CONTROLLING A CLOUD WITH A PHONE

Team Members:

Apoorve Tomer
Apoorve.Tomer@iiitb.org

Bisen Vikrantsingh Mohansingh
BisenVikrantsingh.Mohansingh@iiitb.org

Poulami Debnath
Poulami.Debnath@iiitb.org

Rinitharaj K
Rinitha.Raj@iiitb.org

Shalini Tawari
Shalini.Tawari@iiitb.org

International Institute of Information Technology, Bangalore

Technical Report IIITB-TR-2013-005e

April 2013

Abstract

In this project we have developed a mobile application which would help the cloud administrator to perform routine administrative tasks very easily directly from the mobile. For the cloud setup, we have used Eucalyptus based cloud installation, and for the mobile application, we have used HTML5, mobile jQuery and CSS3 which make the mobile application platform independent. Any other cloud can be accessed through this mobile application if it is based on AWS or Eucalyptus standards. Some features of our application include instant cloud updates, quick links for some frequently used entites like snapshots, instances and volumes; also it has a tab view just like HybridFox so as to keep the administrator familiar with the application. We have implemented twenty-five use cases which are the most frequently used cloud administrative routine tasks.

URL: `https://github.com/osproject5e/control-cloud-using-phone`

Acknowledgement

We are extremely grateful to those who have helped and supported us during the project. Our deepest thanks to our advisor Prof. Shrisha Rao for his continuous encouragement and suggestions throughout the course of this work. It was our pleasure to work under his guidance. Special thanks to Mr.Deependra Shekhawat who is a Technical Support and Community Engineer at Eucalyptus Systems Inc. for helping us in various cloud operational issues.

Contents

List of Figures

1 Introduction

Cloud Computing is defined by the US National Institute of Standards and Technology (NIST) [1] as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Today the traditional access to cloud is moved from console based to web based interface. After looking at the rapid growth of cloud technology users and also the simultaneous increase in capability of mobile devices in terms of processing and storage, it is predicted that in future, cloud administrators will demand more flexible and portable mobile based cloud controller which can provide them ease in accessing and monitoring instances and resources. The idea is to extend the controlling capability of cloud from laptop to mobiles/tablets. This would enable the administrator to be more mobile. He need not carry his laptop everywhere to control the cloud, because he can do it with his mobile/tablet itself. Thus, the idea aims at using cloud computing techniques for controlling, storage and processing of data on mobile devices for controlling the cloud.

1.1 Project Description

Objective

The objective of this project is to create a cloud and a suitable phone interface that can be used to control it for all routine administrative purposes.

Description

This project would enable cloud administrators to have options for managing their systems from any phone, which supports web browsing. It can be used to manage EC2 instances, Simple Storage Service (S3) buckets, network related tasks like associating Elastic IP with instances. The administrator would be able to keep tabs on his EC2 resources (instances, volumes, etc.) using this application running on his mobile/tablet.

The cloud service model Infrastructure as a Service (IaaS) [2]provides users the ability to run their own softwares which include Virtual Machines (VM), operating systems and applications. IAAS clouds help the users to rent the resources needed via the internet. IAAS has the ability to set up a Public or a Private cloud. A private cloud refers to a specific cloud architecture [3]that provides required resources to a certain fixed number of clients on some payment for these services. The private cloud units serve as a computational as well as storage infrastructure for mobile devices. IAAS providers like Amazon web services help in setting up private cloud.

EUCALYPTUS is an open-source software framework [4]for cloud computing that implements what is commonly referred to as Infrastructure as a Service (IaaS); systems that give users the ability to run and control entire virtual machine instances deployed across a variety of physical resources. An infrastructure setup using the cloud computing model is generally referred to as the cloud. Eucalyptus provides EC2-Elastic Compute Cloud and S3-Simple Storage Service. EC2 provides resizable computing capacity. S3 provides compatible cloud storage platform.

1.2 Gap Analysis

Some of the existing systems are as follows:

- HybridFox:
  HybridFox is a plugin supported in Firefox. It is used to perform different cloud controlling tasks such as launching, terminating, starting, stopping of instances; creating, associating, deleting keypairs and similar operations for snapshots as well. But Hybrid Fox is compatible only with firefox and not supported on mobile phones.

- AWS Manager:
  It is an android application for listing reports of Amazon clouds. But it is compatible only with tablets and not with mobile phones. It does not rotate from portrait to landscape orientation. Moreover, it does not allow us to create instances. From a security aspect, it has no password protection. Also, it lacks good quality graphics.

- Rackspace Cloud:
  It is not an open source, it is a commercially available product and only supports Rackspace based cloud.

Applications running only on specific phones and not being portable across various phone-platforms is a major drawback.

We have designed a web-based mobile application which will provide all essential features including, but not limited to managing instances, volume, keypairs, elastic IP, etc.

2 System Design

2.1 System Architecture

The system architecture of our project comprises of four main parts as explained below:

1. Cloud Frontend
It consists of four sub-components as described below:

- Cluster Controller (CC)

  CC communicates with Cloud Controller (CLC) [5] on one side and Network Controllers (NC) on the other side. CC manages one or more Node Controllers and deploys/manages instances on them. It also manages networking for the instances running on them.

- Walrus Storage Controller (WSC)

  WS3 provides a persistent simple storage service using REST and SOAP APIs which are compatible with S3 APIs. It is considered as a simple file storage system.

- Storage Controller (SC)

  SC provides persistent block storage used by the instances.

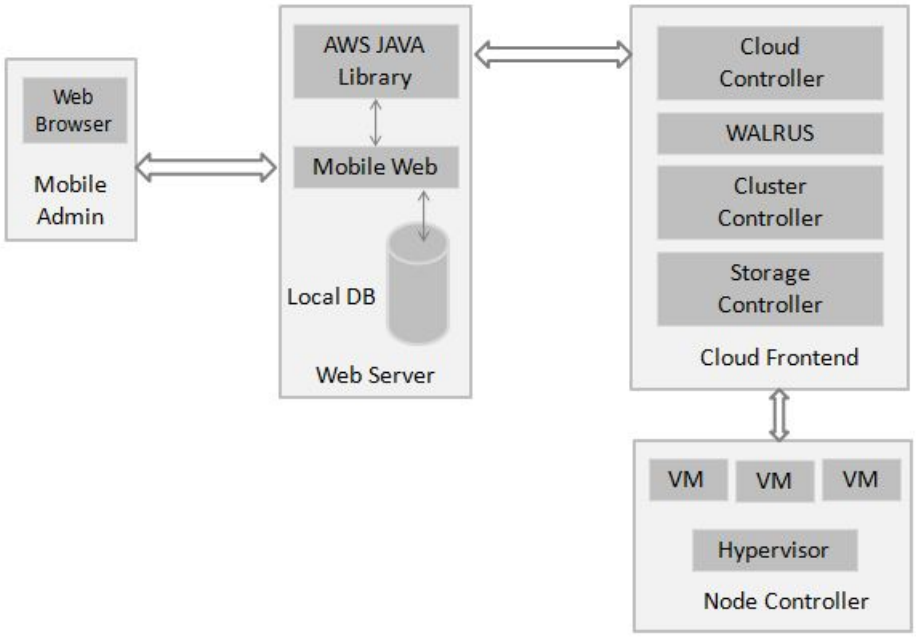

Figure 1: Architecture Diagram

- Cloud Controller (CLC)

  It monitors the running instances and has complete information about the availability and usage of all resources in the cloud. It also helps in deciding which clusters will be used for provisioning the instances.

2. Node Controller (NC)

A node is capable of running hypervisor/virtual machine manager. The virtual machines running on the hypervisor are called instances. Node Controller runs on each node and controls the life cycle of instances running on the node. It also collects data related to resource availability and utilization on the node and reports the data to CC.

3. Web Server

It consists of three main sub-components:

- AWS Library
  Eucalyptus 3.2 provides support for the old version of AWS API 1.2.12

only. So by directly using this API, we have made our application generic. Therefore, we can control clouds based on both AWS and Eucalyptus.

- Mobile Web
  It is a repository which holds all the mobile-specific web-pages. These pages have been created using JAVA, JSP, mobile jQuery and other web 2.0 technologies.

- Local DB
  Mysql database has been used. Its main purpose is to save user credentials, cloud access keys, secret keys and end-points of cloud. It also holds data on UI preferences settings (Quick links customization).

4. Mobile Admin

As we have created a mobile web-application, it is compatible across all smart phones irrespective of their manufacturer/operating systems. Just with the help of a web browser, which is present in almost every mobile, a cloud administrator can operate and control the cloud.

2.2 Class Diagram

A separate class is made for each of the following features of Eucalyptus cloud

1. EucaImages
   This class implements the functionality of listing the existing Eucalyptus machine images which are present in the cloud.

2. Instances
   This class implements the functionality of listing the existing instances, creating instances and stopping instances.

3. Keypairs
   This class implements the functionality of listing, creating and deleting keypairs.

4. Snapshots
   This class implements the functionality of listing, creating and deleting snapshots, and also creating a volume from another snapshot.

5. ElasticIP
   This class implements the functionality of listing the existing IPs and associating IPs.

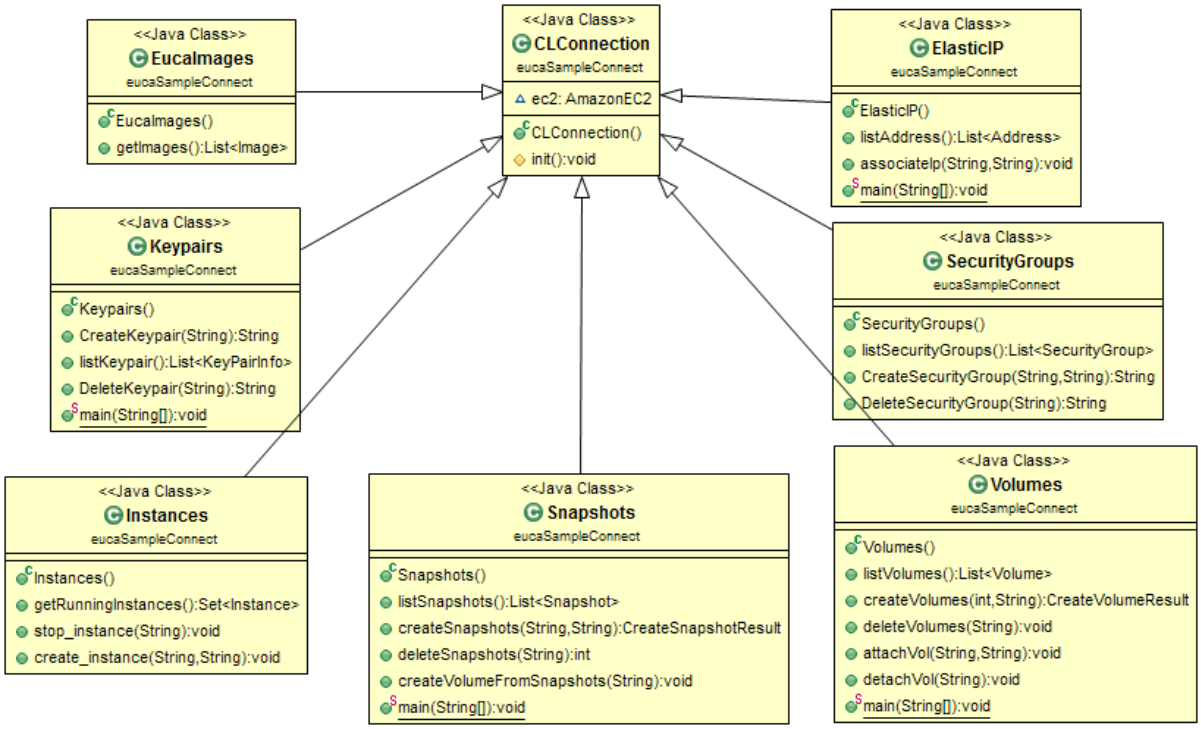

Figure 2: Class Diagram

6. SecurityGroups
   This class implements the functionality of listing, creating and deleting security groups.

7. Volumes
   This class implements the functionality of listing, creating and deleting volumes. It also has the methods for detaching and attaching volumes to instances.

All these seven classes inherit the parent class CLConnection. The class CLConnection contains the code for connecting to the Eucalyptus cloud using secret key, access key and cloud endpoint. So this class serves as an authentication class. The functionality of every feature is implemented in its class.

2.3 Use Case Diagram

Following is the list of use cases which occur most frequently. All other use cases are similar to the following use cases.

1. Launching an instance:
   - Opening the browser
   - Logging into the web application
   - If the user wants to use a new keypair, then keypair name is entered
   - Else an existing keypair is selected
   - The Eucalyptus Machine Image(EMI) is selected
   - The launch instance button is pressed
2. Attaching a volume:
   - Opening the browser
   - Logging into the web application
   - If the user wants to attach an existing volume to an instance, then the volume from drop down list is selected.
   - Else the zone and size for the new volume are selected
   - The Attach Now button is pressed
3. Creating a Snapshot:
   - Opening the browser
   - Logging into the web application
   - Selecting the volume ID from the drop down list

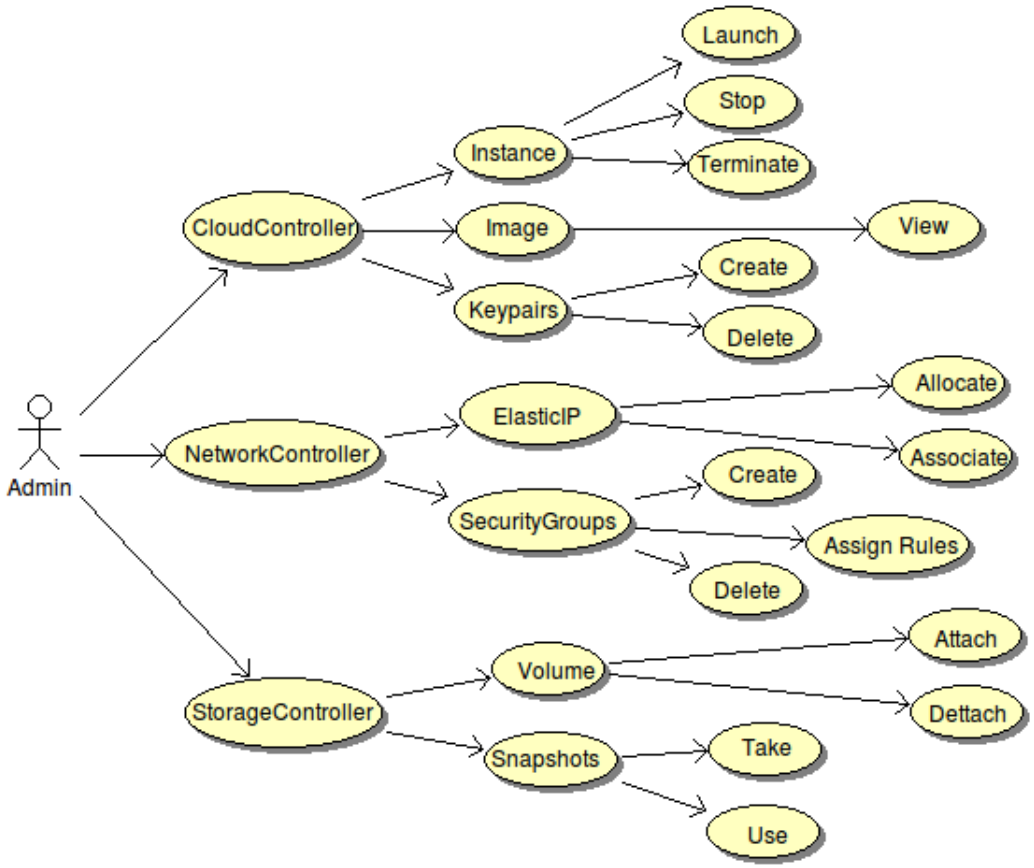

Figure 3: Use case diagram

- Entering the description of the snapshot
- The Create Now button is pressed

3 Implementation

3.1 Cloud Environment Setup

The installation of CentOS 5 was done, which provides Eucalyptus installation packages. The cloud setup was done using two machines to create the private cloud platform using Eucalyptus.

3.1.1 Installation of Node Controller

To install a standalone Node Controller, the instructions given below were followed. It is strongly recommended that any Node Controllers be installed first before the Frontend is installed.

1. The target system is to be booted from the Eucalyptus Faststart media. After the boot screen loads, "Install Node Controller" is selected.
2. The network information should be provided. Static networking is recommended; DHCP will work in many cases, but if DHCP leases change, the Eucalyptus cloud will no longer be functional.
3. The timezone is selected and the root password entered for the system.
4. The Node Controller installation now begins. When this process is completed, the user will be prompted to reboot the system. After rebooting is done, root user logs into the system.Then, the post-installation configuration will begin. The defaults for NTP configuration, networking mode, and network interface are to be accepted.
5. The installation of the Node Controller is now complete. Now if multiple Node Controllers are required, the same procedure is followed for all. After the installation of all the Node Controllers, we move on to install the Eucalyptus Frontend.

3.1.2 Installation of Frontend Controller

To install a Frontend on a separate system, the instructions to be followed are:

1. The target system is booted from the Eucalyptus Faststart media.After the boot screen loads, "Install Frontend" is selected.

2. The network information should be provided. Static networking is recommended; DHCP will work in many cases, but if DHCP leases change, the Eucalyptus cloud will no longer be functional.

3. The timezone is selected and the root password for the system is entered.

4. The range of public IP addresses should be specified. New virtual instances created by Eucalyptus will receive IP addresses from within this specified range. The lower and higher limit of the available public IP addresses is to be entered with a hyphen between them (e.g.: 192.168.1.200-192.168.1.240).

5. The Eucalyptus installation will begin. Software will be installed, and a default Eucalyptus machine image (EMI) will be built. When this process is completed, the user will be prompted to reboot the system. When the system reboots, the user will be prompted to accept the license for this installation.

6. The IP addresses of the Node Controllers that were previously configured are to be entered. A non-root login is to be created, and NTP is to be turned on. Note that NTP is required for Eucalyptus to function properly.

7. The installation is now complete. It can be checked whether the cloud is running, by clicking on the the web browser links from the Desktop.

3.1.3 Accessing the Cloud

The most common cloud controller accessing methods are listed below.

- Eucalyptus Cloud Dashboard: It is a web-based interface.
- Euca2ools : It is a Eucalyptus command line interface. Eucarc is a resource configuration file. Euca2ools can be accessed after sourcing Eucarc file.
- Remote client: Used to access Euca2ools using SSH login

3.2 Cloud Administrative Tasks

Some of the most common cloud administrative tasks with their commands are listed below. All these tasks are implemented in our mobile web-application.

1. **Finding an Image**
   Shows information about an image
   euca-describe-images
2. **Creating Key Pairs**
   Creating a key pair generates two keys: a public key and private key. When attempting to login to a VM instance using SSH, the public key is checked against the private key to verify access.

   Euca-create-keypair <keypair-name>
3. **Deleting a Key pair**
   Deletes an existing keypair.
   Euca-delete-keypair <keypair-name>
4. **Launching an Instance**
   - Adding a Keypair
     The private key is saved to to a file in the local directory
     euca-add-keypair <keypair-name> > <keypair-name.private>
   - To change the file permissions to enable access to the private key file in the local directory:

     chmod 0600 keypair-name.private
   - Querying the system to view the public keys:

     euca-describe-keypairs

- Authorizing Security Groups
  Before logging into an instance, the access to that instance must be authorized. This done by configuring a security group for that instance. When an instance is first created, it is assigned to a default security group. To allow login and usage of a new instance, network access must be authorized to the default security group with the euca-authorize command. The command to grant unlimited network access using SSH is:
  euca-authorize -P tcp -p 22 -s 0.0.0.0/0 default
- Running an Instance:
  To launch an instance, the euca-run-instances command is used, and an image ID and the name of a keypair are provided.
  euca-run-instances <image-id> -k <mykey>
- Status of instances:
  To get the launch status of the instance:
  euca-describe-instances <instance-id>

5. **Terminate Instances**
   The euca-terminate-instances command lets us cancel running VM instances.
   euca-terminate-instances <instance-id>

6. **Stopping an Instance**
   Stops a running instance
   euca-stop-instances <instance-id>

7. **Attach Volume:**
   Eucalyptus offers persistent storage that can be attached to a running instance. These Eucalyptus block storage (EBS) volumes persist autonomously from the running life of an instance. After attaching a block volume to an instance, it can be used like any other physical hard drive.

   - Start running instances
     euca-run-instances <image-id>
   - Creating a new EBS volume in the same availability zone as the running instance. The command displays the ID of the newly-created volume.

euca-create-volume –zone <partition-name>–size <volume-size>

where zone is the availability zone to create the volume and size is the size of the volume in GB.

- Attach the newly-created volume to the instance.

  euca-attach-volume <volume-id> -i <instance-id> -d <device-name>

8. **Detaching a Volume**
   To detach a block volume from an instance
   euca-detach-volume <volume-id>

9. **Creating a Snapshot**
   Creates a snapshot from an existing volume.

   euca-create-snapshot <volume-id>

10. **Associate IP address**
    Associates an elastic IP address with an instance. This assumes that the address supplied has already been allocated with the euca-allocate-address command.
    euca-associate-address -i <instance-ID><IP-address>

11. **Creating a Security group**
    Security groups control network access to instances by applying network rules to instances associated with a group.
    euca-add-group -d <description ><group-name>

12. **Deleting a Security group**
    The euca-delete-group command deletes security groups.
    euca-delete-group <group-name>

13. **Listing Snapshots**
    Shows information about snapshots.
    euca-describe-snapshots <snapshot-id>

14. **Listing Volumes**
    Shows information about volumes.
    euca-describe-volumes <volume-id>

15. **List Keypairs**
    Shows information about key pairs.
    euca-describe-keypairs <keypair-name>

16. **List the status of instance:**
    To get the launch status of the Instance:
    euca-describe-instances <instance-id>

3.3 Mobile Web Application

3.3.1 Tools and Technologies

1. Java 7 SDK
2. AWS Java SDK version 1.2.12
3. Eclipse(Juno)
4. JSP
5. Mobile JQuery
6. HTML5
7. CSS3

3.3.2 System Features

Some of the key features implemented are discussed below with their sequence diagrams.

1. Launching an Instance
   A new instance can be lauched using a keypair and a eucalyptus machine image. We have provided the user with two choices:
   (a) Use an existing keypair.
   (b) Create a new keypair at runtime and use the same.

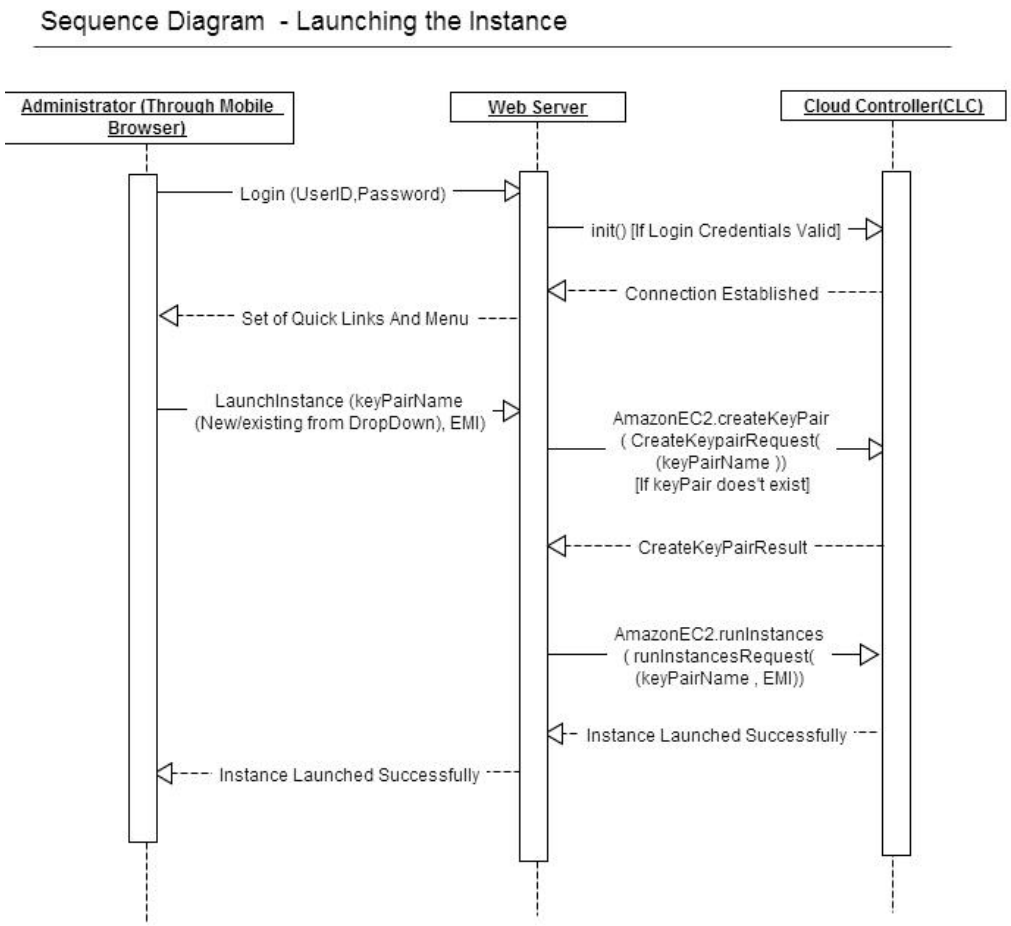

Figure 4: Sequence Diagram for Launching Instance

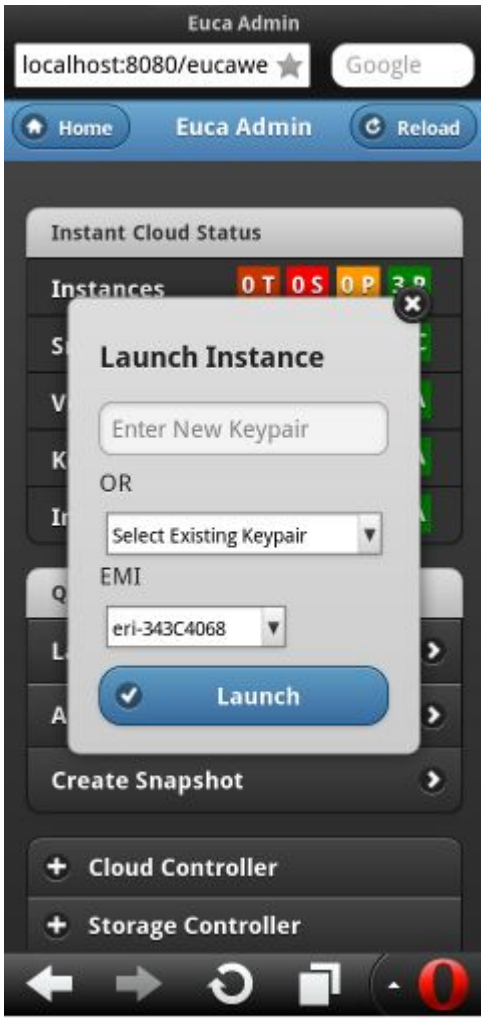

Figure 5: GUI for Launching Instance

2. Attaching a volume to an instance
   In order to make the storage of an instance persistent, we can attach a volume to it. Two paramenters are required for this operation- Volume ID and Instance ID. Here the user has two choices:
   (a) A new volume can be created and attached to it
   (b) An existing volume can be selected from the drop down list.

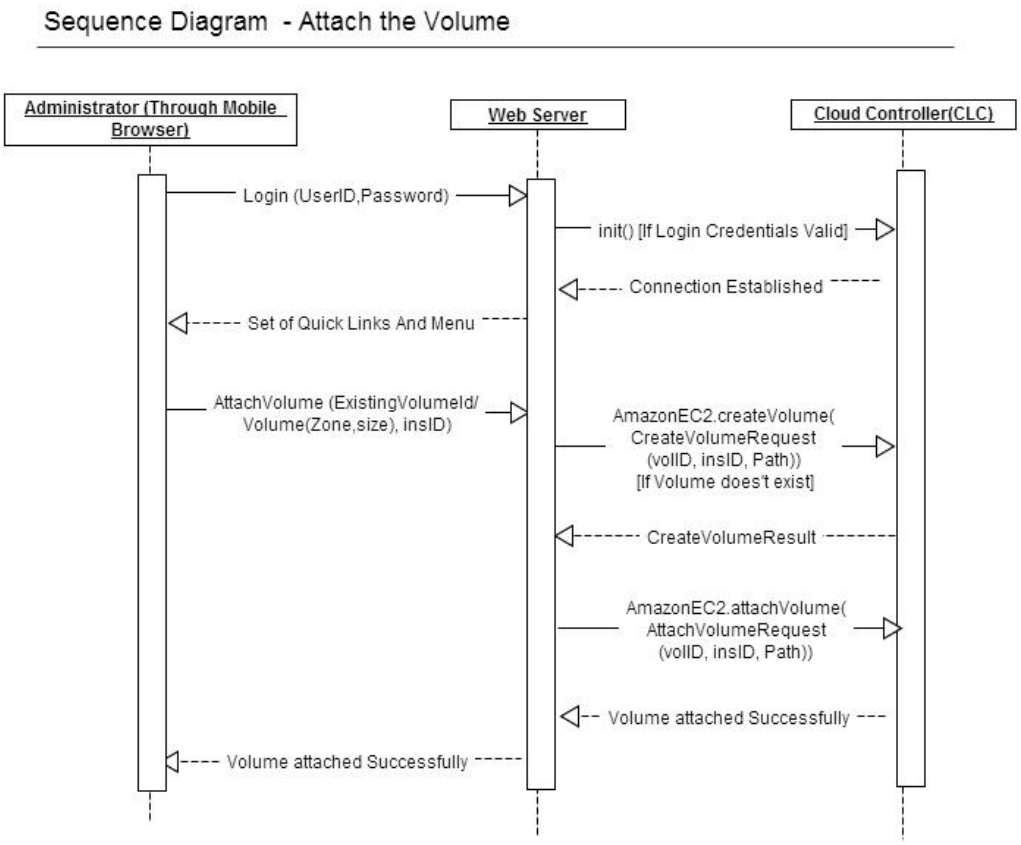

Figure 6: Sequence Diagram for Attaching Volume

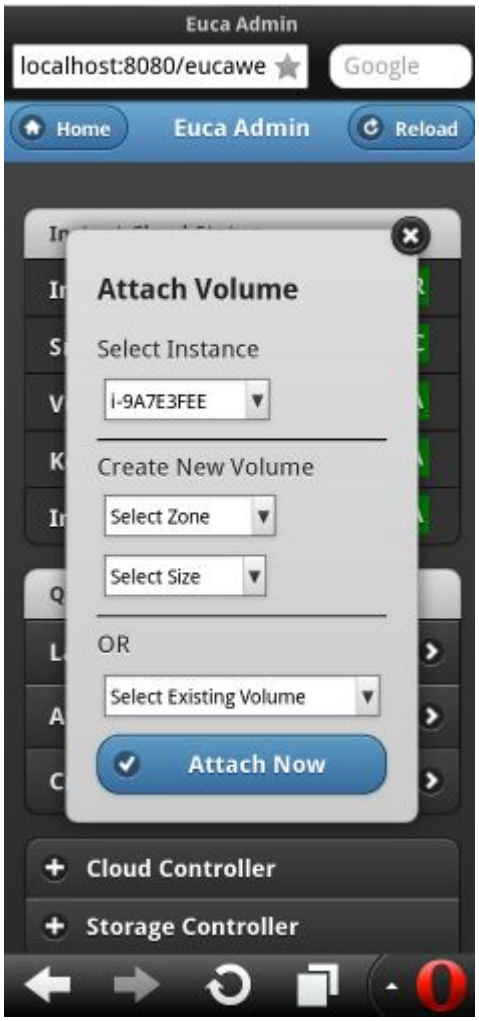

Figure 7: GUI for Attaching Volume to Instance

3. Creating a snapshot
   A snapshot can be used as a backup for a volume. It can be created by providing Volume ID and description (optional). One can re-create volume from snapshot.

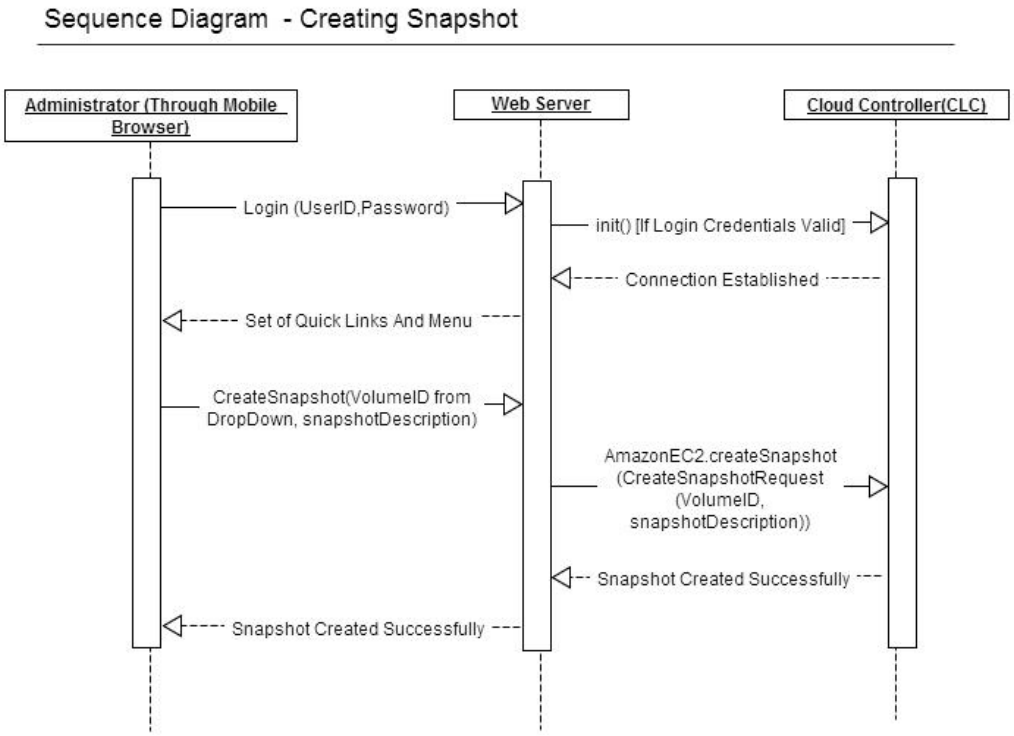

Figure 8: Sequence Diagram for Creating Snapshot

Figure 9: GUI for Creating Snapshot

4 Results

Controlling a cloud through a mobile is not really a simple task. Even though the processing capability of a mobile phone has increased in the past few years, we need to worry about various issues like the amount of memory the application will be going to take, its reponse time, its ease of use and also the personalization of the application.

The first major challenge was how to interact with the cloud server. Initially, we started with SSH Java Library but that is not the best way, so we tried to find if any supportive API was available or not. Then we came across the AWS API, but found no support for the latest version, then we did some regression testing with the old API and found that AWS API version 1.2.12 is supported to some extent. Even though the functionality was limited, but we were able to implement most of our use cases.

We tested our application using multiple smart phones as well as tablets. Some of the major screenshots are displayed below.

Figure 10: Home Screen (Dashboard)

Figure 11: List of Instances

Figure 12: List of Snapshots

5 Conclusion

We started our project by setting up Eucalyptus cloud on two different machines. We selected the Eucalyptus platform, as it is an open source and also has great collaboration with Amazon, a leader in cloud computing. We did time synchronization using NTP across all systems. Firstly, we tried to make an android application for controlling and monitoring the cloud, but later realised that it has no AWS Android API support. Then we switched to Mobile Web application using AWS JAVA API. We made use of mobile jQuery extensively along with CSS3 for designing an attractive interface which is compatible across almost all smart phones.

Our application can run on any mobile/tablet having Internet access. This application has quick links for most frequent activities such as launching instances, creating snapshot, attaching volume snapshots (all these activities can be performed in just 2 clicks). This also has a tab view just like hybridfox so as to keep the administrator familiar with the application. We have implemented twenty-five use cases in in our application, divided into three categories namely cloud controller, storage controller and network controller. As the application is build over AWS API, we can even control Amazon cloud with this application just by configuring the authentication details.

Our mobile web-application provides various features involving instances, images, volumes, snapshots, etc. It is also compatible with any cloud based on AWS and Eucalyptus. Just by registering the secret key, access key and the endpoint url, one can monitor multiple cloud servers using the same application.

References

[1] A. Lonea, D. Popescu, and O. Prostean, "A survey of management interfaces for eucalyptus cloud," in *IEEE 7th International Symposium on Applied Computational Intelligence and Informatics (SACI 2012)*, Timisoara, Romania, May 2012, pp. 261–266.

[2] I. Khan, H. Rehman, and Z. Anwar, "Design and deployment of a trusted eucalyptus cloud," in *IEEE 4th International Conference on Cloud Computing (CLOUD 2011)*, Washington, DC, Jul. 2011, pp. 380–387.

[3] A. Tanwer, A. Tayal, M. Hussain, and P. Reel, "Thin apps store for smart phones based on private cloud infrastructure," in *ITU-T Kaleidoscope: Beyond the Internet? - Innovations for Future Networks and Services*, Pune, India, Dec. 2010, pp. 1–6.

[4] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *IEEE/ACM 9th International Symposium on Cluster Computing and the Grid (CCGRID 2009)*, Shanghai, China, May 2009, pp. 124–131.

[5] Johnson, K. Murari, M. Raju, Suseendran, and Y. Girikumar, "Eucalyptus beginners guide - uec," 2010.