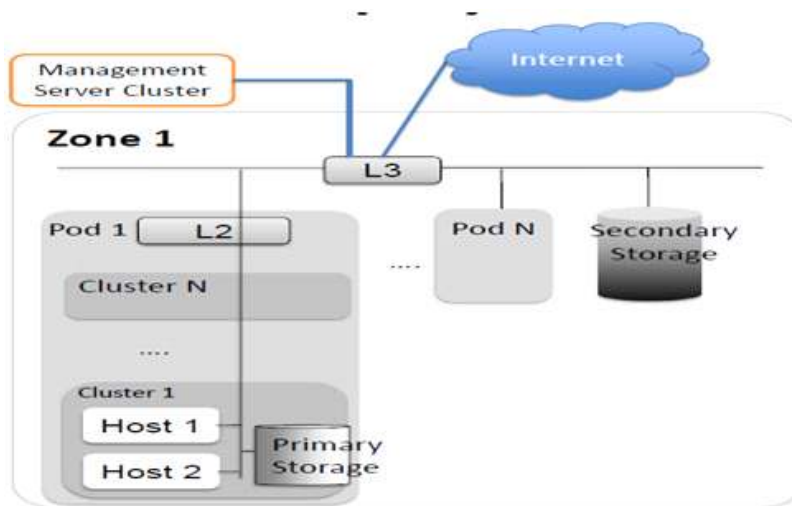


그림 3 Deployment Architecture

그림 3은 CloudStack의 Deployment Architecture이다. 클러스터는 동일한 하이퍼바이저의 하나 이상의 호스트로 구성되는데 클러스터의 모든 호스트는 Primary storage에 접근할 수 있다. Pod는 L2 스위치를 가지며 하나 이상의 클러스터로 구성된다. 가용 Zone은 하나 이상의 Pods를 가지고 Secondary storage에 접근할 수 있다. 관리 서버 클러스터는 물리적 서버 또는 VM으로 배포가능하며 단일 관리서버 노드는 최대 10,000개의 호스트까지 관리할 수 있다.

2) CloudMonkey

CloudMonkey는 CloudStack용 CLI(Command Line Interface)도구이다. CS의 구성 및 관리를 단순화하는 command line tool 그리고 interactive shell의 용도로 쓰인다. CloudStack을 API로 작업할 때 편하게 이용할 수 있다.

3) Docker

도커는 OS나 DB, 다양한 프로그램 그리고 실행환경까지 컨테이너로 이미지화하며 이미지의 배포와 관리에 뛰어난 장점을 가진 플랫폼이다. CloudStack에서 VM의 배포와 공유 기능 구현에 어려움이 있을 때 대체제로 활용 가능하다.

5. 추진 일정(A: 공통, K: Kamila, D: 강다미, H: 김동환)

[표 5] 추진 일정

수행내용 \ 기간	6월			7월				8월				9월				
	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
CloudStack 설치 및 API 사용방법 학습	A															
CloudStack 프레임워크 구축			H													
개발환경 공유 시스템 구축			K													
클래스룸 기능 구현			D													
GitHub 연동시스템 구축								H								
C언어 컴파일 환경 구축								K								
테스트 및 보완											A					
웹 대시보드 구축				A												
보고서						중간 보고서								최종 보고서		

6. 역할 분담

[표 6] 역할 분담

이름	역할 분담
Kamila	<ul style="list-style-type: none">- 유저 정의 개발환경을 생성, 삭제, 공유하는 시스템 구축- C언어 컴파일 환경 구축
강다미	<ul style="list-style-type: none">- 클래스룸 기능 구현
김동환	<ul style="list-style-type: none">- CloudStack 프레임워크 구축- Github 연동시스템 구축
공동	<ul style="list-style-type: none">- CloudStack 개발환경 구축 및 API 사용방법 학습- 웹 대시보드 구축- 기능, 모듈에 대한 테스트 및 보완- 보고서 작성