전자정부에서 비용 효과적인 방법으로 양질의 서비스를 제공하는 서비스형 소프트웨어

매니쉬*, 석주업*, 박종서*

요약

전자정부시스템에서 사용자를 만족시킬 수 있는 유일한 방법은 저비용 고품질 서비스를 제공하는 것이다. 현재까지 서비스의 질은 "더 좋은 서비스를 위한 더 많은 투자"와 같이 투입 자금에 따라 달라졌다. 대부분의 서비스는 제정적인 제약에 따라 절충 되었고, 불확실한 재무구조 때문에 고품질 서비스 제공이 어려워 질 수 있다. 특히 글로벌 경제위기 이후 전자정부 재무구조의 불확실성은 심화되었고, 2008~9년간, 전 세계는 경제 위기에 봉착하게 되었고, 지금도 수많은 나라가 위기로부터 벗어나지 못하고 있다. 다른 국책사업과 마찬가지로, 전자정부 관련 사업도 이러한 위기에 상당한 영향을 받았고, 이는 정책 입안자와 이해관계자들에게 제정 문제에 따른 사업 지속성에 대해 진지하게 되새겨 보는 계기를 마련해 주었다. 본 논문에서는 최소한의 비용으로 양질의 서비스를 제공하기 위해 서비스형 소프트웨어를 사용하였으며, 또한 클라우드 환경을 구성하여, 실험을 수행하였다.

Software-as-a-Service in Providing Quality Services in Cost-effective Way in E-government

Manish Pokharel*, Sok Chuob*, Jong-Sou Park*

ABSTRACT

The quality service with the fewer price is only one way to satisfy the citizen in e-government system. So far, the quality depends upon the money like "Better quality if better money". Most of the services are compromised due to financial constraints. The unpredictable financial structures are always hurdles in providing quality service. E-government could not escape from such unpredictable financial structure especially in global economic crisis. In 2008-2009, entire world was under the shadow of economic crisis and till today many countries have not recovered yet. Like other project, e-government related projects are also deeply affected by the crisis. It has given a lesson to the policy makers and stakeholders to think seriously in this regards of sustainability in spite of economic crisis. We use Software-as-a-Service to provide quality service in a minimum affordable cost in this paper. At the end of this paper, we set up the cloud environment and perform the experiment on it.

Key Words : SaaS, Cloud Computing, E-government, Data Center, Cost-effective

* 경한국항공대학교 컴퓨터공학과(✉manishpokharel@gmail.com)
· 제1저자(First Author) : 매니쉬(Manish Pokharel) · 교신저자(Correspondent Author) : 매니쉬(Manish Pokharel)
· 접수일(2011년 2월 23일), 수정일(1차 : 2011년 3월 23일), 게재확정일(2011년 3월 28일)

Ⅰ. Introduction

The quality service in a low cost is only one way to satisfy the citizen in e-government system. Most of the services are compromised due to financial constraints. The unpredictable financial structures are always hurdles in providing quality service. E-government could not escape from global economic crisis. E-government related projects are affected a lot during the crisis. It has given a lesson to the policy makers to think seriously in this regards of sustainability in spite of economic crisis "Sustainability against every means of crisis" has become the main challenge in e-government system.

E-government takes the opportunity to provide services to the citizens in 24/7 basis. Services are available through the use of Internet. It looks very simple but it consists of complex mechanism along with the involvement of experts, software, hardware, networking devices, storing units and many more. Despite of many advantages of e-government, there is a need of huge efforts in managing resources and implementing it. Moreover, the unpredictable global financial structure has made it more difficult to the nation to set up the e-government system. Setting up the e-government system, operating it, managing it, protecting it and updating it are the big challenges for the entire world. Especially for the least developed country, it has become impossible to sustain the e-government system. In this paper, we propose the use of cloud computing as a noble approach in addressing these challenges which are relevant to both developed and least developed countries.

We begin our paper with introductory concepts in Section 1. We identify the main problems in Section 2 and provide the solution in Section 3. In Section 4, we give our experimental work for using cloud computing in e-government system. In order to show the solution of our approach, we consider the Nepal case using e-government master plan and this represents almost all developing countries in the world[3].

Nepal has very low literacy rate around 45%, as per the World Bank report. Most of the people are under poverty line. Average income is near about US$ 170 per annum. It is a big challenge for Nepal to apply the e-governance project. In order to overcome the poverty and increase the per capita income of the Nepalese people, e-governance is only one solution.

Despite of early starting of e-government project, very less significant works have been done and achieved very less results in the project. The works done in e-government projects are very less as compare to the time frame of one decade. Most of the projects are terminated at the time of implementation. The main purpose of our research is to make e-government projects success, especially in the least developed country like Nepal using cloud computing. We generalize the problems in the context of developing countries and identify the main root cause of these problems in next section.

Ⅱ. Existing and Proposed Approach

Here, we generalize the problems in sustainability of e-government system. For this purpose, we begin right from the starting point of e-government development approach. Most of the countries till today follow traditional approach of building e-government system. They start buying the required resources once t

hey understand the requirements of the citizens.

표 1. 기존 시스템과 클라우드 기반 시스템의 비교
Table 1. Comparison Existing vs. Cloud Based System

Items	Existing System	Cloud Based System
Software / Hardware	Required capital expenses	Not Required
Purchasing Software / Hardware	Required	Very less is required
Maintenance Unit	Required	Done by cloud service Provider
Scalabibity	No	Yes
Human Resource	Maximim	Vert feww

Table 1 depicts the comparison between existing system and cloud based system with the main four items. We can visualize the better position of cloud based as compare to existing system.

We describe existing and recommended approach or process to build e-government system and try to compare them with the help of flow chart in Figure 1. Our recommended modern approach is cloud based approach that we explain as a solution in next Section.

The Figure 1 is the flow chart of developing and deploying e-government system. The flow chart tries to answer of WHAT and HOW questions. We follow every required steps needed for e-government. First of all, goal has to be defined. A general goal of e-government system is "To provide the efficient service to citizen". Once the goal is known then we try to get the answer of WHAT DO WE NEED FOR THAT? We get the answer from second block of the flow chart i.

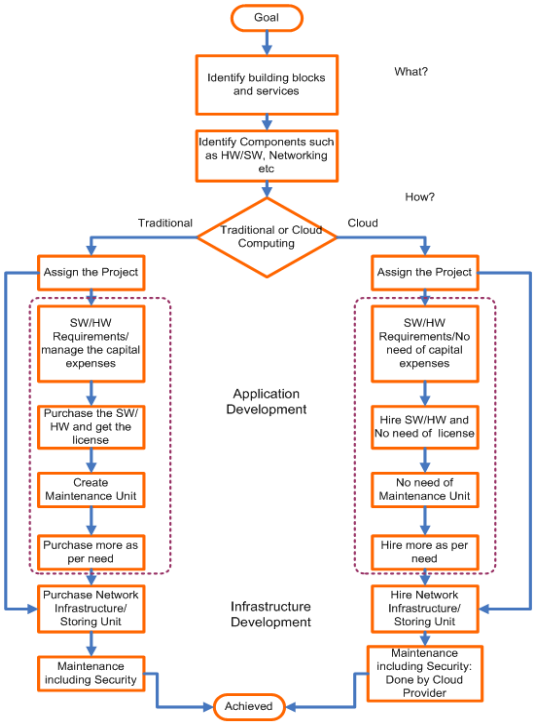

그림 1. 전자정부의 목표의 흐름도
Fig. 1 Flow Chart of E-Government's Goal

e. there is a need to identify the building blocks and the associated services. Like in other nation, Nepal also has identified G2C, G2B and G2G are main building block of e-government. The government identifies the core areas and services needs to its stakeholder. The stakeholder could be citizen, business community and government itself. Each building block is responsible for providing services. We identify the building blocks and its services in following Table 2[3].

After identifying the building blocks and its associated services then we have to get the answer of HOW DO WE GET IT? We need some components to get the services listed in Table 2.

표 2. 빌딩 블록과
Table 2. Building blocks and associated

G2C	G2B	G2G
NID Government Representation Portal, Passport Registration System, Social Insurance Information System, e-health, e-election, e-vehicle, e-driving license, e-petition, e-pension, e-post and e-agriculture	Recruitment and Employment Information System, e-customs, e-procurement, Business Registration and Approval Management System, e-patent, e-tourism, e-commerce	e-tax, Immigration Management System, e-education, e-land, e-MIS, Groupware, e-pollution, e-authentication, KMS, GIS

For this, we identify the required components such as hardware, software, networking etc as given in third block of flow chart. We understand the need of application software to provide the listed services in Table 2. We still need one more answer of HOW. This second answer can be obtained from the decision block of the flow chart. Here, we have two options either we go in traditional approach or to follow the proposed cloud computing approach. Let's discuss both approaches separately and compare it at the end.

2.1 Traditional Approach

This sub-section explains the approach of making the application software, setting up the infrastructures and environment for the e-government system. We describe each step in this category. We do not describe the first two steps since it is only requires to assign the project to the ICT industry through legal bidding system and to analyses the nature of application and find out the software and hardware requirements. We directly move to the other steps as follows:

2.1.1 Purchase SW/HW and get license

After getting the list of software and hardware from the previous steps, we need to purchase them. New software with license, a new set of servers, hardware components are to be purchased. Special experts are in need to use software and manage the hardware to make application software. It requires huge amount of budget. Initially an ICT company needs significant amount of capital expenses.

2.1.2 Create maintenance unit

Once we have set of software and hardware in our premises then a special mechanism is required to monitor them and do maintain as time goes by. A specialized maintenance unit with maintenance experts is needed for this purpose.

2.1.3 Purchase more as needed

It is very difficult to predict the exact number of software and hardware components in the beginning of the project. Especially, predicting the number of servers, amount of storing unit in advance is very difficult task. Sometimes our estimation becomes over-estimated sometimes under-estimated. If it is over estimated then there is a loss of resources and if it is under estimated then we buy the hardware and software components. This makes us to put more money which we cannot claim from the client. So, either way, we are the loser.

2.1.4 Purchase Network Infrastructure/Storing Unit

The steps inside the dotted rectangular are used for developing the application software. And these steps are to be carried out by the application developer side that is by the local ICT companies. In e-government, software should run in distributed environment i

n order to provide the services at any place at any ti me. For this, government should purchase the netwo rk infrastructures, storing unit and many more suppo rting infrastructures.

2.1.5 Maintenance and Security unit

Simply, buying the required infrastructures are no t enough, it has to be monitored regularly. As it is kn own that e-government life cycle is in cyclic in nature and has long life time, it has to be updated regularly with many new components. It is also needed to be protected from intruders. So, maintaining, and protec ting the entire e-government system does not only ne ed money but also draws lot of attentions of govern ment.

Then, finally after following these steps, a goal is a chieved. After analyzing the traditional approach we came to the conclusion that most of the e-governmen t projects do not get success because of the financial c onstrains and lack of emphasizing the core issues of e -government. Many experts are now more involved i n managing resources rather than emphasizing on th e quality of service. Here, we try to make the enviro nment in which the experts are get rid of managerial burden and focus more on providing quality service to the citizen. We provide the solution to overcome t hese problems in Section 4.

III . Solution

We identify the core issues in deployment of e-gov ernment is financial involvements in developing and operating e-government system and also continuity i n monitoring the system. Allocating budget and ma naging it are the big challenges for the nation. Despit e having many expertise and thought many e-govern ment projects are forced to be diverted because of fin ancial constrains and lack of monitoring in the core s ervice parts of the e-government. The existing approa ch mentioned in Section 3 is not appropriate for addr essing these issues. So, we propose a solution using c loud computing in e-government system. A cloud co mputing approach is not only during the developme nt phase of e-government systems but also it is used t hroughout the life cycle of the system. As there are t wo parties in cloud computing such as cloud service user and cloud service provider, we consider the gov ernment data center as a cloud service provider. We explain the details of using cloud computing in follo wing sections.

3.1 Cloud Computing

A cloud computing is an environment in which th e services are available on demand and built up with broad network access with large pool of flexible res ources and charge to its users as per the use like othe r utility services. The data center is the backbone of t he e-government system. It takes huge amount of m oney and efforts in operating and maintenance in exi sting data center. At the same time, the requirement of experts all the time makes very impossible for the l east developed country. we follow the steps such as designing layout of cloud computing in this section a nd the implementation of this layout is done in Sectio n 4. We propose the layout of cloud computing in Fig ure 2. Here, we refer e-government master plan of N epal.

The Figure 2 shows the data center in the cloud co mputing environment. There are four main compon ents in the above Figure 2 such as application softwar e (Server), middleware, computing resources and sto

ring unit. Citizen is main users of this system. A citizen could come in different role. It may be lay man who needs simple service; it may be an application developer who needs computing resources as a service.

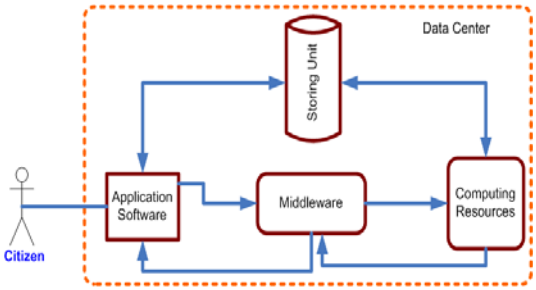

그림 2. 클라우드 내에서의 데이터 센터
Fig. 2 Data Center in Cloud

We explain on these components little after but before that we try to map the services in the e-government building blocks such as G2G, G2C and G2B to the cloud computing building block such as SaaS, Paas, and Iaas. The SaaS stands for Software as a Service in which required software is provided as a service. The PaaS stands for Platform as a Service in which the required environments for developing applications are provided as a service. The IaaS stands for Infrastructure as a Service, which provides the required hardware, software, networking, data storing unit and almost every hardware stuffs as a service. The datacenter is supposed to provide these features as an IaaS but here we expect datacenter should not only confined in IaaS but it should provide all the features or every features as a service as XaaS (Every features as a service). We try to map the cloud services with e-government features in Table 3. The mapping Table 3 shows clearly how the features of e-government system can be obtained through cloud services.

표 3. 전자 정부 및 클라우드 컴퓨팅 사이의 매핑
Table 3. Mapping between E-Government and Cloud Computing

Cloud Services	E-Government Features
SaaS	G2C, G2B, G2G -The application belongs to these categories are provided to the citizen through SaaS.
PaaS	Managing Citizen Request such as Queuing Service, Working as a middleware between Citizen and Government such as broker, Storing unit such as Database services, and also if an organization in government wants to develop the new application then PaaS provides the platform such as OS, programming language etc.
IaaS	Expecting existing data center is to provide all the hardware, storage and network features to the application developers. Data center provides the computing power, memory, storage.

The application software (server) behaves like SaaS whereas computing resources and storing unit behave like PaaS and IaaS. Now, we explain the Figure 2 in details in following sections.

3.1.1 Application Software (Web Servers)

This is entry point for the citizen. A citizen with different needs interact with the data center through this point. It is a gateway for the citizens or users to access the services. There are many servers in this component running application software and providing the environment for the authorized users to enter the government's data center. These servers are called web server instances. Instance is a virtual server that runs our guest OS and it is based on the machine image from which the instance was cloned. A machine image can be a piece of software, libraries, data, information etc which are kept in the elastic block component. We can increase the number of instances during the high demand period and reduce it during the low demand period. Cloud computing has the facilities to create and terminate the instances.

3.1.2 Middleware

It is a bridge between web service instance and co mputing resources. It has a tiny database in which th e details of available resources are kept. It has the ma in task for providing request to the data center and se rvices back to the citizens through web server instanc e. There are two queues for this purpose and it is bas ed upon the message queue system.[4] A request fro m a citizen is kept into the request queue as per first i n first out (FIFO) and also in the same way the servic es from data center are kept in the response queue. E ach request is handled with the instance in the comp uting resources. Even a working instance gets collaps ed; the request will be remaining in the queue so that new created instance can handle it without loss and down. It has overcome the problem of losing data an d information during process migration to new mach ine in tradition approach.

3.1.3 Computing Resources

All computing resources especially, hardware, res ources are kept in this component. We keep the reso urces as per the need in e-government system. While developing application software in e-government, th ere may be need of computing capability, non-persist ent storing capability, communication capability etc. All these resources are kept in this component. These resources can be expanded and reduced as per the us ers' need. For this, we use virtualization as one of the core technologies.

3.1.4 Virtualization

It is a technology that combines or divides comput ing resources to present one or many operating envir onments using methodologies like hardware and soft ware partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and ot hers. It is already known as software technology that is fundamentally changing the way of computation. I n hardware emulation, the virtualization software is usually referred to as a hypervisor. We can understa nd more by the following figure and we explain how it works in e-government system.

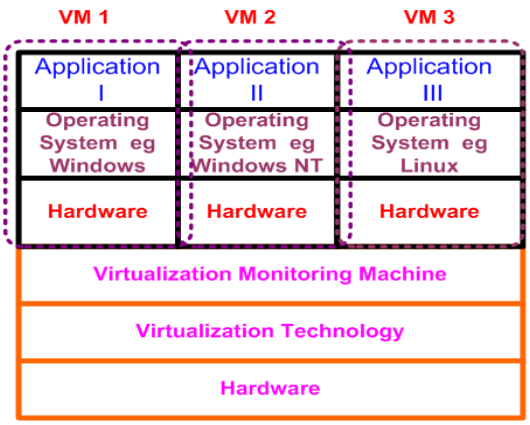

그림 3. 가상화
Fig. 3 Virtualization

We can create any number of VMs as per our nee d. In e-government, application developer can use th e VM to develop their application or use for other pu rpose. If they need more they get it with a simple req uest and if they do not need it can be discarded. Any particular hardware also can be expanded or reduce d. Thus, these features of computing resources make the possibility of high scalability in the resources.

3.1.5 Storing Unit

It is a persistent place for the data, information, sof tware and machine image. These are kept inside the s torage unit in machine image (MI) format. A one ma chine image that store in this component can be insta ntiated into many instances. Example: If "Software A" is kept inside it as machine image and if there is a

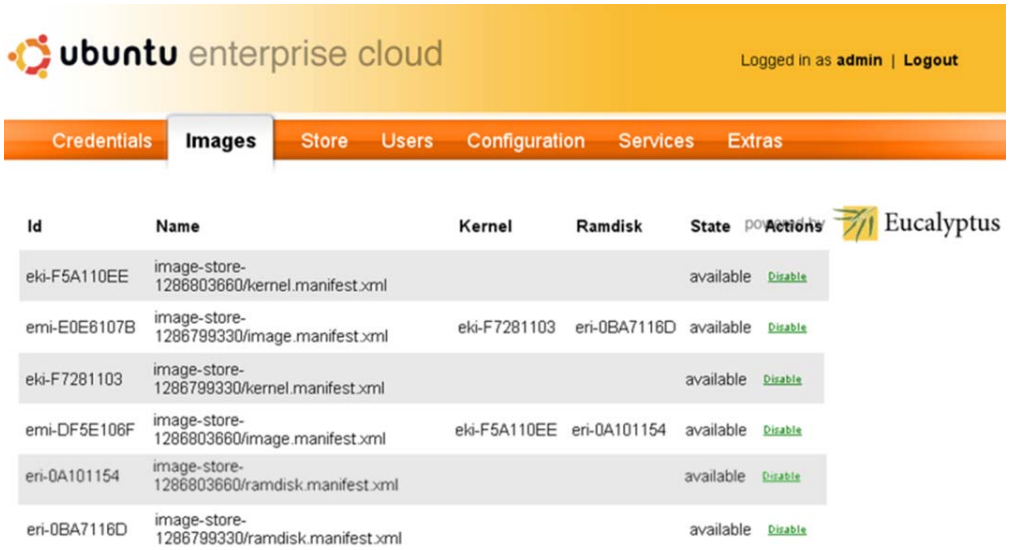

그림 4. 유칼립투스 웹 콘솔을 통해 EMI를 설치

Fig. 4. Install EMI via Eucalyptus Web Console

need of multiple servers with "Software A" then it can be cloned as instances of "machine image of Software A". In order to understand more clearly we treat machine image as a class and instance as an object as we do in object oriented technology. Like, this we can create many instances.

This component is considered as a heart of our architecture because both the components like Web Server Instance and Computing resources, data storage after computing are depended upon the existing of elastic block storage. If this component goes down then new instances cannot be created and storage will not be available that can be attached for instance.

IV . Experiment

In order to know the use of e-government in cloud computing environment, we conduct the experiment in our lab which is based upon the eucalyptus cloud architecture. First we create the cloud environment then we install Eucalyptus Machine Image (EMI). After the image is installed and registered, we take the instance of it as normal via the Hybridfox. The Eucalyptus cloud does not only provides the features for creating, managing and accessing the cloud but also provides an EC2 and S3 as a compatible cloud computing platform as well. In general, we use four steps for creating and verifying the EMI: Create Machine Image, Register the Machine Image, Running Terminating Instance.

There are two approaches for creating machine image but in this paper we present only the one by installing from the Store tab of Eucalyptus Web Console. After the image is verified, we create Elastic Block Storage (EBS) and attach to the instance. The EBS is the permanent storage that we use to install our e-government applications. In this paper, we present Moodle (e-learning application) that will be embedded on the

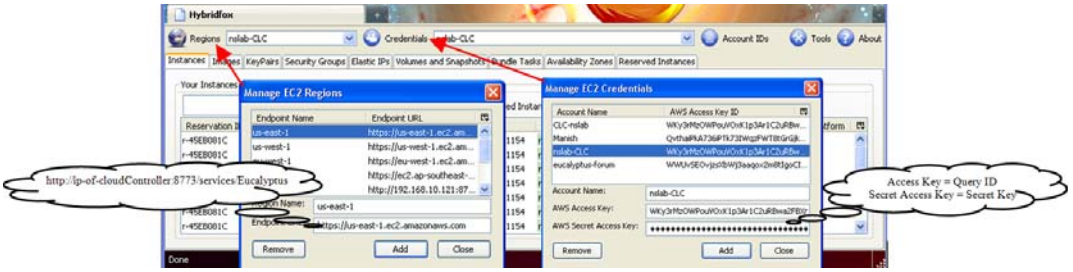

그림 5. 하이브리드폭스 유칼립투스 1.6.2 구성

Fig. 5. Configure Hybridfox Eucalyptus 1.6.2

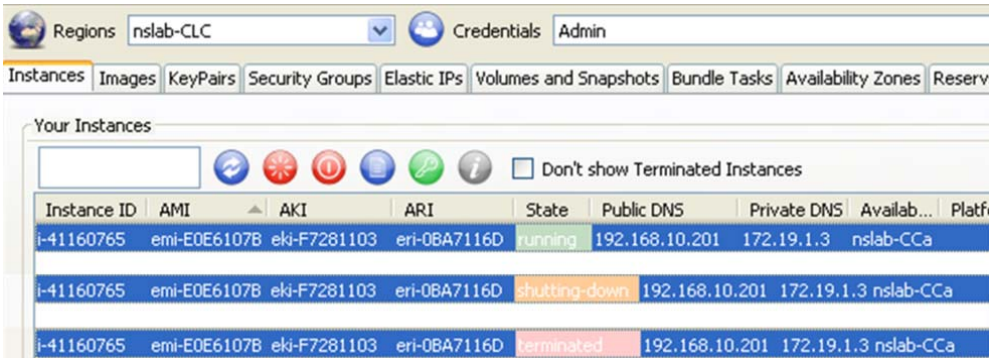

그림 6. 유칼립투스 클라우드상에서의 동작과 종료의 예

Fig. 6. Running and Terminating the Instance on Eucalyptus Cloud

EBS. We name it Moodle_EBS. The Moodle_EBS can easily attach to another instance when one instance g ets crashed. The following steps guide through the e ntire scenario.

4.1 Install and Register EMI

After implementing the UEC, we login to web con sole management. On the Store tab, we installed Ubu ntu 10.04 LTS - Lucid Lynx (amd64). The system wil l be downloaded, installed and registered itself auto matically to the cloud environment. The Figure 4 sho ws a successful installed EMI.

4.2 Running and Terminating EMI

In our verification of the EMI, we use Hybridfox a s it is a cloud management tool. It is a seamless tool f or managing Cloud Computing environment. We ne ed to configure a few parameters such as defining a r egion and adding credential before start using it. To s implify the processes, we create a visual configuratio n picture as shown in Figure 5.

When the configuration is done, a list of available i mages is listed in the Image tab. We launch the instan ce and terminate it by showing in Figure 6.

4.3 Embed package Moodle on EBS

Our instance works fine. First, on the Volume tab of Hybridfox, we create a volume and tag it as moodl e_ebs. Second, we configure services for EBS.

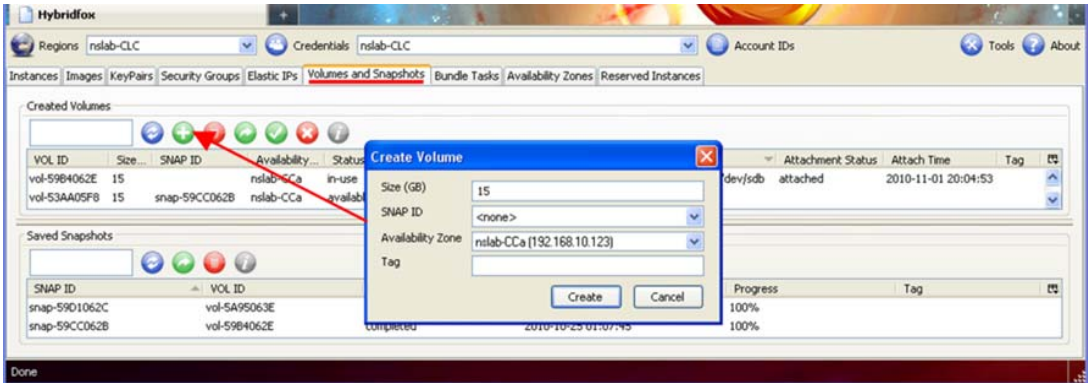

그림 7. 하이브리드폭스와 함께 볼륨의 생성

Fig. 7. Create Volume with Hybridfox

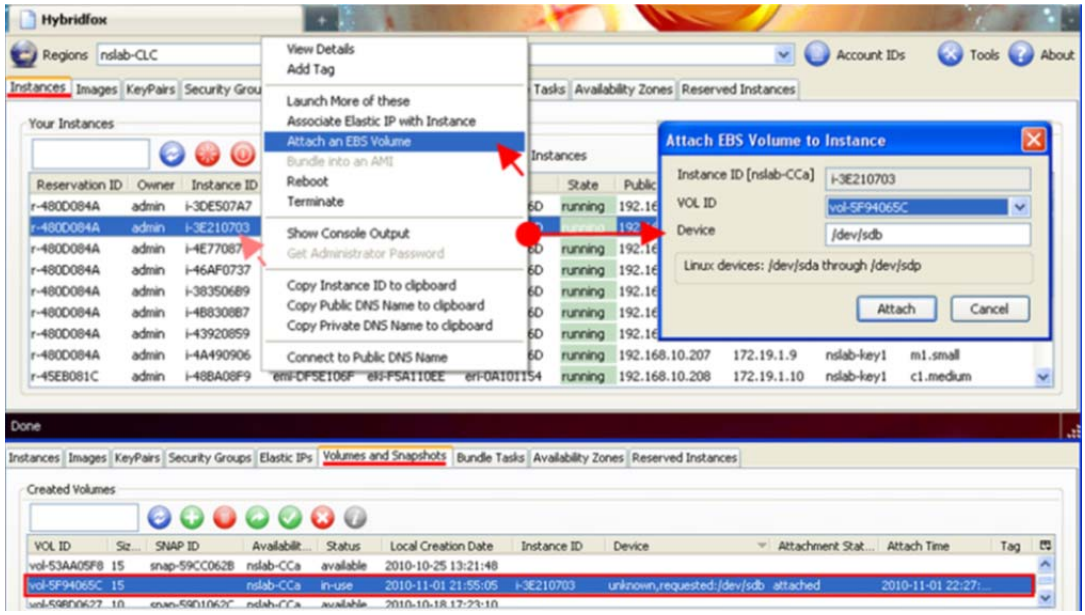

그림 8. 인스턴스에 EBS볼륨을 추가

Fig. 8. Attach EBS Volume to an Instance

4.3.1 Creating Elastic Block Storage (EBS) Volume

Figure 7 shows the processes of creating a volume and after these processes the cloud system named the volume for example vol-5F94065C.

Then we attach this volume to an instance ID, th at is i-3E210703, as shown in Figure 8.

4.3.2 Configure Services for EBS Volume

This section explains the procedure of installing e-government application server, Moodle, on an EBS v olume, keeping in mind of no loose of data in term of instance to crash, unplanned downtime, or maintena nce downtime.

If the instance, where the volume attached is crashed; we just reattach the volume to another instance with a very less configuration.

Therefore, to allow services be able to run from an EBS volume, we install a basic system into the EBS volume. Let's go through the steps as the following.

- Start an instance, for example i-3E210703 in our case.
- Create a volume with the appropriate size, so we create 15GB; Our volume named vol-5F28064E
- Attach vol-5F94065C to i-3E210703
- chuobsok@nslab-CCa:~$euca-attach-volume -i i-3E210703 vol-5F94065C -d /dev/sdc
- Now, log into the instance as root and create a single partition for the volume, sdc.

  root@172:~# mke2fs /dev/sdc
- Create a mount point for the persistence volume; in our case is "nslab_ebs_service" and then mount it there.

  root@172:~# mkdir /nslab_ebs_service

  root@172:~# debootstrap squeeze /nslab_ebs_service
- Since we use Ubuntu, we use debootstrap because it helps to create a Debian base system from scratch without requiring the availability of dpkg or apt. It downloads ".deb" files from a mirror site and carefully unpacks them into a directory which can eventually be chrooted into later i.e. nslab_ebs_service.

  root@172:~# aptitude install debootstrap
- Then we install a basic system in the EBS volume.

  root@172:~# debootstrap squeeze /nslab_ebs_service
- moodle_ebs volume needs support from other system folders of the instance machine such as "proc, sysfs, and dev", so we mount those folders to the volume.

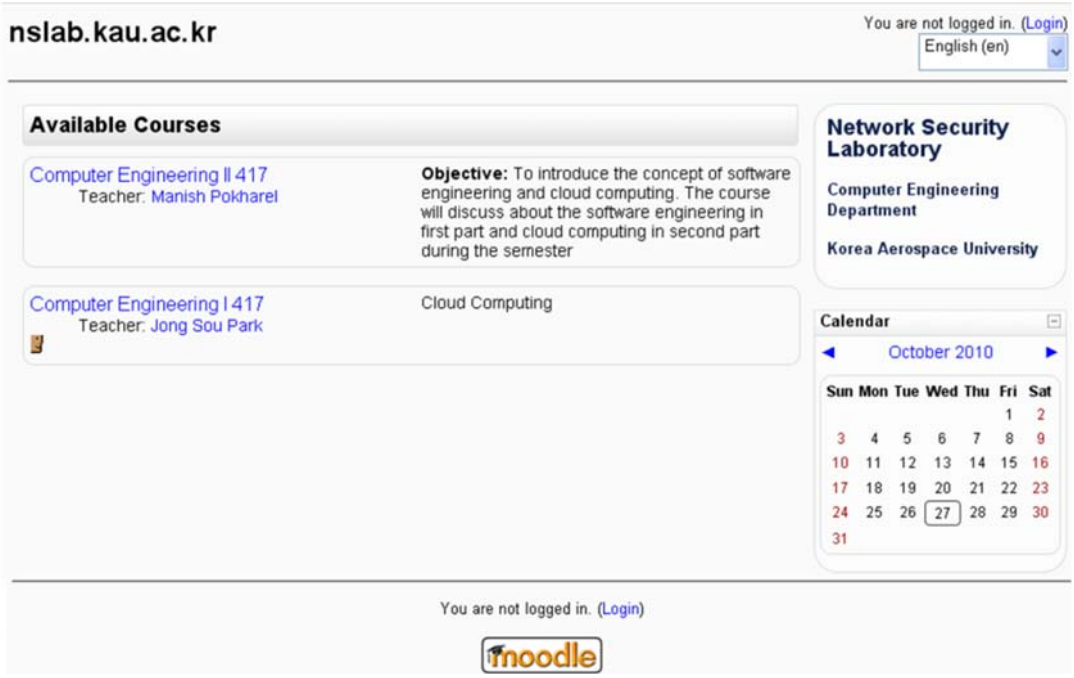

그림 9. 유칼립투스 머신 이미지의 에로서의 무들 이러닝 동작?

Fig. 9. Moodle E-Learning Running on Instance of Eucalyptus Machine Image

- Now, the moodle_ebs volume will act like a new machine image, so we use command "chroot" into it.

  root@172:~# chroot /nslab_ebs_service
- As the nature of Moodle package it requires Apache, MySQL, and PHP in order to run it. So, we install the needed packages for the Moodle of our e-learning application server.
- Finally, we install Moodle package.
- Because Eucalyptus cloud secure its instances within security group then in order to allow our clients or users access our web server instance from outside the network, we allow port 80 for them. Then we find Figure 9.

V . Conclusion and Future Works

Based upon our experiment in our lab, we identify many advantages in cloud computing , especially using SaaS for e-government system. We have listed few of them below:

Economic Benefits

A nation has to invest huge amount of money to deploy the e-government system smoothly for a long time. Physical infrastructures, hardware, software, networking and maintenance on it consume a lot of money of the nation.

A nation goes through the tough time during economic crisis. We find the cloud computing to rescue the nation where it provides all the features as a service, and there is no need of physical infrastructure, no need of maintenance and extra manpower round the clock. Cloud computing does not need initial capital expenses, it only needs operating expenses and charge is made as per the use. So, these features of cloud computing make the nation economic condition better and provide sustainability to the e-government system.

Promoting local ICT industry

Many ICT industries go bankrupt because of financial constrains and less income. Despite the qualified manpower available in the nation, they cannot use their expertise because of the need of big financial supports to set up their industry. In our approach as mentioned above, if the available data center works as a cloud service provider then local ICT industries would get all the resources as a service from data center and can develop the application software using the resources.

Availability

It is strength of a system to make the services available throughout the time. In cloud computing, since all the resources are kept as machine image in storage unit, and even if a machine crash then it can be recovered very quickly and easily by insanitation the machine image from storage unit.

We proposed the software as a service (Saas) as a noble approach to use in e-government system. We also made a point that the datacenter can be acted as a cloud service provider so that including the nation, every local ICT entrepreneur are benefited. We have demonstrated the possibility of providing quality service in minimum cost through Software-as-a-Service. Cloud computing leads the government to overcome the financial constrains during development, deployment and management of e-government system. It also increases the quality of service since it takes out the responsibility of managing the resources during the life cycle of e-government system.

The security aspects are the main limitations of ou

r proposed approach. Governments are very reluctan t to store their sensitive data in the cloud and most of the time governments are not aware about the existe nce of their data. We decide to consider these limitati ons as out future research works. We will work on se curity aspects of cloud computing in E-government s ystem.

References

[1] R. Buyya, C.H. Y, S. Venugopal, J.Broberg , andI. Brandic, " Future Generation Computer System", *Elesevier, 2009.*

[2] http://download.moodle.org

[3] KIPA, "E-government Master Plan Consulting Report for Ne pal", 2006

[4] http://open.eucalyptus.com/wiki

[5] L.Youseff ,B.Maria , "Toward a Unified Ontology of Cloud Computing", *IEEE*

[6] F.M.Aymerich, G.Fenu and S., Surcis , "An approach of clo ud computing network", *IEEE 2008*

[7] "*Ubuntu Enterprise Cloud Architecture"*, Technical White Pa per, http://www.cononical.com 2009

[8] M. Armburst et al, "*Above the Clouds : A Berkeley View of Cl oud Computing*", Technical Report No. UCB/EECS-2009-2 8, 2009

[9] E. Estevez and T. Janowski,"*Building a Dependable Messagi ng Infrastructure for Electronic Government*". UNU/IIST R eport No 368. April 2007

저자소개

Manish Pokharel

In 1995, Bachelor of Engineering in Computer Science, Karnataka University

In 2000, Master of Engineering in Software System, Birla Institue of Technology and Science

2007 ~ present Ph.D. student in Korea Aerospace University

※ Interest : E-government, Enterprise Architecture, Fault Tolerance and Cloud Computing

Sok Chuob

In 2002, BS in Computer Science Royal University of Phnom Penh

2011 MS in Computer Engineering Korea Aerospace University

※ Interest : E-government, Enterprise Architecture, Fault Tolerance and Cloud Computing

박종서(Jong Sou Park)

In 1986, Master of Electrical and Computer Engineering, North Carolina State University

In 1994, Ph.D of Computer Engineering, Pennsylvania State University

1996 ~ present Professor in Korea Aerospace University

※ Interset : information security, embedded system and hardware design.