

# 클라우드 컴퓨팅 기반의 협업 이미지 제작 도구

임 양 미<sup>†</sup>

## The Collaborative Image Editing Tool based On the Cloud Computing

Yang Mi Lim<sup>†</sup>

### ABSTRACT

In recent times, IaaS (Infrastructure as a Services) have been rapidly evolving to allow developers to easily and efficiently access work in the server and network areas for development of a web of App based on cloud computing. In this study, we developed the collaborative image editing tool App based on Cloud-computing, by adopting AWS of representative company that develops IaaS. First, it is crucial to understand various situation conditions for representative infrastructure services: AWS, Azure and Google (GCP). This may have the effect of reducing manpower and development time, but as each company has different policy and technical support, we need a new study every time the environment changes of infrastructure services. We tried to develop a hybrid-App so that users with various devices can collaborate work each other by utilizing the infrastructure service AWS through the process of developing the image editing authoring tool based on the cloud computing. The future studies should continue about compatibility issues and support issues in order to minimize the problems of overseas infrastructure services, but we think that domestic cloud computing policies and developments should be urgently considered.

**Key words:** Infrastructure Services, Collaborative Tool, Cloud Computing, Hybrid-App, AWS

### 1. 서 론

최근에 클라우드 컴퓨팅 환경 기반의 서버 가상화 기술이 급속도로 발전하고 있으며, 서버 공유 제공을 위해서 오픈소스 및 상용 프로그램들의 지원이 많아지고 있는 추세이다. 클라우드 컴퓨팅 환경 개발이란 분산컴퓨팅으로 사용자의 요구에 맞는 자원을 사용자 요구에 맞게 확장하고 유연성 있게 지원하여 여러 사람이 실시간으로 공동의 소프트웨어를 활용하여 작업 분산화를 이루어 작업의 효율성을 높이기 위한 기술개발이다. 이러한 특징으로 인해 여러 사용자가 공동으로 작업할 수 있는 환경이 조성되어 혼자서

작업하기 어려운 콘텐츠를 협업을 통해 완성도 높은 콘텐츠를 제작할 수 있다 [1-2].

국내외에서 클라우드 컴퓨팅을 활용한 교육 시스템 구축은 PC의 운영 및 관리의 효율성을 높여 많은 사람들이 적극적으로 참여하고 있다. 2012년 6월에 교육과학기술부에서 추진한 연구를 보면 디지털교과서 개발 및 적용, 온라인 수업활성화, 교원의 스마트교육 역량 강화, 시·도 교육청 스마트교육 활성화, 클라우드 교육서비스 기반 조성 등 전반적인 교육의 시스템을 클라우드 컴퓨팅 환경에서 구현하고자 하는 지속적인 노력들이 있었음을 볼 수 있다. 하지만 많은 시도들에 비해 뚜렷한 결과가 없는 이유는 클라

\* Corresponding Author : Yang Mi Lim, Address: (132-714) 33, 144-gil, Samyang-ro 680, Dobong-gu, Seoul, Korea, TEL : +82-2-901-8350, 010-3208-0435, FAX : +82-2-901-8646, E-mail : yosimi@duksung.ac.kr

Receipt date : Jul. 17, 2017, Approval date : Aug. 1, 2017  
<sup>†</sup> Dept. of Digital Media, Duksung Women's University  
\* This research was supported by the Duksung Women's University Research Grant 2017.

우드 기반 기술 확보 부족이 가장 큰 이유이다. 특히 우리나라의 경우 해외 플랫폼 SW의 의존도가 높은 이유도 무시할 수 없다. 또한 공공기관 및 기업이 클라우드 시스템을 사용하기 위해서는 법적 절차들이 존재하고 있기 때문에 개발 이후에도 상용화까지 많은 시간이 필요하다. 마지막으로 클라우드 적용 범위가 좁은 편이고 대규모의 인프라 구축 또한 무시할 수 없는 이유 중 하나이다[3]. 이러한 한계점으로 인해 국내에서 사용되는 클라우드 기반 협업 콘텐츠 서비스 소프트웨어들은 다소 존재하고 있지만, 콘텐츠 저작 도구와 같은 소프트웨어는 거의 전무한 상태이다. 협업의 저작 도구는 여러 사용자가 동시에 접속할 수 있는 가상머신들의 네트워크 형성과 상호작용을 위한 소프트웨어 스택 구현 단계, 사용자들의 작업 환경 및 실시간 데이터 처리 단계, 다양한 디바이스 환경이 협업 도구 환경에 쉽게 접근할 수 있도록 하이브리드 기술 개발 단계 연구가 필요하다.

본 연구에서는 소규모 팀에서 공동으로 작업할 수 있는 AWS 기반의 공동작업공간을 만들어 교수자와 학습자가 공동의 실습 결과를 쉽게 접근할 수 있도록 하는 클라우드 기반의 이미지 저작 도구 앱을 개발한다. 본 저작 도구는 프로젝트 환경을 자유롭게 생성할 수 있고 다양한 스터디 원들을 초대하여 어디에서든지 장소에 구애 받지 않고 상호작용이 가능한 프로젝트 작업을 진행할 수 있는 것이 특징이다. 본 논문의 구성은 2장에서 클라우드 컴퓨팅 주요 기술에 대해 설명하고, 3장에서는 공동 협업 저작도구 구현, 4장에서는 구현 환경에 대한 결과를 제시하고 5장에서 결론을 맺는다.

## 2. 관련연구

공동 협업 저작 도구로 대표적인 것은 구글의 인터넷 기반의 문서, 스프레드시트, 프레젠테이션 등이 있다. 이들은 서버에 프로그램을 두고 다양한 단말기에서 사용할 수 있는 웹 또는 앱에 기반을 둔 소프트웨어 서비스이다. 이와 같은 클라우드 컴퓨팅 기반의 웹 또는 앱을 개발하기 위해서는 상당히 여러 가지 기술이 필요한데 개발자들의 서버 및 네트워크 분야의 작업에 쉽게 접근할 수 있도록 인프라서비스(IaaS) - 플랫폼 서비스(PaaS) - 어플리케이션(application, App) 서비스(SaaS) 형태로 분류된다[4]. 2005년 이후, 클라우드 컴퓨팅의 대부분의 내용들은

SaaS(software as a service)에 집중되어 있었으나, 2008년부터는 IaaS(infrastructure as a service), PaaS(platform as a service)로 그 영역을 넓혀가고 있다 [5-6]. IaaS를 개발하는 대표업체는 AWS, 마이크로소프트 애저, 구글클라우드 등이 있다. 알담(Motaseem Aldiab)은 이들의 컴퓨팅, 스토리지, 네트워킹 및 가격/모델 기능에 대해 Table 1에서 자세히 설명하고 있다 [7]. 또한, 가트너도 엔터프라이즈급 IaaS 퍼블릭 클라우드 공급업체의 필수 기능으로 분류한 기준에서 AWS는 92%를 충족, 마이크로소프트의 경우 88%와 70%라는 발전을 보인다고보고 했다[8].

본 연구에서는 Table 1을 기반으로 분석한 결과를 토대로 Ragions의 확장성, 하이브리드 클라우드 지원, 초기 투자 비용이 저렴하게 드는 AWS를 채택하였다. 앱 개발을 위해서는 하이브리드앱(hybrid App)을 선택하였다. 하이브리드앱은 모바일 웹/앱(mobile Web/App)의 단점을 해결하기 위해 네이티브 앱(Native App)으로 모바일 웹/앱을 컨테이너 형식처럼 포장하는 기술이라 할 수 있다. 해당 도구로는 폰갭(phoneGap), 앱스프레소(appspresso), 티타늄(titunum) 등이 있다. 하이브리드를 지원하는 Appspresso가 더 이상 업데이트를 지원하지 않아 대부분 개발자들은 PhoneGap(cordova)과 Sencha Touch를 이용하여 UI를 개발하였지만, 파일과 Full JS 스택 개발에 한계가 있다. 그 외에 AngularJS 라는 웹 어플리케이션 프레임워크도 있었으나, 이것도 모바일 앱에 최적화된 UI개발에 한계가 있다. 본 연구에서는 ionic을 채택하여 개발하였는데 ionic은 HTML5 API를 사용할 수 있는 컴포넌트들과 클래스 플랫폼을 개발하는 Cordova 기반의 하이브리드 앱 개발을 지원하는 프레임워크이다. 모바일앱 개발에 HTML을 활용하여 UI제공하며, JavaScript의 데이터 업데이트가 빠르고 Node.js 기반의 환경 제공, MVC 개발 환경 지원, Plugins 허용, 크로스 플랫폼 빌드 지원 등이 가능하다[9-12].

## 3. 제안한 어플리케이션의 구현과 방법

### 3.1 가상화 환경 구현

본 연구는 클라우드 컴퓨팅 환경을 기반으로 협업 소프트웨어를 구축하는 것을 목적으로 하고 있다. 클

Table 1. AWS vs Azure vs Google: Compute, Storage and DB, Networking and Pricing/ Models

		AWS	Azure	GCP
compute	Instance Families	7	4	4
	Instances types	38	33	18
	Regions	yes	yes	yes
	Zones	yes		yes
Storage and databases	Ephemeral (temporary)	yes	Temporary Storage-D Drive	yes
	Block Storage	EBS	Page Blobs	Persistent disks
	Object Storage	S3	Block Blobs and Files	Google Cloud Storage
	Relational DB	RDS	Relational DBs	Google Cloud SQL
	Archiving	Glacier		Nearline
	NoSQL and Big Data	DynamoDB, EMR, Kinesis, Redshift	Windows Azure Table, HDInsight	Cloud Datastore, Big Query, Hadoop
Networking	Virtual network	VPC	subnet	VNet
	Public IP	yes	yes	yes
	Hybrid Cloud	yes		yes
	DNS	Route 53		
	Firewall/ACL	yes	yes	yes
Pricing and Models	Pricing	Per hour - rounded up	Per minute - rounded up commitments (pre-paid or monthly)	Per minute - rounded up (minimum 10 minutes)
	Models	On demand, reserved, spot	On demand - short term	On demand - sustained use

클라우드 기반의 교육 실습 환경을 구축하기 위해서 다음과 같은 기능들을 해결한다. 첫째, 다수의 사용자가 동시 접속했을 시 서버부하를 예방하기 위한 가상머신들의 네트워크 형성이 필요하다. 네트워크 형성 후에는 가상머신들 간의 상호작용을 위해 소프트웨어 스택 구현이 필요하다. 이것의 역할은 시스템의 요구사항인 어플리케이션, 라이브러리, 서비스, 데이터, 운영체제와 같은 정보를 담은 이미지 형태의

컨테이너와 같은 역할을 하게 된다. 두 번째로 교육용 실습환경이 작동되기 위해서는 자원의 생성, 동작, 대기, 회수, 초기화 등을 위한 일괄적인 도구가 필요하다. 또한 이곳에서 발생하는 실습데이터 및 가공된 산출물 데이터를 보관하고 관리하기 위한 공용의 파일시스템도 필요하다. 셋째로 사용자들이 플랫폼에 제한을 받지 않고 다양한 디바이스에서 가상환경 기반의 교육 실습환경에 접근 할 수 있도록 하이

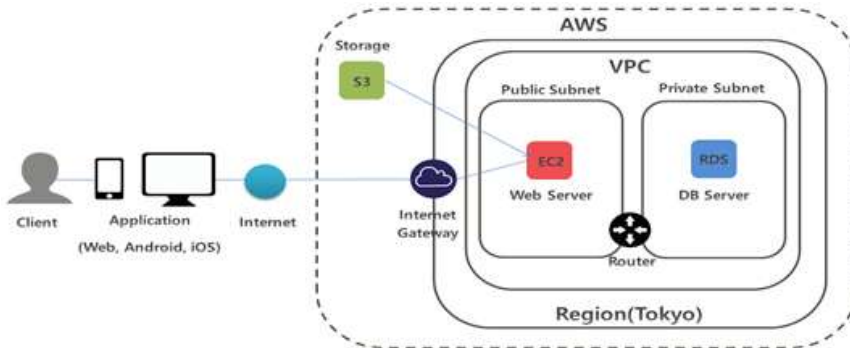


Fig. 1. AWS system structure.

Table 2. The development tools for proposed Application

	Develop Tool
Server	Node.js, Angular.js
Database	MySQL
Client	Cordova, ionic
OS	Ubuntu
Environment	AWS(Amazon Web Services)
Management	Git, Github

브리드 기술이 요구된다. Fig. 1은 이러한 전체적인 요구에 따른 가상화 실험 환경 구성도이다.

AWS는 컴퓨팅 파워, 데이터베이스 스토리지, 콘텐츠 전송 및 기타 기능을 다양하게 제공하는 VPC, EC2, RDS, S3를 사용하여 시스템구조에 사용하였다 [13].

Amazon의 EC2(Elastic Compute Cloud)는 클라우드 컴퓨팅, 클러스터 GPU, 고용량 메모리 클러스터 인스턴스로 고성능 네트워크 기능을 제공하도록 설계되어 있으며, 프로그램 방식을 통해 클러스터에 실행할 수 있어 저지연 네트워크 상황에서도 노드 간에 필요한 통신을 가능하도록 지원한다. AWS 환경을 구축하기 위해 Ubuntu 운영체제를 사용하였고, MySQL 엔진 기반의 RDS인스턴스로 연동하여 DB를 구축하였다. 또한 실시간 기술을 구현하기 위해 “NodeJS” 언어를 사용하였고, 다양한 기기의 접근을 위해 하이브리드 앱 구축에 최적화 된 AngularJS, cordova, ionic과 같은 언어와 툴을 사용하였다 [14]. 마지막으로 프로젝트 유지보수와 배포 용이성을 위해 Git를 사용하였다.

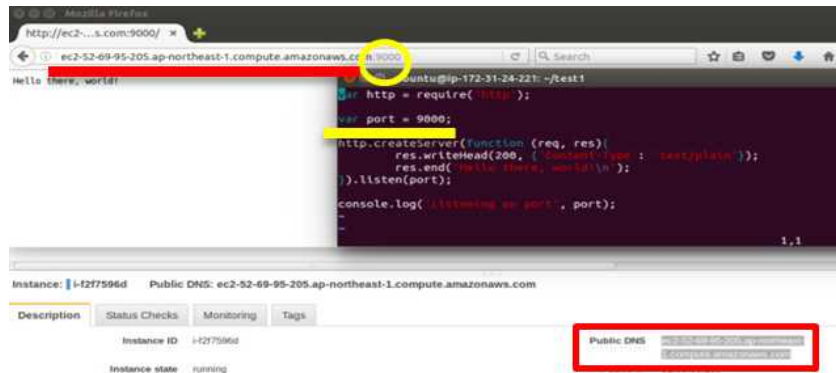


Fig. 2. EC2 instance connection verification.

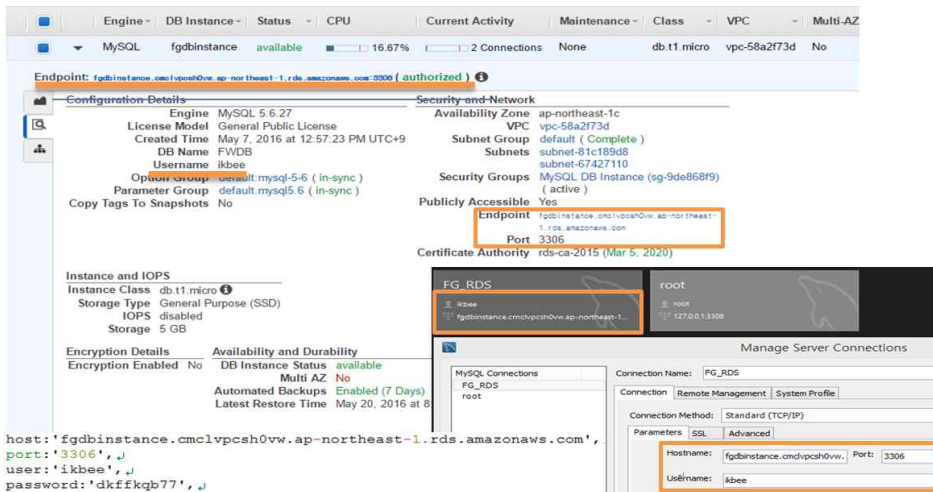


Fig. 3. AWS RDS : MySQL connection.

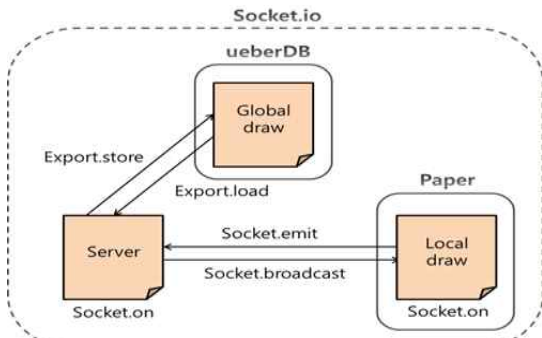


Fig. 4. Real-time data change flow chart.

### 3.2 제작환경 구현

제안한 공동협업 저작 도구는 앞 서 논의에 기반을 두어 구현하였으며, 이에 사용된 개발도구들은 다음과 같다.

AWS는 재난예방 및 고가용성을 위해 주요 지역에 Region을 위치시킨다. AWS를 사용함에 있어서 가장 먼저 가상 네트워크를 형성해야 하는데 이때 사용되는 것이 VPC(virtual private cloud)이다. VPC는 아마존 클라우드 내에서 private ip를 사용하는 일종의 가상 private network를 만들어줄 수 있는 서비스이다. 이 서비스 전에는 EIP 외에 정적 서비스를 사용할 수 없으며, 또한 10.0.x.x와 같은 private ip를 사용할 수 없었다. VPC 서비스와 함께, 내부 ip 대역을 사용할 수 있게 되며 조금 더 유연한 네트워크 관리가 가능하게 된다. VPC안에서

EC2 인스턴스, RDS DB 인스턴스, ElasticCache 클러스터 등을 생성하고 사용한다. VPC 안에는 서브넷을 여러 개 추가할 수 있다. 여러 개로 나뉜 서브넷은 네트워크를 격리할 수 있고, 이 서브넷 간에 접근제어를 설정할 수 있다. 인터넷에서 접근해야 하는 웹 서버는 Public Subnet을 만들고, 외부접근이 필요 없는 데이터베이스 서버는 Private Subnet으로 구성한다. 이들의 구축은 보안에 유리하다.

가상 서버를 제공하는 EC2는 가상서버를 제공한다. EC2 기반의 가상 서버 구축은 보안 및 네트워크 구성과 스토리지 관리가 가능하다. EC2에서 생성한 가상 서버를 각각의 객체로 보고 인스턴스라고 하는데, 하나의 인스턴스는 하나의 가상 컴퓨팅 환경이 된다. EC2 서비스를 사용하려면 먼저 VPC의 Public Subnet에서 EC2 인스턴스를 생성하고 포트 설정을 한다. 키 페어를 생성하는 과정에서 키 페어(key pair) 이름을 설정하고 키 페어 파일(.pem)을 저장 관리해야 한다. 키 페어 파일(.pem)은 EC2 인스턴스에 접속 시에 사용된다. Fig. 2는 인스턴스가 연결됐는지 확인하는 과정이다.

다음은 ssh로 접근과정이 남아있다. 기본적으로 Linux와 Ubuntu 운영체제 컴퓨터에는 ssh 클라이언트가 포함되어 있어 별도의 설치 과정 없이 명령줄에 ssh를 입력하여 ssh 클라이언트를 사용하여 인스턴스에 연결한다. 이때 키 페어(.pem)파일과 user\_name@public\_dns\_name을 지정한다. Fig. 3의 박스가 입력된 Public DNS로 지정된다. 인스턴스 안에

The image shows a multi-part screenshot. On the left is the '회원가입' (User Registration) form with 'google' entered in the email field. In the center is a table of IAM users:

usertype	userid	password	salt	nickname	userprofileimage	online
local	dimmon	f8D+2VU67...	Mp4EFR...	fg-team	https://s3-ap-northeast...	Y
local	duksung	emCetj+sfy...	blq4/a1L...	hello	https://s3-ap-northeast...	Y
local	gt	CS7kqRlccjX...	5/63k9d...	adna	https://s3-ap-northeast...	N
local	google	5yFefnasTIT...	Q8j7Umj...	korea	https://s3-ap-northeast...	N
local	kakao	LhCorwKALC...	ItOn355V...	daum	https://s3-ap-northeast...	N
local	line	ypSkWESopH...	OxsSLZC...	naver	https://s3-ap-northeast...	N
local	pki	3g7So5/L2bJ...	TZulmAg...	pig	https://s3-ap-northeast...	N
local	roqte1017	LOtCu26RND...	6w57WVJ...	jungmin	https://s3-ap-northeast...	Y
local	zoocafe	6PJfw7v/11e...	z0v7DFR...	grape	https://s3-ap-northeast...	N

On the bottom left is the 'All Buckets / unibee / userprofile' directory listing, showing 'google\_profile.png' highlighted. On the bottom right is a code snippet for uploading a file to S3:

```

var s3 = new AWS.S3();
var params = {
  'Bucket': 'unibee/userprofile',
  'Key': 'fields.userid + '_profile.png',
  'ACL': 'public-read',
  'Body': fs.createReadStream(files.userprofile.path),
  'ContentType': files.userprofile.type
};
s3.upload(params, function(err, data) {
  if (err) throw err;
  console.log('success upload image to s3');
});
    
```

Fig. 5. Sign up & Registered DB table.

프로젝트를 생성하고 저장관리하게 된다. RDS(re-relational database service)는 클라우드에서 관계형 데이터베이스를 더욱 쉽게 설치, 운영 및 확장할 수 있다. S3(scalable storage in the cloud)에서 제공하는 단순한 웹 서비스 인터페이스를 사용하여 웹에서 언제 어디서나 원하는 양의 데이터를 저장하고 검색 가능하다. S3안에서의 DB Server는 MySQL 환경의 RDS를 의미한다. RDS안에 사용자 정보, 그룹방 정보, 멤버 정보와 같은 데이터가 저장되고, 사용자들이 협업으로 만들어낸 미디어 파일들이 저장된다.

Fig. 4는 사용자들이 이미지 작업을 하고 데이터가 생성, 동작, 대기, 저장, 취소, 초기화 등의 데이터 흐름과 NodeJS의 paper와 socket.io 모듈을 사용하여 실시간으로 변경사항을 반영하고 데이터를 저장

하는 구조를 보여준다. 실시간의 느린 반응 속도를 해결하기 위하여 출력과 저장이 분리되어 동작하도록 하였다. 즉, Local draw에서 작업된 내용을 클라이언트 측에서 실시간으로 출력하되, 저장은 서버 측에서 Global draw를 통해서 저장되도록 하였다. 이 과정에서 방대한 path정보를 저장하는데 ueberDB를 사용하였고 이로 인해, ueberDB는 방대한 데이터를 쓰고 캐쉬로 빠르게 읽을 수 있게 되어 속도의 효율성을 높일 수 있다[15].

#### 4. 구현 결과

본 장에서는 구현된 공동협업 저작 도구의 단계별 결과 화면에 대해 회원 가입, 로그인, 뷰바인딩, 방(Bee)에 입장하고 방 관련 데이터 불러오기, 방 멤버

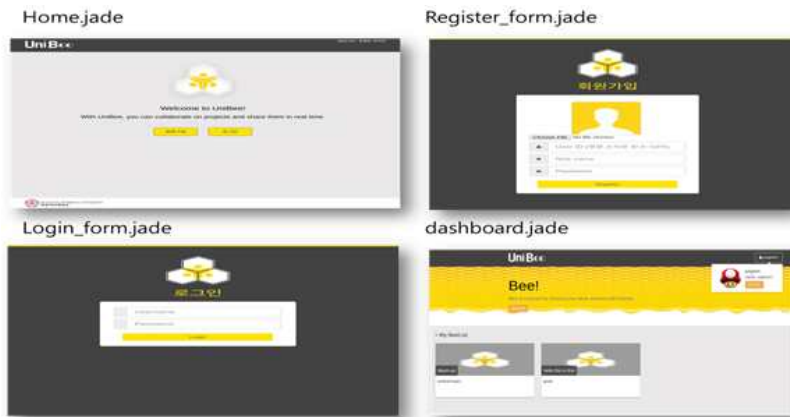


Fig. 6. Log in and room creation result.

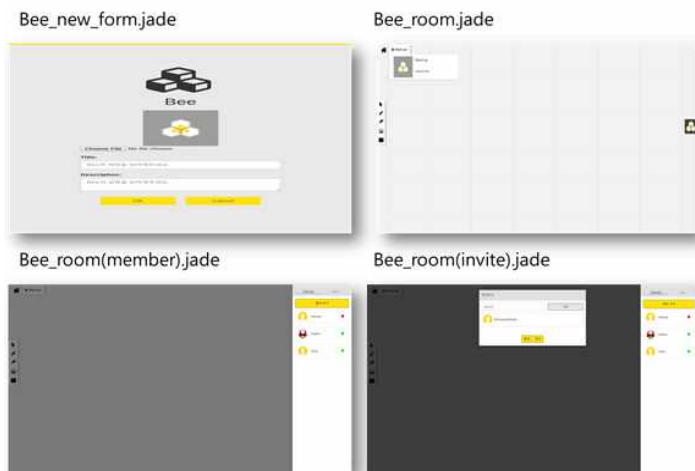


Fig. 7. Entry to study-room and add to friends.

추가하기, 그리기 순으로 설명한다.

회원가입 단계에서는 formiable이라는 모듈을 사용해 사용자가 form에서 입력한 데이터들을 가져온다. formiable을 사용하지 않았을 경우, form에서 보내준 local 이미지 파일의 경로가 너무 커서 서버에서 받아오지 못한다. 다음 Fig. 5는 form으로부터 받은 데이터를 서버에서 처리하는 코드이다. Formiable은 form.parse(error, fields, files)에서 fields 인자를 사용하여 html상에 지정한 name으로 접근할 수 있다. 예를 들어, 사용자 아이디를 입력 받는 input의 name이 userid이면 formiable상에서 field.userid로 값을 받아 올 수 있다. 그 다음 사용자가 프로필 이미지를 지정했으면 AWS모듈을 통해 S3로 로드하고 아니면 default 이미지의 url을 콜백함수로 넘겨준다. 입력 받은 패스워드에 salt을 섞어 해시함수로 암호화한 후, 데이터베이스에 사용자를 등록한다.

Fig. 6의 로그인은 사용자 권한을 인증해주는 passport.js를 사용하였다. passport.js는 ID, 패스워드 기반의 인증을 기본으로 제공하고, 별도의 패키지 설치를 통해 Facebook, Google+, Twitter 등의 로그인을 지원한다. 추후 다른 서비스와 계정 연동을 위해 사용하였다. Passport는 Strategy라는 이름으로 로그인 인증을 구분하는데 이 프로젝트에서는 ID, 패스워드 인증하는 LocalStrategy을 사용하였다. 로그인 라우터에서 Local passport 설정을 해주면, 입력 받은 아이디와 비밀번호로 사용자의 유무를 판단한다. 해당 사용자가 있으면 Session에 등록하여 인증

된 로그인 상태를 유지시켜준다.

Fig. 7은 생성된 방(Bee)에 사용자가 입장하면 session에 저장된 사용자 ID와 입장한 방 ID를 socket을 통해 서버에 보낸다. 서버에서 socket 방을 나누고 사용자를 등록하게 되고 방 ID에 해당하는 project가 있는지 검사하고, ueberDB에서 해당 데이터를 project에 import 하여준다. 같은 방에 같이 작업할 친구(멤버 추가) 추가를 위해 멤버 초대 버튼을 클릭하면 대화 검색 창이 나타난다. 이 창에서 사용자 ID로 검색한 후 서버에서 추가 작업이 이루어진다. 이미 추가된 사용자가 여러 번 추가되지 않도록 검사 후 DB insert를 한다. 서버에서 추가 메시지를 받으면 클라이언트에서 list를 갱신한다. Fig. 8은 스터디 방에서 여러 사람이 공동으로 그림을 작업하는 결과 화면이다.

### 5. 결 론

지금까지 다양한 디바이스를 갖고 있는 사용자들 간의 공동 작업이 가능하도록 클라우드 컴퓨팅 기반의 협업 이미지 제작 도구 앱 개발에 대해 제안하였다. 서로 간에 그린 이미지 데이터를 저장 및 관리하는 방식과 작업하는 동안에 느린 반응 속도로 인해 다자간의 작업 걸친 현상이 발생하여 이를 해결에 대한 노력을 하였으나, 최적화된 환경 구축은 미흡하다고 판단한다. 향후 연구에서는 여러 사용자가 동시에 작업하는 동안 변경된 데이터의 저장, 관리에 대해 진행할 예정이며, 실제 응용 서비스의 연동을 통하여 모바일 클라우드 서비스를 위한 플랫폼 환경에

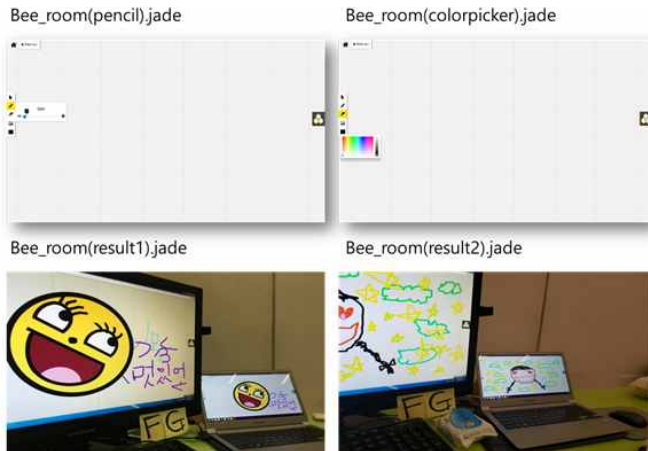


Fig. 8. Drawing collaboration result screen in study-room.

대해서도 연구하고자 한다. 또한 가상화 환경 기반의 앱을 구축하기 이전에 여러 상황 조건을 분석하고 파악하는 단계가 상당히 중요하다고 생각한다. 앞으로 IaaS, PaaS, SaaS 플랫폼 등이 개발될 때마다 웹과 앱 개발자들의 연구 방향은 상당히 복잡하고 넓어질 것이다. IaaS, PaaS, SaaS 플랫폼의 발전은 서버나 네트워크 등의 개발 범위가 축소되어 인력 및 개발 시간 등이 감소되는 효과가 있을 수 있으나, 플랫폼을 개발하는 기업들의 오픈 공유가 어디까지 일지는 의문스럽다. 네트워크, 스토리지, 서버, 가상화를 지원하는 AWS를 사용하면서 많은 시행착오와 IaaS를 개발하는 회사들마다 정책과 기술 지원이 달라서, 상황이 변경될 때마다 새로운 공부 요구 시간이 필요했다. 이를 최소화하기 위해서 국가의 클라우드 컴퓨팅 정책과 발전이 시급히 고려되어야 한다고 판단한다.

## REFERENCE

- [ 1 ] S.H. Park, H.J. Lee, S.H. Lee, K.R. Kwon, Y.K. Park, and K.D. Moon, "Collaboration Contents Fractal Service and Intrusion Detection Framework Based on Cloud," *Journal of Korea Multimedia Society* Vol. 20, No. 1, pp. 58-65, 2017.
- [ 2 ] H.J. Lee, K.R. Kwon, S.H. Lee, Y.K. Park, and K.D. Moon, "Design of Open Scenario Structure for Content Creation Service Based on User Defined Story," *Journal of Korea Multimedia Society*, Vol. 19, No. 2, pp. 170-179, 2016.
- [ 3 ] KERIS, *Feasibility Study on Strategic Planning to Build a Cloud Computing Environment for Smart Education*, KERIS Issue Report Research Data RM 2011-19, 2011.
- [ 4 ] D.W. Nurmi, R. Grzegorzczuk, and C. Obertelli, "The Eucalyptus Open-Source Cloud-Computing System," *Proceeding of 9th IEEE/ACM International Symposium, Cluster Computing and the Grid*, pp. 124-131, 2009.
- [ 5 ] B.Y. Lee, J. Park, and J.S. Yoo, "Software Architecture of the Grid for Implementing the Cloud Computing of the High Availability," *Journal of the Korea Contents Association*, Vol. 12, No. 2, pp. 19-29, 2012.
- [ 6 ] B.D. Lee and S.U. Shin, "Design of Secure Log System in Cloud Computing Environment," *Journal of Korea Multimedia Society*, Vol. 19, No. 2, pp. 300-307, 2016.
- [ 7 ] Cloud Academy Blog of Motasem Aldiab, <https://cloudacademy.com/blog/public-cloud-war-aws-vs-azure-vs-google> (accessed July, 16, 2017).
- [ 8 ] IDG Network, <http://www.ciokorea.com/news/> (accessed July, 16, 2017).
- [ 9 ] Adobe Systems Inc, <http://www.phonegap.com> (accessed July, 16, 2017).
- [ 10 ] Axway, <http://www.appcelerator.com/> (accessed July, 16, 2017).
- [ 11 ] SmartPhoneMart™, <https://www.smartphone-mart.net/> (accessed July, 16, 2017).
- [ 12 ] S.M. Han, Y.S. Son, and Y.S. Lee, "Design and Implementation of the Smart Virtual Machine for Smart Cross Platform," *Journal of Korea Multimedia Society*, Vol. 16, No. 2, pp. 190-197, 2013.
- [ 14 ] Amazon Elastic Compute Cloud (EC2), <http://aws.amazon.com/ec2> (accessed July, 16, 2017).
- [ 15 ] J.M. Kim, I.B. Park, and Y.M. Lim, "Real-Time Collaborative Tool Based on Cloud Computing," *Proceeding of Korea Multimedia Society Autumn Conference*, Vol. 19, No. 2, pp. 819-820, 2016.



임 양 미

1998년 큐슈대학교 정보전달학과 졸업(석사)  
 2009년 중앙대학교 첨단영상대학원 졸업(박사)  
 2010년~현재 덕성여자대학교 디지털미디어학과 교수

관심분야 : 멀티미디어, 인터랙티브아트, UX/UI etc.