Software Requirements Specification

Akshay Jadhav

Pranav Salunke

Prof. Vinit Kotak

Table of Contents

1. Introduction

1.1 Purpose

To build a Cloud-based System that enables users to have files present on their devices.

Users have multiple devices based on different platforms and access to their files and the service from these devices. Thus, the user environment is fragmented and the need to organize files and folders across these devices is necessary. Since a user can modify one particular file from multiple devices at different instances, it becomes necessary to check for the integrity of this file. A user environment involving more than one user-access to a folder makes it mandatory to have a system that keeps the management of files and folders in check. Also, checking for data loss is one factor that needs to be of the highest priority.

Hence, the aim to build this system would be to deliver (nearly real-time) file synchronization across multiple devices, ensure there is no data loss. A solution for explicit control over deployment of a cloud and its configuration and upgrade options would be provided.

The system is a valuable one to the market since it not only provides users cloud storage along with synchronization but also provides businesses to setup a local Cloud environment as we would be delivering NaaS, IaaS, PaaS and SaaS.

1.2 Overview

Along with real-time synchronization, our system makes sure that the data is securely stored on and transferred among the users' devices. This is implemented with three levels of security:

> File transfer is done securely over a SSL connection. This ensures secure data transfer during synchronization between clients.
>
> Secure encrypted storage is also implemented which is missing in most of the current applications of similar functionality.
>
> Security on the third level is the most important as our services will be monitored and data stored and the other information will be stored on a database that will be implemented on cloud.

We would be taking an operational approach to help automate our cloud system configurations and management; we will be integrating tools and processes from software development methodologies into IT operations (often referred to as “DevOps”). To take the “Cloud Operations” model even further, we would be integrating it with an open source platform model. Explicit control over deployment, configuration and upgrade options would be taken care of along with Monitoring, scaling and adjusting deployment parameters in real time.

1.3 Project Scope

The system has been developed so as to keep in mind the fragmentation of the market in terms of device configuration. Keeping in mind the fragmentation of the market, the system is designed to serve mobile devices such as tablets, smartphones, laptops, etc. and make the user experience truly mobile.

Since the project happens to be in the initial stages, an application to serve Linux and Windows based users has been developed. Support for users using devices based on popular platforms such as iOS, Android and Windows would be added at later stages.

Since the project delivers SaaS along with NaaS, PaaS and Iaas, it has a vast scope since organizations now-a-days require a sophisticated and independent infrastructure to handle ever increasing data traffic and volume storage needs.

1.4 References

[Canonical Ltd.] ubuntu.com

[Canonical Ltd.] juju.ubuntu.com

[Bruade] The principal source of textbook material is “Software Engineering: An Object- Oriented Perspective” by Eric J. Bruade (Wiley 2001).

2. Overall Description

2.1 Product Perspective

The application is a cross-platform, self-contained and independent product. It is an Open Source project that targets the areas in cloud computing which has not been implemented successfully on a large scale.

Our application provides nearly real-time file synchronization.

Deployment of Cloud Infrastructure would be fully automated using latest technologies such as MaaS and Juju, which would also be used for scaling purposes.

Centralized management of the Cloud Infrastructure would be established using OpenStack.

Integration with Public Clouds such as Amazon EC2 would be possible and therefore, the Cloud could seamlessly be converted to a Hybrid one, if required.

Hence, the project focuses not only on providing SaaS but also on providing a sophisticated approach to deploying and managing the Infrastructure required to deploy the service.

2.2 Project Features

Nearly real-time synchronization of files
Automated Deployment of Cloud Infrastructure
Centralized management of Cloud Infrastructure using OpenStack
Seamless integration with Public Clouds such as Amazon EC2, etc.

2.3 Operating Environment

Operating System: Ubuntu 10.10+

2.4 Design and Implementation Constraints

The Linux based application will work on Linux Kernel 2.4.0 + and preferable operating system is Ubuntu 9.04+

The system will use industry-standard 128-bit encryption with an SSL certificate for communication.

2.5 User Documentation

The product website would have an animation to describe the functioning of the application.

2.6 Assumptions and Dependencies

Each user would be assumed to have at most 3 devices per account.

3. System Features

Nearly real time multi-way file synchronization.
Seamless transfer of files between users' machines.
when disk space on the Cloud gets exhausted
Compatible with a plethora of platforms.
Major computing done at Middle-tier leading to a lighter user application that requires minimal system requirements and hence, is more portable.

4. Software/System Requirements

4.1 Hardware Interfaces

Server Side: Middleware

Operating System: Linux Ubuntu 12.04/12.10 Server

Processor: Intel Chipset/AMD Chipset (VT Enabled)
Memory: 8GB

Hard Drive Capacity: 500 GB expandable

Server Side (IaaS)

Cloud Controller Node (Runs network, volume, API, scheduler and image services)

Operating System: Linux Ubuntu 12.04/12.10 server
Processor: Intel Chipset/AMD Chipset (VT Enabled)64-bit x86 _ Two NICS

Memory: 12GB

Hard Drive Capacity: 30GB (SATA or SSD)

Volume Storage: Two disks with 2TB (SATA) for volumes attached to the compute nodes.

Network: one 1GB Network Interface Card.

Memory: 16GB

Compute Node (Runs virtual instances)

Operating System: Linux Ubuntu 12.04/12.10 server
Processor: Intel Chipset/AMD Chipset (VT Enabled) 64-bit x86 – Two NICs

Memory: 32GB

Hard Drive Capacity: 30GB (SATA or SSD) Volume Storage: Two disks with 2TB (SATA) for volumes attached to the compute nodes.

Network: Two 1GB Network Interface Cards.

Memory: 16GB Hard Drive Capacity: 750GB expandable

Client Side

Operating System: Linux

Processor: Intel Chipset/ AMD Chipset

Memory: 256 MB

Hard Drive Capacity: 60GB

4.3 Communications Interfaces

Broadband Internet
NIC card

Appendix A: Glossary

iOS: Apple's mobile operating system

IaaS: Infrastructure as a Service

MaaS: Metal as a Service

NaaS: Network as a Service

NIC: Network Interface Card

PaaS: Platform as a Service

SaaS: Software as a Service

SSL: Secure Sockets Layer