CHAPTER 3

AN AUDIT OF SERVER HARDENING IN MULTI-TENANT ARCHITECTURE SYSTEM IN CLOUD COMPUTING

Cloud computing technology is having trust on sharing computing resource and services over internet. The massive requirement of cloud computing need in information technology is also increasing security threats on the other side (Padhy et al. 2011). The “Server Hardening” is a technique and process of implementing security through a variety of means, thus making the system secure either single or multi-tenant based architecture in cloud computing and it can be applied for either Windows or Linux or UNIX based system.

System security is one of the high and important priorities for enterprises. All the systems built on the foundation of confidentiality, integrity and availability. This chapter highlights the general principles of hardening Linux OS and explores the ways a management tool could be used to discover its OS vulnerabilities and patch up the loopholes. The reason for hardening is to maintain the system in the most secured state possible while maintaining its functionality and minimizing possible threat factors. Hardening was not only about hardening an OS, but also to harden all the components on the host machine. The OS could be compromised due to wrong configuration, buggy device driver, other poor design system or code running on top of the OS. The core theme to avoid vulnerability is, keep the server light, simple and strong security process.
The major contribution is to secure the server by using server hardening technique which makes that

- Remove unwanted services running on the server
- Remove unwanted packages or installation
- Change the default configuration and setting in the server
- Protect the server with firewall enabled
- Keep upgrade the security and server patches always.
- Maintain the problem alert mechanism if any unwanted behavior or intrusions happens in the server end

To have all the above mentioned protective approach, enterprises can build their own custom tool or script to apply all at once to save cost and time. In the typical server hardening process, most of the system administrators are not well aware of server hardening, sometimes it depends with their proficiency as well. In the typical process, system administrators should write their own programming script to implement this process and there are circumstances the other system administrator might not aware of the programming which can cause not to extend further. This becomes cumbersome to the enterprises. By using a server auditing tool which will resolve this issue and it will be upgraded or extended by the dedicated team frequently. So the system administrator can have audit tool knowledge instead of programming.

This chapter explores and highlights the existing and fundamental security configurations with the default installation of UNIX family Operating System installation that should be performed in order to harden the security posture. This will help us to identify common mistakes
and weaknesses in configuring a UNIX family servers and the result of such kind of weakness. At the same time, it will not explain about complete security and server hardening technique guide. Yet it outlines the some major portion of hardening technique of a system, so that it cannot be an easy target for attacks. In general, Most of the system administrators do not realize the fact that a default installation will have lots of vulnerable to allow a variety of attacks. So this document helps for the basic security and industry best practices to secure the Linux Server as well as some more popular application services that commonly run on a Linux Server.

3.1 Server Hardening

Server hardening is probably one of the most important tasks to be handled in the enterprise servers, becomes more understandable when enterprise understands all the risks that are involved. To protect the servers, the enterprise must establish solid and sophisticated server hardening policies (Ant’on et al. 2004) for all servers in the enterprise. Developing a server hardening checklist would likely be a great first step in increasing the server and network security. Make sure that the checklist includes minimum security practices that enterprise expects from their staffs. If they go to the consultant, enterprise can provide them with their server hardening checklist to use as a baseline. Servers are relied upon to deliver data in a secure, reliable fashion. There must be assurance that data integrity, confidentiality and availability are maintained. One of the required steps to attain this assurance is to ensure that the servers are installed and maintained in a manner that prevents unauthorized access, unauthorized use, and disruptions in service.
Server hardening is the process of implementing security through configuration by only having minimum and required packages or services for the operating system. There are the main aspects that are inherited with server hardening implementation process like to ensure security, availability and reduce operational cost.

The approach towards server hardening should start by creating a checklist. Let’s peek into what the model checklist would like for server hardening in the Operation System and Application server side.

- Avoid Using FTP, Telnet and rlogin / rsh. Instead, need to use OpenSSH, SFTP or FTPS
- Keep operating system up to date, especially security patches.
- User Accounts and Strong Password Policy Password Aging
- Restricting Use of Previous Passwords
- Locking User Accounts After Login Failures
- Do not permit empty passwords
- Disable root Login. Logging and Auditing Disable unwanted Services Configure the system firewall (IP tables) and consider also using a hardware firewall
- Encrypt Data Communication
- Linux Kernel /etc/sysctl.conf Hardening Separate Disk Partitions, Protecting Files, Directories and Email
- Use A Centralized Authentication Service (OpenLDAP)
- Install And Use Intrusion Detection System
- Check the listening network port and disable if any unwanted ports
- Lockdown Cron jobs
- Disable USB stick detect
- Disable reboot process (Ctrl+Alt+Del) from inittab
- Monitor user activities
- Enable alert mechanism
- Disabling Runlevel system services
- Disabling Xinetd services
- Reviewing boot scripts which are configured in the inittab
- Restricting network and system access from another network

3.1.1 Types of Server Hardening

It is an evidence and crucial that the changes to be made in the systems for harden, will not higher impact on system and application level in the specific business environments. The fundamental idea of hardening approach can be same, but that has been classified as three major different types in real time.

- **Operating System** - It is a process which can make an operating system as more secure from the internal or external attacks. It has numerous of actions such as update the latest or security patches, boot up security, clean up the unused or temporary files, remove the vulnerable applications, install firewall and antivirus, prevent the execution of malicious code and configure the hardware (like network card, hard drive, USB
Port, CD-ROM Drive and any external devices) components in the OS level.

- **Network** - Network hardening includes preventing unauthorized switch port access to detecting and preventing unauthorized network traffic from both inside and outside the corporate network. This technique applies to Routers, Switches, Firewall, Intrusion Detection/Prevention Sensors and Servers

- **Application Server** - It includes application-based preventative measures application administrative, application isolation, implement access control list (ACL), enable secure socket layer (SSL), remove unwanted user to access application, rename the default application server ports, unauthorized users are not allow to access application log file and application development security.

### 3.1.2 Server Hardening Implementation High Level Process

Enterprises should have a high-quality hardening checklist before implementing the hardening process and it also requires some groundwork.

- Identify the required security level. Sometimes enterprises will have their own security policy requirement

- Gather the specifications of the current server or system in details

- Identify the things which need to be harden

- Estimating the cost of the hardening process
- Identify the right tools or scripts for hardening process. If the available tool or script does not meet their security policy requirement, they have to start building their own tool or script.

**Figure 3.1 Server hardening implementation layers**

This hardening process can be adapted in various layers or level like hardware, O/S and application and the figure 3.1 shows the implementation stages in the layer. As part of the implementation, the first step is back out or rollback process that should be ready. Then enterprises should be prepared with the server hardening checklist. The next step is that the prepared checklist should be implemented in layer wise and parallel it has
to be validated. At the last setup, overall validation has to be executed by enterprises to make sure that it compliance with enterprises standard or policy.

3.2 Auditing Tool

Before to implement the hardening steps, need to know about the existing hardening index level, so that enterprises can plan for checklist which has to be improved or not. For that, “Lynis” is tool which is identified to find the index level in any UNIX family and Mac operating systems.

Lynis is an open source system and security auditing tool which can be very helpful to system administrators, security professionals, system auditors and consultant to determine the security defenses of the Linux/Unix-based systems. It runs on the host itself to perform extensive security and vulnerability scan. Lynis runs in almost all UNIX family systems which are below:

- Solaris
- Mac OS
- AIX
- HP-UX
- Linux
- FreeBSD
- NetBSD
- OpenBSD
It is very easy and more flexible to use by anyone. Since it is written as shell script, end user does not want to install to execute. Moreover, there will be another advantage of this tool is that if the end user is expert in dealing with shell script, they can customize this tool for their convenient because it is released with open source software (GPL) license.

3.2.1 Auditing Tool Usage

The security scan consists number of steps to be executed by the tool and end of execution it will show a report which contains all the security scan metrics with hardening index level. This report is also stored as a log file which can be used by anyone for further reference. The security scan involves with:

- In figure 3.2, the Lynis tool verifies and prints the Operating System and its database details.

Figure 3.2 Operating system detail shows by Lynis tool
- The latest versions of Lynis tool should be always recommended and the figure 3.3 shows the command to check the latest version.

```
root@lynis# ./lynis --check-update
== Lynis ==
Version       :    1.3.8 [ Up-to-date ]
Release date  :     25 December 2013
Update location:     http://rootkit.nl
```

**Figure 3.3 Lynis tool checks with latest version**

- In figure 3.4, the scan process initial starts and also the subsequent screens will shows other available tools, utilities and security policy rules which are installed in the system.
Figure 3.4 (a) Lynis tool initiates the scan process

Figure 3.4 (b) Lynis tool shows the installed software and database
Figure 3.4 (c) Available security compliance shows by Lynis tool

- In figure 3.5, Lynis tool displays the scan report with the present security depths (hardening index level) for the system.

Follow-up:

- Fix findings, see security controls overview and documentation
- Upload data to Lynis Enterprise for further analysis
- Create a report and implementation plan

Enterprise support and plugins available via CISOFy - http://cisofy.com

```
Hardening index : [76] [########################################]
```

Files:
- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat

Tip: Disable all tests which are not relevant or are too strict for the purpose of this particular machine. This will remove unwanted suggestions and also boost the hardening index. Each test should be properly analyzed to see if the related risks can be accepted, before disabling the test.

Lynis 1.4.5

Figure 3.5 Lynis tool shows the present hardening index level
3.2.2 An Audit of Server Hardening in Multi-tenant Architecture System in Cloud Computing

Enterprises can gain systems from cloud environment based on single or multi-tenancy architecture. For either case, the attained system can have more vulnerability (Gonzalez et al. 2012) and it will be highly risk to enterprises. The figure 3.6 visually shows the advantage of running the script and presents an understanding of server hardening in single or multi-tenancy architecture.

The approach towards server hardening should start by creating a checklist. Once the checklist is ready with us, it’s as simple as following the checklist. It is tedious to manually carry out the disabling of the unwanted services, root logging, configuring a system firewall, Linux kernel hardening and install an Intrusion detection system.

Figure 3.6 An Audit of Server Hardening in Multi-tenant Architecture System in Cloud Computing
Thus, this process needs to be automated. The checklist that is provided in this section is just a basic checklist; it can be made exhaustive thus, making the system impenetrable to the attacks from external sources. It is just a matter of updating the shell script for enterprise needs and executes it. The shell script can be automated to run while booting so that the system can be secured in the startup before to work on.

3.2.3 Typical Audit Scenarios for Server Hardening in Cloud System

There are audit scenarios which is required practically in the real time environment to improve privacy and security requirement (Breaux et al. 2008). The scenarios are list as below:

- Security purpose auditing and compliance (PCI, HIPAA, SOx) auditing
- Vulnerability detection and scanning
- System and software hardening

The server hardening is measured as an index in the Lynis tool. The tool was run and various parameters were listed, the most useful for us being the “Hardening Index”. The figure 3.7 shows that the hardening index of various Linux operating systems which were created from the cloud instance is measured and a chart for the same is generated. It also shows the analysis report that the hardening index of Red Hat Linux operating system is the best among other Linux versions. But the Hardening quotient is not high.
Figure 3.7 Chart representing the of Hardening Index of Linux OS

The Table 3.1 shows that the various Linux operating system versions, kernel version and hardware platform with hardening index.

Table 3.1 Hardening Index of various Linux Operating Systems

<table>
<thead>
<tr>
<th>Amazon EC2 Instance Type</th>
<th>Kernel version</th>
<th>Processor</th>
<th>Harden Index (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Linux 3.4.73</td>
<td>x86_64</td>
<td>x86_64</td>
<td>53</td>
</tr>
<tr>
<td>Ubuntu 12.04</td>
<td>3.2.0-58-virtual</td>
<td>x86_64</td>
<td>45</td>
</tr>
<tr>
<td>Ubuntu 13.1</td>
<td>3.11.0-15-generic</td>
<td>x86_64</td>
<td>45</td>
</tr>
<tr>
<td>Red Hat 6.4</td>
<td>2.6.32-358.14.1.el6.x86_64</td>
<td>x86_64</td>
<td>58</td>
</tr>
<tr>
<td>SuSE ES 11</td>
<td>3.0.82-0.7-ec2</td>
<td>x86_64</td>
<td>49</td>
</tr>
</tbody>
</table>

All the instances which are created as cloud instance, in the Amazon Web Service. Thus, this chart proves that the instance is susceptible to various attacks.
CHAPTER 4

A MODEL OF SECURITY ARCHITECTURE ON PRIVATE CLOUD USING OPENSTACK

Cloud computing is becoming more widely utilized, it is important for enterprises to understand ways to maximize benefits and minimize risks of a move to the cloud also to carefully consider its operating expenses essential. Since most of the enterprise already has traditional IT setup and the major investment would be spent for having as a data center which has limitation and difficulty to change the amount of storage and workload.

The Private cloud is a model or architecture and often presented as being the solution for all computing issues in the enterprise. It is distinct and secure cloud based environment which can be accessed and managed by the enterprise. It is very closer to the more traditional model of individual local access networks (LANs) that is used in the past by enterprises but having the added advantages of virtualization. This can be also called either “Internal” or “corporate” or “enterprise” cloud and it will be protected by a firewall. The enterprise will have more control over its data and applications with this kind of setup. It also promises benefits such as energy savings, cost savings, rapid deployment and customer empowerment.

There could be additional security offered by ring fenced cloud model that could be ideal for any enterprises which needs to store and process private data or carry out some sensitive tasks. For instance, a private cloud service could be utilized by a financial company that is required by regulation to store sensitive data internally and who will still want to benefit from some of the advantages of cloud computing within their business infrastructure,
such as on demand resource allocation. On the other hand, technically the virtualization is not private cloud and private cloud is far beyond virtualization. Data Storage is the one of the important and primary resource enterprises wanted to keep with their control.

4.1 Private Cloud – Strategy Planning

Enterprises has to build a strategy or roadmap for the private cloud planning. The figure 4.1 shows the model for the planning and each will be described below. Since the model should be a reiterative process, it can have more repetition.

Figure 4.1 Roadmap - Private cloud build strategy

- **Application Classification** - need to classify the list of applications that can be moved to the private cloud.

- **Measure Application** – need to measure the impact or importance of the application and make it in the proper queue then this can be taken with the order.

- **Categorize Application and Platform** – most of the enterprises should be having a heterogeneous application and platform, need to categorize by platform.
- **Migrate** – need to start migrate the application with the respective platform and also has to think about the support services for the platform that enterprise builds.

### 4.2 Private Cloud Challenges

There are key challenges that should be considered before the enterprises to build private cloud. So enterprises that

- Should think of bridging private cloud with the existing infrastructure or integrate with legacy systems and data.
- Should be able to have 24x7 support service for their end user when they experience any issue in private cloud.
- Have enough on-premises servers that can be virtualized and scaled for current and future demands.

There are some additional challenges that businesses need to consider on day-to-day business management.

- Managing the enterprise applications and responsible for updates and patches in the applications.
- Handling the security issues using like cryptography and authentication.
- Monitors and tests the system to ensure data and applications are properly backed up and readily retrievable.
- Managing the business critical deployment in the application and those activity tracking.
4.3 Architecture of OpenStack

OpenStack (OpenStack, 2012) is a free and open-source software cloud computing software platform. Users primarily deploy it as an infrastructure as a service (IaaS) solution. OpenStack works with popular enterprise and open source technologies making it ideal for heterogeneous infrastructure. It is highly configurable for many reasons. OpenStack community is one among the fastest growing open source communities in the world.

The figure 4.2 shows the series of OpenStack’s interrelated components that control pools of processing, storage, and networking resources throughout a data center in which the user manages these all, through a web-based dashboard and command-line tools. The aim of OpenStack is to produce open source cloud computing platform that will meet the needs of public and private clouds regardless of size by being simple to implement and massively scalable everywhere.

![Figure 4.2 An Architecture for OpenStack](image_url)
Each component is used to deal with specific resource in the overall OpenStack cloud platform which is explained below in detail.

- **OpenStack Identity Keystone** – Keystone (Keystone, 2012) is responsible for managing the restricted access services like authorization and authentication details for all other components in OpenStack platform which provides a central directory of users mapped to the component services.

- **OpenStack Networking (Neutron)** - This is pluggable and scalable API driven component to manage networks for all the virtual hosts (instances) and provide IP address both private and public. End user or enterprises can control and monitor the virtualization by dashboard called “Horizon”.

- **OpenStack Horizon (Dashboard)** – This component has web based portal to interact with all other OpenStack components to administrate the resources. It has core support to ship with three different dashboards like User, System and Settings dashboard. For the dashboard application development purpose, the Horizon application comes with a set of abstraction APIs for the core OpenStack components in order to provide a consistent, stable set of reusable methods for developers.

- **OpenStack Volume (Cinder)** – It can be detachable block storage to any virtual instance similar to an external hard drive. It provides the APIs to create and manage a service that provision the storage in the form of block storage called as “Cinder volumes”. It also features basic storage capabilities like snapshot or volume clones and management which are often enhanced through vendor-specific drivers.
- **OpenStack Compute (Nova)** - It allows to create and manage virtual servers using the machine images through the dashboard or API for Infrastructure as a Service (IaaS). It is the core service component in the OpenStack platform to build a cloud which provisions and manages even large networks of virtual machines.

- **OpenStack Image (Glance)** - It is a imaging service access by Restful API which helps to discover, register and retrieve virtual machine images. It follows the client-server architecture to make VM images available and it can store in a various location from simple file systems to object-storage systems like the OpenStack Swift project.

- **OpenStack Object Store (Swift)** - It is used for Object storage that can be highly scalable redundant unstructured data store designed to store large amounts of data with lower cost. Since it supports highly scalable architecture which means that it can scale to thousands of machines with tens of thousands of hard drives. It is designed to support horizontally scalable so here is no single point of failure. It can be more appropriate for large-scale deployments as the cluster grows larger. It can store static objects like multimedia files, backups, store images, archives etc.

### 4.3.1 Key Benefits of OpenStack Private Cloud

- Open Architecture – any component that can be adopted and used.
- Robust – there should be common or centralized administration to obtain best performance benchmarking.

- Scalability – automatic infrastructure creation when the server gets high load or downtime.

- Hybrid Cloud – bridge private and public cloud.

- No Downtime – environment should be always up and running.

- Client Support – there should be defined SLA and 24/7 support for the users.

4.4 A Model of Security Architecture for Private Cloud using OpenStack

The basic requirement to setup private cloud environment using OpenStack is network and right hardware. Since OpenStack allows us to setup everything in same flat network, it will not be the best approach for security reason. At least two different network (support VLAN) environments (Nurmi et al. 2009) are strongly suggested for the best model to build a secure private cloud. In this, one can be used to manage the network traffic and for virtual machines or cloud instances to communicate with to each other. This means each compute nodes need to have two network cards and the network manager. Each network should be configured and run on different IP ranges.

The next step is to identify and determine what and where to deploy all the OpenStack components. The minimal requirement and mandatory components to build private cloud in OpenStack platform is that networking,
compute, keystone and storage. In figure 4.3, shows the model architecture to build secure private cloud. A classic deployment should have one controller and series of compute nodes. The controller manages the message server, database and other components to orchestrate the private cloud when the compute nodes run the instances.

![Diagram of Security Architecture for Private Cloud using OpenStack](image)

**Figure 4.3 A Model of Security Architecture for Private Cloud using OpenStack**

There can be also another approach to break out the individual pieces of the controller for the better performance such as placing the database server (e.g., MySQL database) on a different physical box. To ensure the security measure, each of these pieces is installed on a secure host and that will be attached into the network which is required to keep the private cloud running.
As part of best practice for enterprises private cloud to end user that web management and administrative console and network manager can be exposed. For the security practice, by default the console runs with “http” protocol and it should be configured with Apache and SSL to make sure it runs with “https” secure protocol. Additionally, all these servers need to be hardened (Chapter 3) and back-end traffic isolated from user connections.

4.4.1 OpenStack Components

4.4.1.1 Security Architecture using Keystone Identity Service

Keystone provides a single point of integration for OpenStack identity, token, catalog and Policy services for projects. Keystone ensures the listed below things to help us to model the secure architecture to build private cloud.

- **User** - incoming requests are from valid or approved from person, service or system

- **Role** - group of user assigned to set of privileges and perform specific operations.

- **Credentials** - specific user provides username and password or an authentication token.

- **Authentication** - identity service issues authentication token that user is allowed to make subsequent requests.

- **Token** - each token has a scope describing accessible resources. A token may be revoked at any time and is valid for a finite duration.
- Endpoint - network accessible address which usually described by URL for authenticated users.

There are two primary functions which can be achieved by using keystone.

- User Management - tracks of user and their security scope in which they are permitted to do.

- Service Catalog - provides a catalog of what services can be available for the user.

4.4.1.2 Networking Management using OpenStack Networking (Neutron)

OpenStack Network is called as “Neutron” and it provides “networking as a service” between interface devices which is managed by other OpenStack services components. It allows to do list of below options:

- To build complex and rich “Network” topologies by applying advanced network policies.

- To create “Subnet” to represent IPv4 or IPv6 address blocks from which IPs to be assigned to VMs on a given network are selected.

- Allows us to create “Port” that represents virtual or logical switch ports in the network. Virtual instances attach their interfaces into ports and the logical port defines the MAC address and the IP address which is assigned to the interfaces plugged into them.
4.4.1.3 Infrastructure Management using OpenStack Compute

The main strength of OpenStack Compute is to support different virtualization technologies (King et al. 2006) in the infrastructure level.

- Manage virtualized commodity server resources like CPU, memory, disk, and network interfaces
- Programatically allocate IPs and VLANs to manage networks
- Designed for automation and security to make it easy end user to manage compute resources and prevent users from impacting each other with excessive API utilization
- Supports the distributed and asynchronous architecture.
- Virtual Machine (VM) image management like run, reboot, suspend, resize and terminate instances
- Security Groups and Role Based Access Control (RBAC)

4.4.1.4 Storage and Backup Management using OpenStack Storage

OpenStack Storage supports Object and Block storage with different and unique architecture for deployment and many enterprises have variety of storage needs with respect to performance and price requirements.

- Reliability and data redundancy to protect from failures
- Scale vertically and horizontally-distributed storage
- Account/Container/Object structure and Built-in replication
- RAID not required
- Fully integrated with Compute to attach block volumes and exposed the usage report.
4.4.2 Monitoring Dashboard (Horizon)

OpenStack Horizon is a web based GUI application and it provides a custom management interface to create a typical base dashboard structure for the OpenStack underlying resources. It brings all OpenStack projects together in a single window. It is also exposed as an API so that end user can create their own custom dashboard using the API and framework. It is built on Django with Python. Django helps to create the complex database-driven websites and it emphasizes reusability and plug ability.

4.4.3 Applications (Corporate)

Applications are business specific and owned by enterprises or corporate. It can be mobile or web based application. In this dissertation, there are two main applications (Chapter 6) are written, which are called “File Management Application (FMA)” and “ Efficient File Transfer Management (EFTM)” . FMA is a web based GUI application and EFTM is a service based offline application, but both are deployed in the tomcat web server.

4.5 Design and Implement Private Cloud Infrastructure

To maximize the benefits of a private cloud to be sure the enterprises to design and implement by choosing best and flexible cloud automation tools and processes which can avoid unnecessary overhead to hire more system administration. These are key points to be implemented:

- **Automation** - Infrastructure automation provisioning which can be achieved by automation tools. These are called as “DevOps” tools (DevOps for Developers, 2012).

- **Resource Management Dashboard** - The cloud management dashboard is essential and the infrastructure automation provisions are carried out behind the scenes by scripts. It
presents an easy-to-use interface that allows users to provision and de-provision resources, track resource use, modify the access controls to resources and view services available in the private cloud.

- **Workload and Service Monitoring** - Cloud automation monitoring will be useful for real-time monitoring (hardware failures or disrupted services) and long-term business support planning (aggregate information like use of cloud resources, demand for particular types of resources and costs).

- **Virtualization** – It allows the resource sharing with more than one environment. The resource will be like operating system, storage, server or network components.

### 4.6 Post Private Cloud Migration

Once the private cloud is up and running, then the enterprises has to provide the right tools to help users to maintain and get maximum benefit out of cloud services. There are few important points highlighted below.

- Implement automation for the actual private cloud benefits. This can be done by using some famous DevOps tools like “Puppet” or “Chef”.

- Meter the private cloud usage on “Bill for the usage or infrastructure” basis.

- Security tools are essential for the private cloud and it can be done by using self-service model.

- Maintaining a private cloud which means that server failover and downtime must not lead any problem.

- Determines that it is cost-effective with “Return on Investment (ROI)” software or tools.
CHAPTER 5

A MODEL ARCHITECTURE OF DATA COMPRESSION AS A SERVICE IN CLOUD

Data compression is very important as it was in the early days of computing. Though all computer and storage resources were very limited in earlier days, the data process and storage in use were much minor than today. In the modern information technology space, the data processing by enterprises or applications require storage for large volumes of data and they are spending huge investment to storage devices. Storage is often very expense in the data center. This technique is to reduce the amount of storage required to meet the needs of the business (Blelloch et al. 2002).

Data compression can be a branch of information theory in which the primary objective is to reduce the amount of data to be sent in and out. It follows a technique in which the file size is reduced by applying some standard or user-defined algorithm to re-encoding the file data using fewer bits of storage than the original file. The original file can then be recreated from the compressed format using a reverse process called decompression, by using the same algorithm.

The compression technique may vary based on the data type or format like audios, videos, images and raw text. The compression is classified as lossless and lossy method. Lossless compression reduces data bits by identifying and eliminating the redundancy, but information is will be lost in lossless compression. Lossy compression technique reduces the data bits by identifying redundant information and removing it. The lossless
(LZW compression) technique (Parvinder et al. 2006) can be applied for raw text data where as lossy is for binary data like image, audios and videos. In this chapter, the lossless data compression technique is used to build the compression service (Salomon et al. 2004) that is commonly used to archive and transmit the text oriented content, but not binary content.

5.1 Key Requirement for Data Storage

In the context of data storage reservation and data transmission between application and data center, should have more benefits but cost money. In general, Information is in the form of file. The more information being dealt with, the more involves cost. The fundamental of data compression technique proves that takes the steam of symbols to transforming them into codes.

Data Storage can be managed by storage administrator and they can determine how storage is being allocated using storage capacity planning tools that analyze storage systems and then generate a report on available data storage capacity and storage performance.

Storage capacity planning could be the critical and it is the practice of assessing and forecasting future storage requirements to get or procure the enough disk space and that can be purchased to meet the needs of users and applications. Effective storage capacity planning will allow the data storage administrators to delay buying disk by purchases, which is important because the price of disk continues to drop. Saving money on storage resources that would otherwise go unused enables enterprises to better allocate their storage budgets.
5.1.1 Effective way of managing data storage

Compression is best technique to efficiently manage the data storage space or transmission capacity. Reducing the data bits used in representing a piece of information to eliminate the repetition of identical sets of data bits, called redundancy in an audio/video, graphic, or text data file (Gripon et al. 2012). White spaces in text and graphics, large blocks of the same color in pictures, or other continuously recurring data, is reduced or eliminated by coding (encryption) with a program that uses a particular type of compression algorithm (Jain et al. 2009).

5.1.2 Objective of Building Data Compression in Software as a Service

There could be several reasons to model the data compression in Software as a service (SaaS). Software as a service is one of the cloud solution model which is part of cloud computing. The fundamental architecture of SaaS is to deploy as a hosted service and access that through the Internet. SaaS divides the control and ownership of software from its usage and the applications can be accessed from any device through mobile, desktop or any thin client interface.

Since information technology is becoming primary need for small scale to large scale (IT and IT enabled) industries, everyone wanted to manage the data storage effectively. Small scale based enterprise cannot afford some amount of investment for creating or using available best data compression approach for their need. But somehow they could manage to pay the service utilization, having “Pay-as-you-go” model. The idea of making this as service will help industries to have common place (cloud
platform) to access this service over the internet. This could not be implemented in the current IT trend and this can hit the IT space soon.

It is a good option from a traditional on-premise software model and it brings more benefits to customer, provider and enterprises. The end user does not know and worry about to manage or control the underlying cloud infrastructure, like operating systems, network, servers, storage, or individual application capabilities and configuration settings for the SaaS application.

5.2 Data Compression and Decompression Technique

In general, Information is in the form of a file. The more information being dealt with, the more involves cost. The fundamental of data compression technique proves that takes the steam of symbols to transforming them into codes. The Model is the core which contains a collection of data and rules. In figure 5.1, the model helps to define the probabilities for each symbol and coder produces an appropriate code base based on the probabilities.

![Figure 5.1 Flow for Data Compression Technique](image)

In figure 5.2, the model does reverse process to decompress. The probabilities for each code will be interpreted with appropriate symbols base based rules.
Figure 5.2 Flow for Data Decompression Technique

LZW (Lempel Ziv Welch) algorithm (Hasan Md et al. 2011) is most widely used for lossless data compression. In general, this algorithm needs more memory from the system. There are many reasons to build the compression and decompression as a service in cloud which is explained in subsequent sections. Amazon Web service is the cloud platform (Chapter 1) which helps to implement this solution.

5.3 A Model Architecture of Data Compression as a Service in Cloud

The fundamental idea about making data compression technique as a service model is that this service could be utilized by small scale to large scale enterprise for their data compression need. This solution belongs to Software-as-a-Service (SaaS) architecture and since it has been designed and deployed in the cloud platform, anyone can access using this service by internet and also generally there could be a management portal where user can view this service utilization report which can help user to pay for it. In this data compression service, the Lempel-Ziv (LZ) compression (Ozsoy et al. 2011) methods are used and it could be the most popular algorithms for lossless storage.

Each end-user can register them with data compression service through the administrative portal and they get a unique identity key after successful registration. The given unique identity key should be added or
configured into their application which requires to use the data compression API before the data reaches to on-premises storage center.

Figure 5.3 A Model Architecture of Data Compression as a Service in cloud

In figure 5.3 shows the model architecture for the data compression as a service and Amazon Web Services (AWS) is used to provisioning the infrastructure to this architecture and to have public cloud where the data compression as a service deployed (IBM, 2014 and Hisham et al., 2016). The list of below which are part of the implementation and in the architecture.

- Public Cloud –Amazon Web services (Data Compression as a Service)

This module consists of Compression services and Load balancer. The compression service can be written by using languages like Java, .NET, PHP, Ruby and Python.
• **Load Balancer**

  It is a software load balancer which is used by designing apache web server. It makes the service load balanced with help of given attribute like number of user access or data bandwidth.

• **Organization Data storage(on-premises) Center**

  Amazon web services allow enterprises to adapt their convenient scripting language to write this service and also allows user to map their on-premises data center to store their data.

• **Usage and Administration Portal for Data Compression service**

  Cloud service providers will create and share the management portal to enterprises to manage and monitor their utilization for the cloud services. This management portal will be secured and authenticated by using Identity and Access Management (IAM).

• **Multi-tenant model**

  This architecture also provides solution business end user in a multi tenancy model. Multi tenancy architecture simply involves a single instance serving multiple customers. Tenants may also be given the opportunity to customize some features such as color, or business rules, but changing the application codes is impossible and it is only possible for service vendor the one who owns this service (IBM, 2013). This approach is pocket friendly because almost, if not everything, is shared; these include the software development plus maintenance costs. It also allows components like portals and APIs to link enterprise application repeatedly into usage metering services which are normally coupled to billing capacities. Having a record on how enterprise application is doing is critical, since it determines the cost of
the software product. When they have factored the amount, they will need a great mechanism to collect the money; either by bank account transfer, credit card, or even traditional check. Software-as-a-Service is the best approach to get the job done. Multi-tenancy service providers generally will offer a friendly user interface to help manage GUI application configuration and also control enterprise software utilization. This assists experts such as system administrators and integrators to analyze and manage their service components within minutes.