

# Establishing Trust In Cloud Using Machine Learning

Submitted By  
**Reetu Gujaran**  
17MCEI12



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
INSTITUTE OF TECHNOLOGY  
NIRMA UNIVERSITY  
AHMEDABAD-382481

16 May 2019

---

# Establishing Trust In Cloud Using Machine Learning

---

## Major Project

Submitted in partial fulfillment of the requirements

for the major project of

Master of Technology in Computer Science and Engineering (INS)

Submitted By

**Reetu R. Gujaran**

(17MCEI12)

Guided By

**Dr. Madhuri Bhavsar**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

16 May 2019

# Certificate

This is to certify that the major project entitled "**Establishing Trust in the Cloud Using Machine Learning**" submitted by **Reetu Gujaran (Roll No: 17MCEI12)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering specialization in Information and Network Security of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I and part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Madhuri Bhavsar  
Guide, Professor and Head,  
IT Department,  
Institute of Technology,  
Nirma University, Ahmedabad.

Dr. Sharada Valiveti  
Associate Professor and Coordinator,  
CE Department  
Institute of Technology,  
Nirma University, Ahmedabad

Dr. Alka Mahajan  
Director,  
Institute of Technology,  
Nirma University, Ahmedabad

## Statement of Originality

---

I, **Reetu Gujaran**, Roll. No. **17MCEI12**, give undertaking that the Major Project entitled "**Establishing Trust in Cloud Using Machine Learning**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering specialization in Information and Network Security** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

---

Signature of Student

Date: May 16, 2019

Place: Ahmedabad

Endorsed by  
Dr. Madhuri Bhavsar  
(Signature of Guide)

## Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr Madhuri Bhavsar**, Professor and Head, Information Technology Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sharada Valiveti** , Sr Associate Professor and PG Coordinator (INS), Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Reetu Gujran  
17MCEI12

# Abstract

In the modern era of digitization, cloud computing is playing an important key role in it. Tons of user data is on the cloud and to manage it with all the odds is a mesmerizing task. All the major community is on cloud having all their confidential data resides on a cloud and to provide user data integrity. Security is the major concern task for any service provider. In the last few decades research has been motivated to provide security on a cloud with reference to the categorization and classification of security concerns. Growing security risk in the last few years in various components of the cloud is a major concern. Research studies have been motivated to handle risk, threat, and vulnerability imposed within the environment of the cloud. In the cloud, trust is a major concern as a security point of view. In cloud computing, user's data stored on the remote server which may be operating by others and can be accessed through the internet connection. The facilities provided by the cloud are too attractive for customers but it has distributed and non-transparent nature due to some obstacles using in cloud computing service because users lose their control over the data, and they are sure about whether cloud provider trust or not. So, customers confused with cloud providers regard the trust issue. This paper mainly focuses on establishing trust in the cloud using machine learning methods. There are high chances that data may be lost or compromised. So how people can trust that their data is secure or not on the cloud. Hence, trust is becoming a serious issue. In this research study, the main aim is to minimize the security risk, threat and vulnerability as a trusted perspective in a cloud environment. For this, we have study machine learning methods. In machine learning, there are mainly three methods: supervised learning, unsupervised learning and reinforcement learning. Supervised learning means the system already knows what is the output. If the system gets the desired output then closed the process. Unsupervised learning means the system doesn't know what is the output, and the last one reinforcement learning, it is based on reward and feedback oriented. We also discuss trust parameters like availability, confidentiality, accuracy, and integrity, resource management, non-repudiation, risk management, protecting communication, hardware security, reliability, and secure architecture.

# Contents

<b>Certificate</b>	<b>iii</b>
<b>Statement of Originality</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 What is Trust . . . . .	2
1.3 What is Security . . . . .	2
1.4 Machine Learning methods . . . . .	6
<b>2 Literature Survey</b>	<b>7</b>
2.1 Pros and Cons Of cloud trust model . . . . .	9
2.2 Work done by the researcher's in this domain . . . . .	12
2.3 Cloud trust Parameters . . . . .	12
<b>3 Problem Statement</b>	<b>14</b>
3.1 Issues needs to address . . . . .	14
3.2 Problem in the trusted cloud . . . . .	15
<b>4 Proposed Solution</b>	<b>17</b>
<b>5 Implementation and Results</b>	<b>21</b>
5.1 Experiments on public cloud (AWS) . . . . .	22
5.2 Install jupyter in AWS . . . . .	31
5.3 Using AmazonSageMaker . . . . .	38
5.4 Using Linear Regression . . . . .	39
<b>6 Conclusion &amp; Future Work</b>	<b>47</b>
6.1 Conclusion . . . . .	47
6.2 Future work . . . . .	47
<b>References</b>	<b>48</b>

# List of Tables

2.1	A list of Survey Papers on Cloud Trust Model. . . . .	11
2.2	A list of researcher's who has found trust in cloud . . . . .	12
5.1	Dataset . . . . .	40

# List of Figures

1.1	Security Diagram . . . . .	3
1.2	Flow of data integrity . . . . .	5
4.1	Model diagram . . . . .	18
4.2	Security parameter . . . . .	19
4.3	Parameters of SLA . . . . .	20
5.1	Launch the instance . . . . .	22
5.2	Instance description . . . . .	22
5.3	Download key-pair . . . . .	23
5.4	Open the putty configuration . . . . .	23
5.5	Open the terminal . . . . .	25
5.6	Create an image . . . . .	26
5.7	Create an image . . . . .	26
5.8	install jupyter . . . . .	31

# Chapter 1

## Introduction

### 1.1 Overview

As of late, in the period of digitalization, the principle concentrated in on giving the trust in a cloud. Cloud organizations are recognizable inside the private, open and business territories. An impressive part of these organizations is depended upon to be constantly on and have a fundamental nature; as needs are, security, protecting communication, secure architecture, confidentiality and quality are continuously basic edges.

Trust plays an important role in the cloud for security, availability, turnaround efficiency, reliability, IoT, etc. Trust is important everywhere. Trust can be calculated by various methods like Delphi programming, algebraic and etc. The facilities provided by the cloud are too attractive for customers but it has distributed and non-transparent nature due to some obstacles using in cloud computing service because users lose their control over the data and they are sure about whether cloud provider trust or not. So, customers confuse with cloud providers regards the trust issue. this paper mainly focuses on establishing trust in the cloud using machine learning methods. Dealing with a single cloud provider is less popular due to the risk of service availability failure and there is a possibility of the malicious insider in the cloud, which decrease the acceptance of trust issue in the customer's mind. so with the help of the machine learning method, we establish the trust method which increases the acceptance of the trust.

In this research study we have analysis some method for trust establishing in cloud .A recent survey regarding the use of Cloud services, security is the greatest challenge for cloud.In cloud computing we use virtual environment to achieve multi-tenancy, but

vulnerabilities in virtual machines pose direct threat to the privacy, trust and security of the Cloud services .In the security domain availability is major parameter.

## 1.2 What is Trust

Trust means believe in the reliability, truth, or ability of someone or something.trust means increase the productivity, reduce the cost of conducting business and provide the backup option.

Things which make cloud is trusted cloud:-

- The cloud provides the private platform to the users or companies so companies store their data on a private cloud. Therefore, it reduces the cost of conducting business.
- Customers do not need to use a pen drive or hard disk to store or for backup, the data so trusted cloud increases the productivity.
- Make use of service level agreement (SLA), means infrastructure service providers guarantee a minimum quality of service. The QoS is related to the data storage, CPU memory, etc. .
- automatically update and maintain when new things happen in the cloud environment.
- data integrity helps customers to check that, received data is coming from the CSP side or third person.
- Clouds make the life of people very easier because they can share data or any information with high speed.

## 1.3 What is Security

Cloud security is the fast-growing service that provides much functionality like secure the confidential data from the theft, authorization people can use the services. So our data is secure with data centers. Cloud security has issues like confidentiality, availability, integrity, and privacy. These issues are associated with cloud storage. Data integrity associated with the data storage to provides secure data from the cloud. Data integrity



Figure 1.1: Security Diagram

means it provides the original data which is required. Security has many issues like confidentiality and all to explain these issues in detail:

- **Confidentiality:** Confidentiality means the need to protect the data from spoiled access. Only authorized people can access the data.
- **availability:** Information needs to be available to authorized person's only.
- **Data integrity:**  
Verify that received data are exactly the same data or not. It provides perfect data. Any corrupted or deleted data can be timely identified and this is the major point for data recovery..

Data integrity has the following properties:

1. **Unrestricted challenge frequency:** There is no limitation for the number of challenges made by the client to verify the integrity. Data integrity, not a one-time-

activity. When the client executes verification, at that time data integrity identifies the corruption or deletion data.

2. Soundness: The entrusted server should not be able to entrap the challenge request. This property of data integrity ensures data reliability. If the CSP sends a challenge request with corrupted data to the client than the client never identifies corruption data exactly. So the reliability of the data becomes in demand.
3. Stateless verication: The client and server do not need to store previously results to verify future results. Because each and every challenge request is a self-determining form all the past results. This is the essential requirement of data integrity..
4. Robustness: Means identify the data corruption even the data in small size. Not applicable for large data set is the limitation of data integrity, so for this limitation data integrity adopt a probabilistic approach. This approach work independently even the data corruption is too small.
5. Data recovery: In data integrity, it is not sufficient to only determine or find corrupted data or act in an inappropriate manner. Cloud clients also want or interested in complete data or recovery of data. All properties of data integrity identify data corruption but some property also recovers the data. Mostly error correcting code is used for data recovery.
6. Dynamic data handling: Two types of data: static and dynamic. Static data means not change or fix data whereas dynamic data means data may be change, It uses some operations like deletion, insertion, and modification. Dynamic data more challenging compare to static data. Because dynamic data have demanded that the data integrity should remain unbroken even when insertion, deletion, operation implement.
7. Public/Private auditability: Two approaches are there, first one for data owner analysis, and second one third-party auditor. The analysis process for both approaches is implemented without recouping the remote data. Outsourced data can verify by data owner only so this the private analysis. It is not possible that all the time data owners remain online for data integrity confirmation so data owners can give the responsibility to the third-party auditor. So this supports public auditability.

8. Privacy preserving: When the analysis process is going on the confirmation process should protect data privacy. In privacy preserving property, the third-party auditor cannot effect on confidential information of the client.
9. Fairness: Provide the protection to fair CSP, not to unfair user's. If data integrity doesn't support to fairness that means unfair user's harm the CSP reputation.

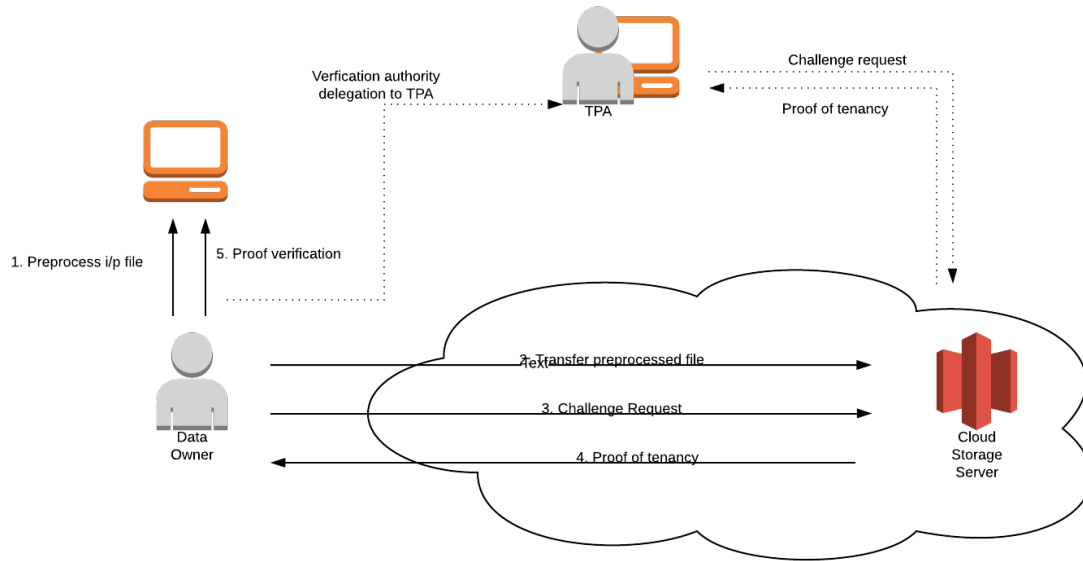


Figure 1.2: Flow of data integrity

## 1.4 Machine Learning methods

Machine learning (ML) is an application of Artificial Intelligence (AI). Artificial intelligence means to learn automatically from experience and improve system ability. The machine learning means focuses on the development of a computer program that can access data and use it too learns for themselves. Machine learning has mainly three types: Supervised Learning, Unsupervised Learning and Reinforcement Learning. These methods are mostly used nowadays. Machine learning methods provide an accurate solution, and it's easy to implement.

**Supervised Learning :**The system already knows what is the output, if the system gets the desired output then closed the process. In supervised learning to calculating trust in the cloud, we are using metrics: Confidentiality Matrix (C) Integrity Matrix (I) Availability Matrix (A) Reliability Matrix (RM). And after that find the rank of these all metrics. There is a limitation when trust is calculated using this method, it uses space and time. If data is not received in a timely manner then this method is failed.

**Unsupervised Learning :**System don't know what is the output. For trust purpose, in this method we uses Naive Bayes trust model. This model is composed of three factors: 1) The cloud provider which performs the service requests. 2)The system doesn't know what is the output. For trust purposes, in this method, we use the Naive Bayes trust model. This model is composed of three factors: 1) The cloud provider which performs the service requests. 2)The cloud user which makes the service requests. 3) The trust manager which helps the cloud users in selecting the most trustworthy providers. In this model, there are two phases that happen within the trust manager namely:

Handling service requests

Trust computing.

We have not considered the situation where the cloud users may give unfair high or low ratings to benet some cloud providers or deceive others. So at this point, this method cannot be used.

**Reinforcement Learning :**The RL method depends on the Reward-based and Feedback oriented. Reward means a machine self-set and feedback oriented means to decide on the environment. It has the ability to learn and re-learn the proposed model. To achieve more efficient detection in the cloud the RL method is used.

## Chapter 2

# Literature Survey

This literature survey shows the lists of various clouds security used in the trust environment. Below description shows the list of different cloud security which is surveyed that are used in monitoring. Various techniques are used for analysis that they are used for prediction. These papers mainly contain the study related to various securities and trust problems in the cloud and their countermeasures to avoid such cases in the cloud.

- H. Wang, C. Yu, L. Wang and Q. Yu, "Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences," in IEEE Transactions on Services Computing, vol. 11, no. 4, pp. 644-657, 1 July-Aug. 2018. [?]

In this paper tackle heterogeneous preference- and trust-based service selection by developing a novel multi-objective optimization approach to make trade-off decision between service's trust value and user's QoS preference to rank candidate Cloud services based on their match degrees with users' requirements.

- D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman and D. Woods, "Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds," in IEEE Transactions on Cloud Computing, vol. 5, no. 3, pp. 523-536, 1 July-Sept. 2017.doi: 10.1109/TCC.2015.2415794[?]

In this paper, present a cloud architecture reference model that incorporates a wide range of security controls and best practices, and a cloud security assessment model—Cloud-Trust—that estimates high-level security metrics to quantify the degree of confidentiality and integrity offered by a CCS or cloud service provider

(CSP). Cloud-Trust is used to assess the security level of four multi-tenant IaaS cloud architectures equipped with alternative cloud security controls.

- Varadharajan and U. Tupakula, "On the Design and Implementation of an Integrated Security Architecture for Cloud with Improved Resilience," in *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 375-389, 1 July-Sept. 2017. [?] In this paper propose an integrated security architecture that combines policy-based access control with intrusion detection techniques and trusted computing technologies for securing distributed applications running on virtualized systems. Our security architecture incorporates access control security policies for secure interactions between applications and virtual machines in different physical virtualized servers. It provides intrusion detection and trusted attestation techniques to detect and counteract dynamic attacks in an efficient manner. We demonstrate how this integrated security architecture is used to secure the life cycle of virtual machines including dynamic hosting and allocation of resources as well as the migration of virtual machines across different physical servers.
- S. Deshpande and R. Ingle, "Trust assessment in cloud environment: Taxonomy and analysis," 2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, 2016, pp. 627-631. [?]

This paper presents the taxonomy of trust models and the classification of information sources for trust assessment in the cloud paradigm. It analyzes further the existing approaches of trust assessment in the cloud environment and portrays the potential for future research. The intent of the paper is also to identify different dimensions that are needed for effective assessment of trust in the cloud environment.

- R. Neisse, D. Holling and A. Pretschner, "Implementing Trust in Cloud Infrastructures," 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Newport Beach, CA, 2011, pp. 524-533. [?]

This paper presents a system that enables periodical and necessity-driven integrity measurements and remote attestations of vital parts of cloud computing infrastructures. Building on the analysis of several relevant attack scenarios, our system

is implemented on top of the Xen Cloud Platform and make use of trusted computing technology to provide security guarantees. We evaluate both the security and performance of this system. We show how our system attests the integrity of cloud infrastructure and detect all changes performed by system administrators in a typical software configuration, even in the presence of a simulated denial-of-service attack.

- M. Fugini and G. Hadjichristofi, "Security and trust in Cloud scenarios," 2011 1st International Workshop on Securing Services on the Cloud (IWSSC), Milan, 2011, pp. 22-29.[?]

In this paper, consider two real-life scenarios; 1) risk management in work areas, and 2) the execution of scientific experiments in cooperation among various computations nodes. Also, investigate how we can leverage Cloud capabilities and extend the aforementioned scenarios to the Cloud.

## 2.1 Pros and Cons Of cloud trust model

There are various parameters for measuring the pros and cons of the system. The following table describes the parameter which will more discuss the things which make the security of the cloud more desirable and in the same table we are discussing the parameter which makes us concerned about the security of the cloud.

Paper	Advantage	Disadvantage
Trust Assessment in Cloud Environment: Taxonomy and Analysis [?]	effective assessment of trust in cloud environment.	multi-cloud environment demands context-aware trust computation. Mutual trust assessments of service provider and consumer also need to be addressed.
Trust Assurance in Cloud Services with the Cloud Broker Architecture for Dependability. [?]	improvements of DBA, CloudSim toolkit.delivering services to the cloud consumer compliantly to its requirements.	OpenStack will enable us to benet from the cloud own concepts and paradigms at the IaaS service level.

<b>Paper</b>	<b>Advantage</b>	<b>Disadvantage</b>
Trust in cloud computing[?]	Easily determine the most effective path towards earning trust. We analyse that, it useful for cloud service providers to assess their approach towards customers and make change.	Improve security of consumers.
Achieving Trust in Cloud Computing Using Secure Data Provenance[?]	protect the privacy and confidentiality of the user and avoid any unfair information or abnormal behaviour.	Formalize the trust model which will be more actual terms to prove its security features.
Establishing Trust in Cloud Computing Security with the Help of Inter-Clouds[?]	improve the acceptance of trust issue in end users mind.	malicious attack is still a major problem.
Towards a process-oriented framework for improving trust and security in migration to cloud[?]	clearly defined repeatable processes in the cloud	different parameters and internal. environmental criteria could affect the results.
SVM- A novel trust measurement system in cloud service SVM=Security and Vulnerability Matrix [?]	virtualization potential, high storage capacity, multi tenancy , high service availability.	mathematical model which is based on simulation is still in the progress.

<b>Paper</b>	<b>Advantage</b>	<b>Disadvantage</b>
Trusted platform modules in cyber-physical systems: On the interference between security and dependability [?]	Provide the security and increase the system reliability.	development of more robust root of trust implementations. High security needs of future CPS

Table 2.1: A list of Survey Papers on Cloud Trust Model.

## 2.2 Work done by the researcher's in this domain

Name Of Researcher's	Methods	Parameters
Ali Sunyaev	Architecture of continuous auditing (CA) and continuous auditing methodology	Reliability, security
Sebastian Lins	Log Inspection, Data Integrity Validation	High level of security and reliability
Rajesh Ingle	Trust Assessment Techniques	Adaptability, Credibility, Trust Dynamics
Ricardo Neisse	BonaFides system,	Confidentiality, integrity
Mariagrazia Fugini	Risk measurement services, DL Services, Protection services	risk management, security

Table 2.2: A list of researcher's who has found trust in cloud .

## 2.3 Cloud trust Parameters

Over the period cloud has noted various parameters on each level of the cloud. In this section, we are describing some parameters that are placed over the period on the cloud.

- Availability:

Availability means system or data is accessible when required to use. In availability, there are two terms mainly: MTBR-mean time between failure and MTTR-mean time to repair. When the resource is too busy and resource is shut downed then cloud resources said to be unavailable.

- Reliability :

Reliability means the success rate. For trusted cloud. The reliability is the most important component. Reliability means the ability of a system to perform it's required functions under defined rules in a timely manner. When cloud resources accept the data or job then cloud identifies how much data is reliable and those data complete the job or not. We can measure the reliability by successful completion of accepted jobs by the cloud resource.

- Data integrity:

Verify that received data are exactly the same data or not. It provides perfect data. Any corrupted or deleted data can be timely identify and this is the major point for data recovery. Data integrity is a big and efficient term, and it includes, security, accuracy, data safety.

- Resource management :

Resource management is based on virtualization and distributed nature. Virtualization provides flexible and on-demand resources. When the task is complete all the resources become released. Resource management provides the best performance, and it also used in hardware. In the cloud environment, there are various resources, these resources are virtualized and share it to multiple clients.

- Risk management

In risk management, there are three terms, risk assessment, risk control, and risk treatment. The risk also includes security risks and issues. Manage or reduce the risk and decrease the workload.

- Protecting communication:

The communication is based on encryption protocols like Secure Socket Layer and Transport Layer Security. With the help of these protocols, it provides the basics of confidentiality and integrity. These protocols also provide security in the communication environment.

- Hardware security:

Hardware security means a physical device. Hardware security is also important for monitoring network traffic and scans the system. Hardware security provides powerful security as compared to software. The protection of the hardware system is also known as hardware security. For hardware security, it is necessary to consider the vulnerability of the existing system and provide security against it.

- Secure architecture:

If cloud architecture is secure that means the cloud is also secure. The cloud platform provides some security capabilities like IaaS and PaaS, so for secure architecture, the necessary step is to understand these capabilities.

# Chapter 3

## Problem Statement

In cloud computing user's data stored on remote servers which may be operating by others and can be accessed through the internet connection. The facilities provided by the cloud is too attractive for customers but it has distributed and non-transparent nature due to some obstacles using in cloud computing service because users lose their control over the data, and they are sure about whether cloud provider trust or not. So the user's confused with cloud providers regards trust issues.

When downtime occurs due to power loss or network connectivity issues. At this stage, the cloud trust model does not work properly.

### 3.1 Issues needs to address

- The availability, data storage, CPU, network efficiency are related to QoS parameters. Due to loss of power the data are not available on the cloud.
- Cloud provides some legal agreement with infrastructure service providers to guarantee a minimum QoS. This legal agreement is called the Service Level Agreement. If SLA hardware availability is not Uptime then it does not work properly.
- The trust is based on data security, if due to some network issues the data may be lost.
- Without proper or protect communication clients not get secure or trusted data.
- Due to high risk or vulnerability, the cloud does not provide sufficient services.

- If communication is not protected then un-authorization person gets the access and use confidential information.
- Without protecting communication the data on cloud may be lost.

For business relationships. The cloud makes use of service level agreements. The service level agreement provides a framework for both sellers and buyers also provides a profit to both. In cloud there are two types of service level agreement: Infrastructure SLA and Application SLA.

SLA has some key elements like hardware availability, power availability, data center network availability, backbone network availability, outage notification guarantee, internal latency guarantee, etc. These elements are most important for SLA. These all are elements make the trusted cloud. If these elements are not available when the process is going on then trust cannot be built.

## 3.2 Problem in the trusted cloud

Growing security risk in the last few years in various components of the cloud is a major concern. In cloud; trust is the major concern as a security point of view. In public cloud computing user's data stored on the public cloud which may be operating by others and can be accessed through the internet connection. The facilities provided by the cloud are too attractive and for customers but it has distributed and non-transparent nature due to some obstacles using in cloud computing service because users lose their control over the data, and they are sure about whether cloud provider trust or not. So customers confuse with cloud providers regards trust issues.

When cloud downtime happens the trust cannot be built. The reason for downtime is a power loss and network connectivity issues. Due to a network connectivity problem, data on the cloud may be lost. Another issue is an unprotected communication, due to unprotected communication. The sender may not send to the authenticated person or it may be lost. The trusted cloud depends on many parameters like confidentiality, availability, risk management, resource management, secure communication, and secure architecture, etc. These parameters and their value are the most important for the cloud equation. And also SLA has some key elements like hardware availability, power availability, data center network availability, backbone network availability, outage notification

guarantee, internal latency guarantee, etc. These elements are most important for SLA. These elements make the trusted cloud. If these elements are not available when the process is going on then trust cannot be built. Cloud provides some legal agreement with infrastructure service providers to guarantee a minimum QoS. This legal agreement is called the Service Level Agreement. If SLA hardware availability is not Uptime then it does not work properly.

# Chapter 4

## Proposed Solution

System Architecture of any framework describes the overall flow of the system in which manner it will work and flow to perform the given task. In this section of cloud system architecture, we have described the flow of cloud while providing various services to the users. Fig. 4.1 Cloud flow describes the flow of cloud toward trust parameters.

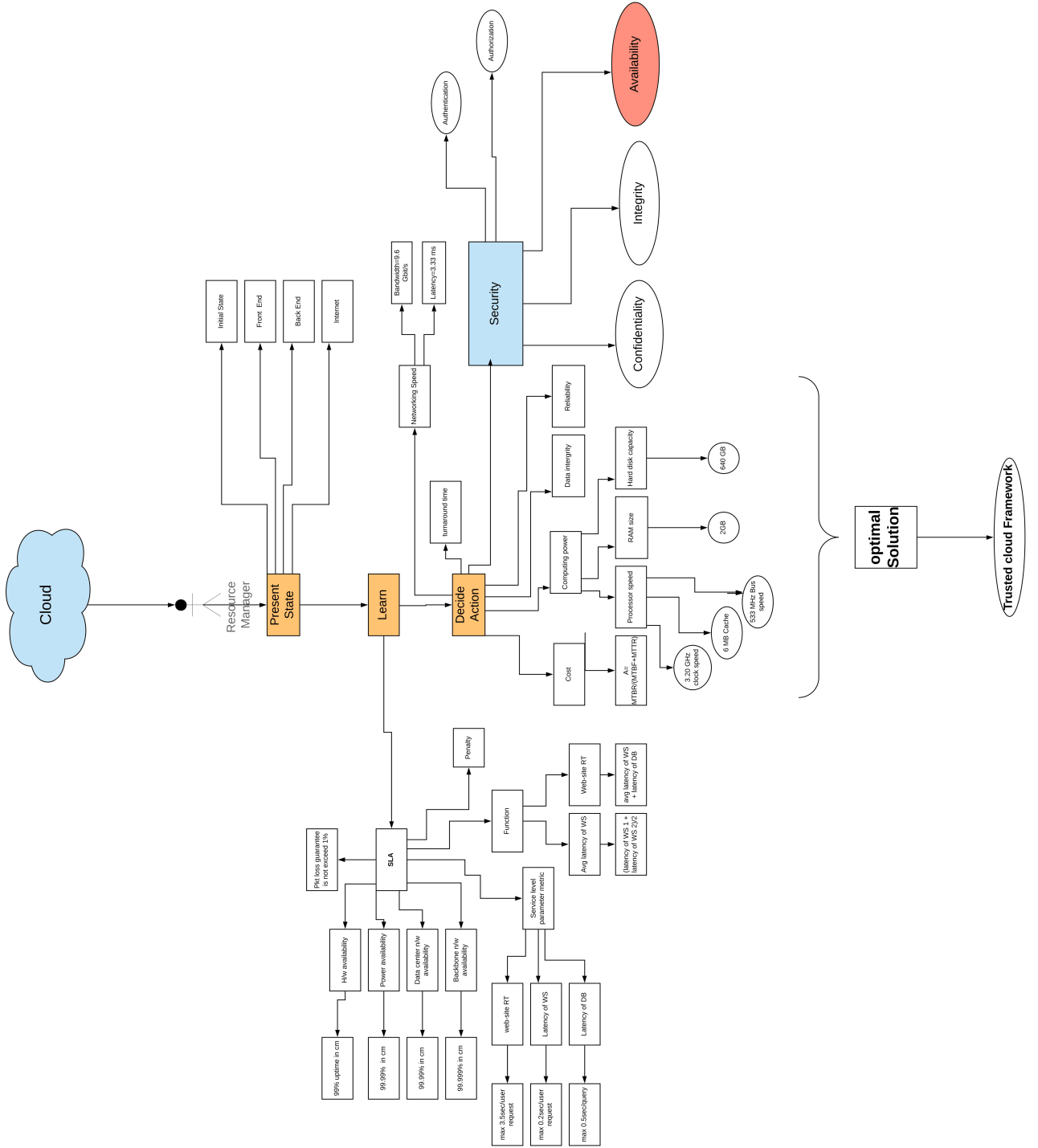


Figure 4.1: Model diagram

Fig. 4.2 mainly describes the security of the trusted cloud. There are mainly three elements; authentication, authorization, and data integrity. Authentication means every user has its own user id and password so when users want to communicate with CSP, this information is required for login purposes. Authorization means a person who has no authority to access the data of cloud then it can't be accessed, only authorization person can access the data from the cloud.

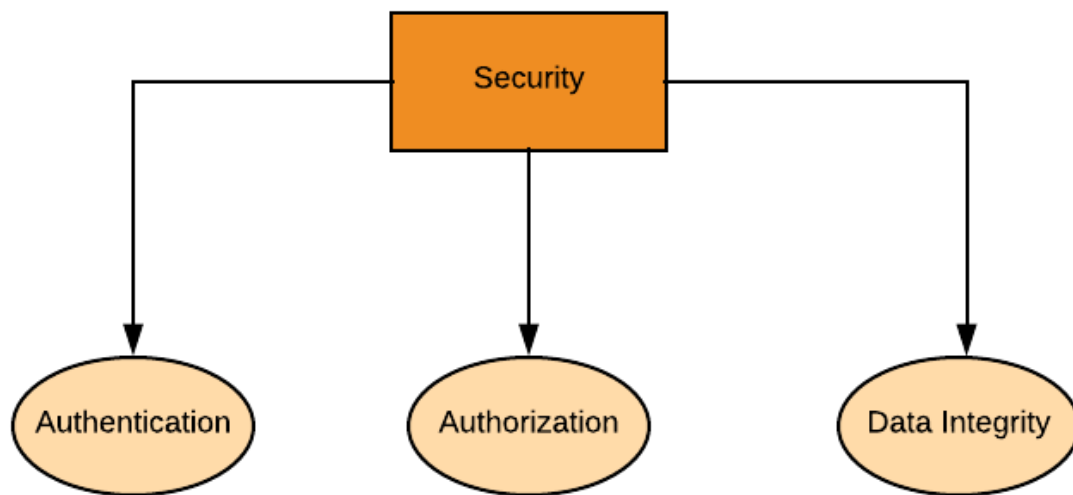


Figure 4.2: Security parameter

In a controlled cloud. The data integrity is the most effective parameter. It verifies that received data are exactly the same data or not. It provides perfect data. Any corrupted or deleted data can be timely identify and this is the major point for data recovery.

Fig.4.3 contains the information based on the SLA. The SLA means service level agreement. SLA has some key elements like hardware availability, power availability, data center network availability, backbone network availability, outage notification guarantee, internal latency guarantee, etc. These elements are most important for SLA. These all are elements make the trusted cloud. If these elements are not available when the process is going on then trust cannot be built.

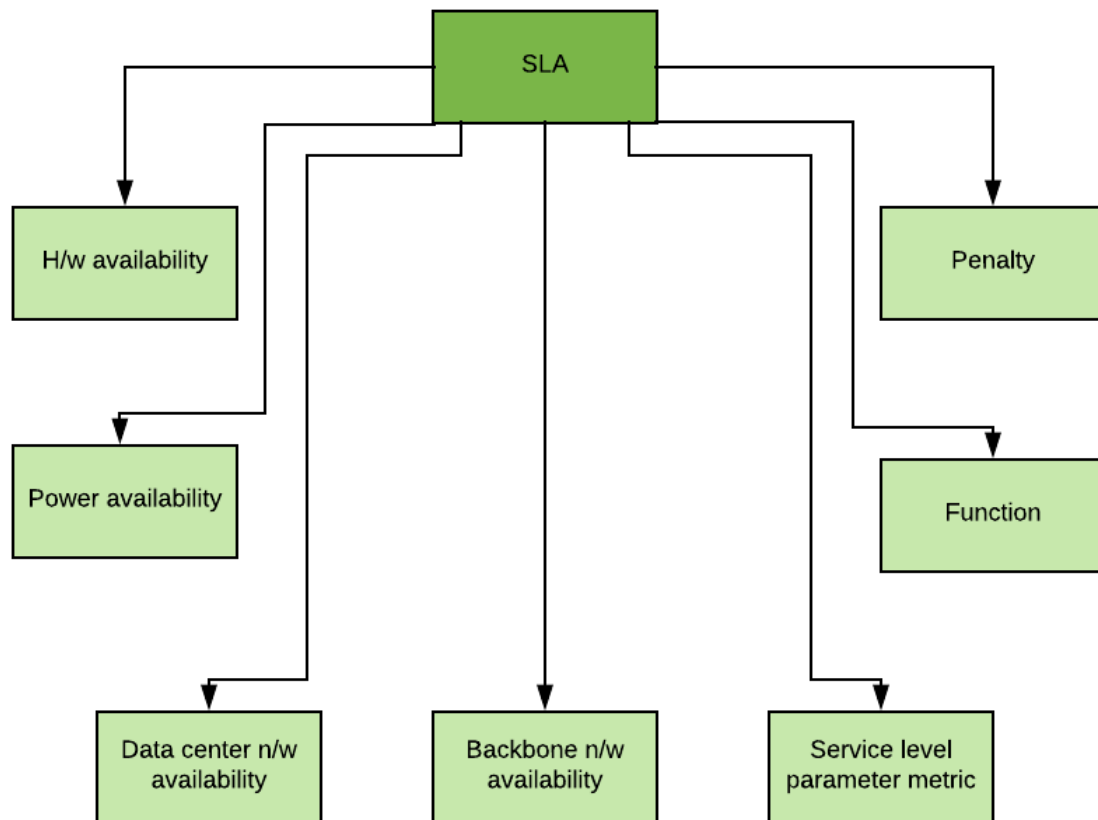


Figure 4.3: Parameters of SLA

# Chapter 5

## Implementation and Results

This section mainly focus on algorithm. Mainly it cover the cloud trust parameters.

---

**Algorithm 1:** Algorithm for Trust

---

- 1 Set a cloud environment
  - 2 Let performance parameters be CPU Cycle Usage be P1,Memory Usage be P2,  
RAM Usage be P3,..etc.
  - 3 Let threshold of any performance parameter for t1.
  - 4 Let threshold of any performance parameter for t2.
  - 5 Let P1 or P2 or P3 of new job= P.
  - 6 **If**  $P > t1$  or  $P \leq t2$  **then**.  
    Start monitoring process.  
    **end if**
  - 7 Calculate AV, R and DI
$$AV = \frac{MTBR}{MTBF+MTTR}$$
$$R = \frac{C}{A}$$
$$DI = \frac{D}{C}$$
  - 8 Calculate Trust.
  - 9  $T=AV+R+DI$ .
- 

Availability, data integrity and reliability are the most important parameter as cloud trust perspective.

## 5.1 Experiments on public cloud (AWS)

AWS is the cloud services of Amazon, which provides the various services to the client and the client uses these services according to their needs. Public cloud AWS provides the various domain which is widely used are: compute, storage, database, migration, network, and content delivery, management tools, security, and identity compliance, etc. Based on our parameter Availability, in aws autoscaling and load balancing is used for achieving high availability. In this process:

1. Launch the instances
2. Instance description

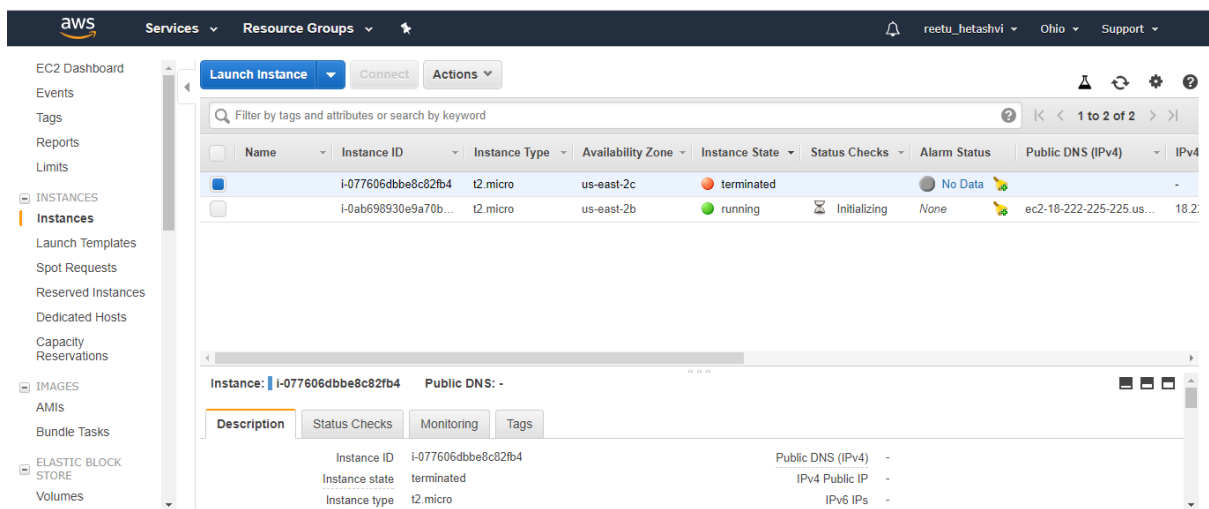


Figure 5.1: Launch the instance

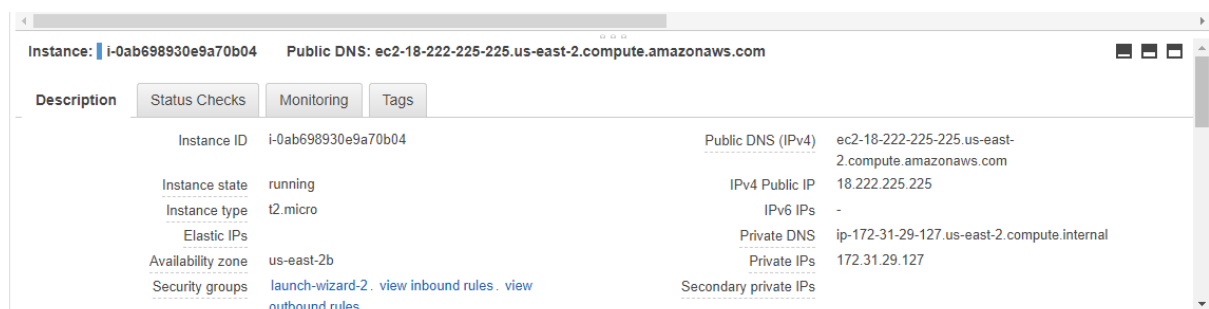
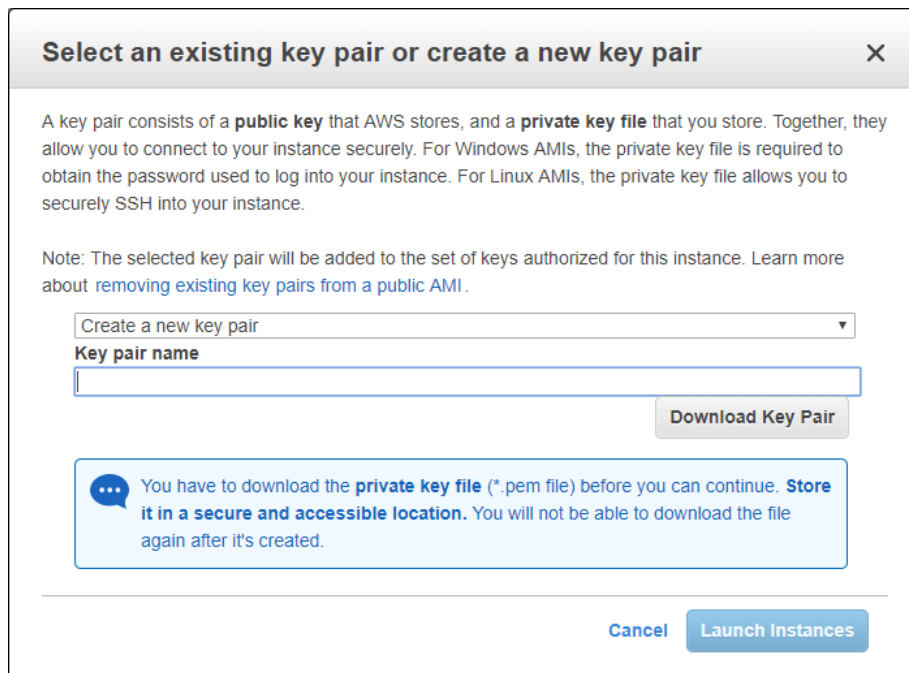


Figure 5.2: Instance description

3. Create a new key pair and download in .pem file.



**Select an existing key pair or create a new key pair** [X]

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair [v]

**Key pair name**

[Text input field]

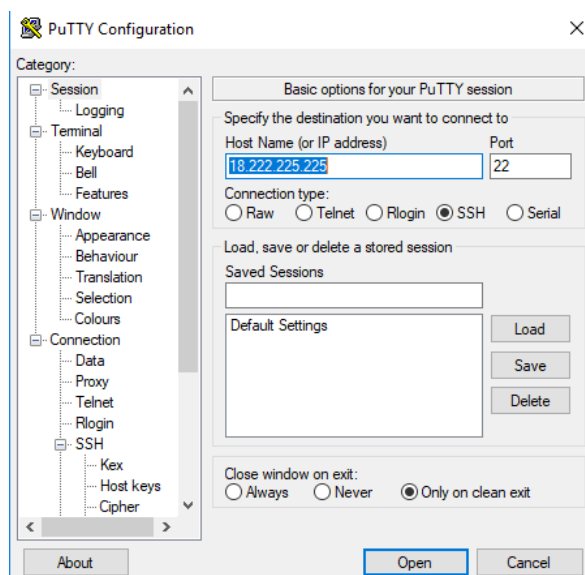
Download Key Pair

**You have to download the private key file (\*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.**

Cancel Launch Instances

Figure 5.3: Download key-pair

4. Convert .pem file into .ppk file in puttygen. After that copy public IP into putty and open.



**PuTTY Configuration** [X]

Category:

- Session
- Logging
- Terminal
- Keyboard
- Bell
- Features
- Window
- Appearance
- Behaviour
- Translation
- Selection
- Colours
- Connection
- Data
- Proxy
- Telnet
- Rlogin
- SSH
- Key
- Host keys
- Cipher

**Basic options for your PuTTY session**

Specify the destination you want to connect to

Host Name (or IP address) Port

18.222.225.225 22

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

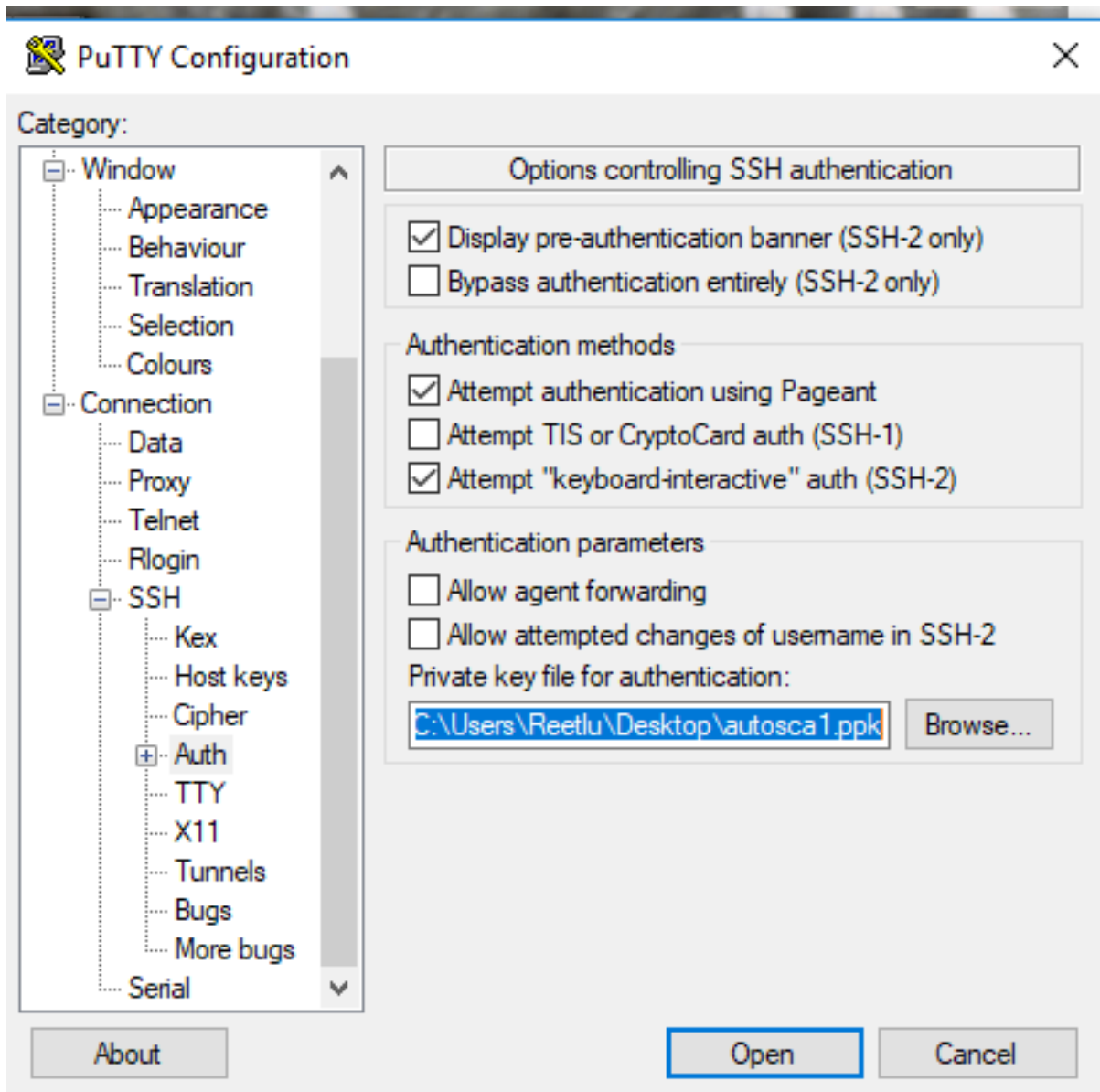
Load Save Delete

Close window on exit:

☐ Always ☐ Never ☒ Only on clean exit

About Open Cancel

Figure 5.4: Open the putty configuration



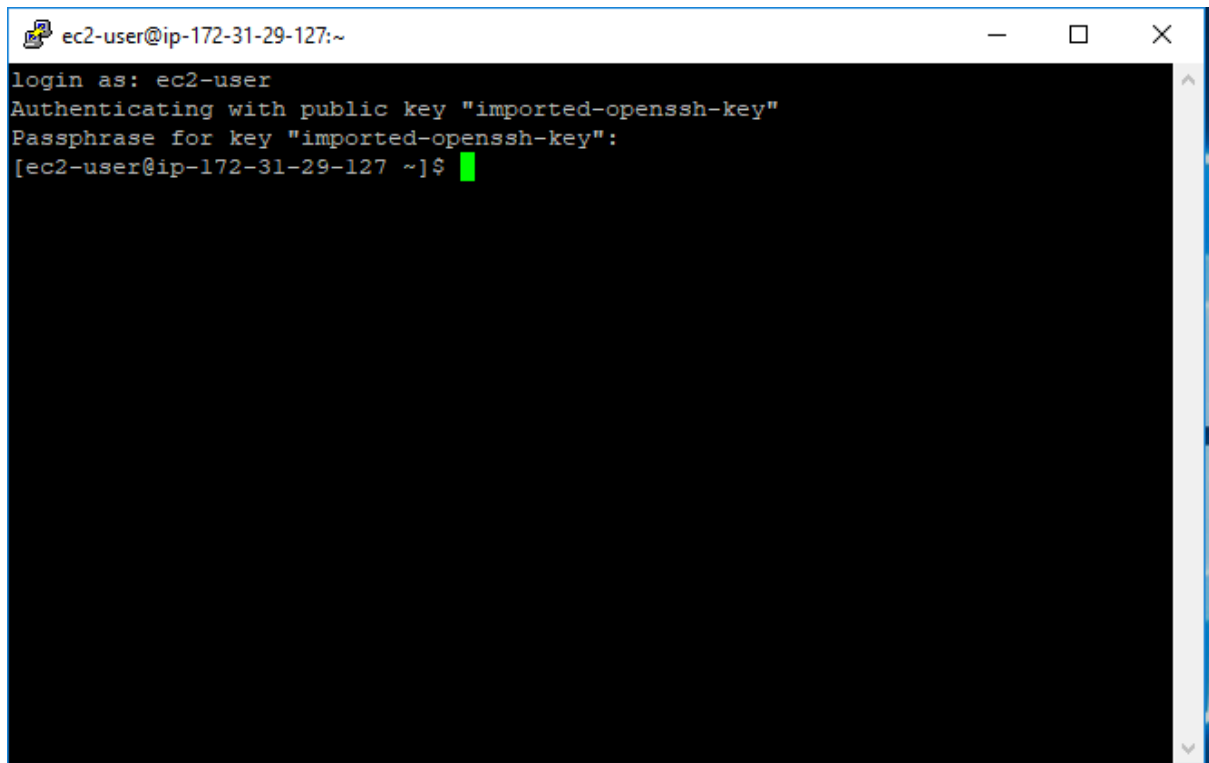


Figure 5.5: Open the terminal

5. Right click on running instance and select on create image (AMIs).

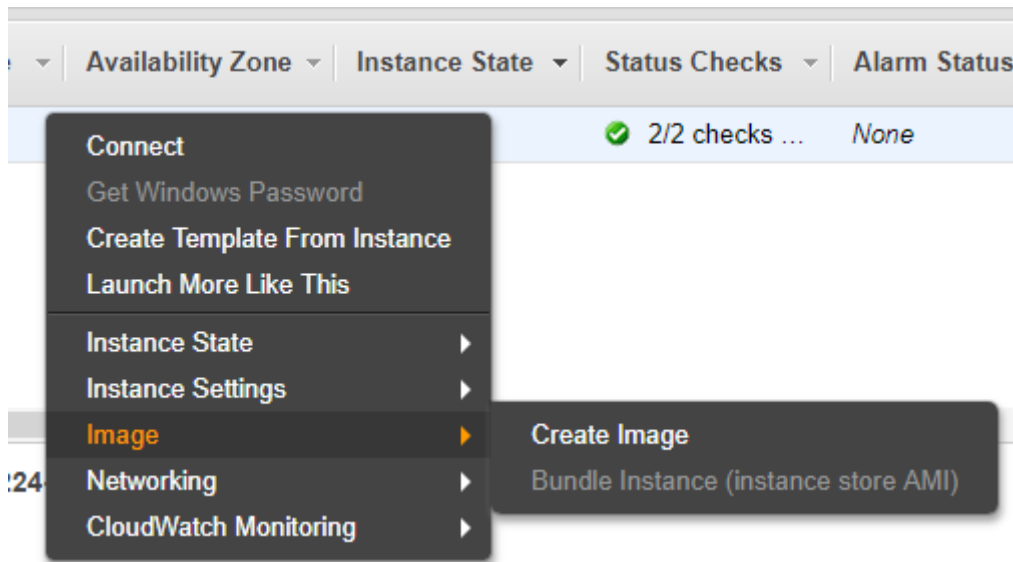


Figure 5.6: Create an image

Create image name and image description

The 'Create Image' dialog box is shown. It contains the following fields and options:

- Instance ID**: f-049e94edd8800889e
- Image name**: RedHat
- Image description**: WebServer
- No reboot**: ☐

**Instance Volumes**

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-02831d0ed9ced7c0c	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

**Add New Volume**

Total size of EBS Volumes: 8 GiB  
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

**Cancel** **Create Image**

Figure 5.7: Create an image

In auto scaling there are two phases:

1. Launch Configuration
2. Auto Scaling Group

Launch configuration;

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Capacity

Reservations

Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).

Create launch configuration

Create Auto Scaling group

Copy to launch template

Actions

Filter:

<<

<

1 to 1 of 1 Launch Configurations

>

>>

	Name	AMI ID	Instance Type	Spot Price	Creation Time
<input checked="" type="checkbox"/>	lpw	ami-976152f2	t2.small		April 27, 2018 at 11:06:06 AM U...

1. Choose AMI

2. Choose Instance Type

3. Configure details

4. Add Storage

5. Configure Security Group

6. Review

Create Launch Configuration

Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Ownership

1 to 1 of 1 AMIs

<<

<

>

>>

RedHat - ami-0add30a975378ff11

WebServer

Root device type: ebs   Virtualization type: hvm   Owner: 547838237415

Select

64-bit

Create Launch Configuration

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

Create a new security group

Select an existing security group

Security Group ID	Name	VPC ID	Description	Actions
<input type="checkbox"/> sg-d02234bb	AutoScaling-Security-Group-1	vpc-ceb76ba6	AutoScaling-Security-Group-1 (2018-04-27 11:05:11.680+05:30)	<a href="#">Copy to new</a>
<input type="checkbox"/> sg-865aa1ed	default	vpc-ceb76ba6	default VPC security group	<a href="#">Copy to new</a>
<input type="checkbox"/> sg-2b58a340	launch-wizard-1	vpc-ceb76ba6	launch-wizard-1 created 2018-01-22T19:28:28.106+05:30	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/> sg-5734223c	launch-wizard-2	vpc-ceb76ba6	launch-wizard-2 created 2018-04-27T10:17:56.967+05:30	<a href="#">Copy to new</a>

Inbound rules for sg-d02234bb Selected security groups: sg-5734223c.

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0

Cancel

Previous

Review

27

## Create Launch Configuration

Review the details of your launch configuration. You can go back to edit the details of each section before you finish.

**⚠ Improve security of instances launched using your launch configuration, web. Your security group, launch-wizard-2, is open to the world.**  
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

### AMI Details

[Edit AMI](#)

RedHat - ami-0add30a975378ff11

WebServer

Root device type: ebs    Virtualization Type: hvm

### Instance Type

[Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory GiB	Instance Storage (GiB) GiB	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

## Auto Scaling Group

First terminate all the instance then create Auto Scaling Group.

### Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).

Create Auto Scaling group

Actions



Filter:

1 to 4 of 4 Auto Scaling Groups

<input type="checkbox"/>	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
<input checked="" type="checkbox"/>	web	webserver	1	2	2	2	us-east-2a, us-east-2b, us-e...	300	300

## Create Auto Scaling Group

[Cancel and Exit](#)

You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)

Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned.

[Learn more](#)

[Create new launch template](#)

- ☐ Create a new launch configuration
- ☒ Use an existing launch configuration

Filter launch configurations...

1 to 4 of 4 Launch Configurations

Name	AMI ID	Instance Type	Spot Price	Security Groups
<input type="checkbox"/> serversec	ami-0d3c793a5cbf074e5	t2.micro		sg-5734223c
<input type="checkbox"/> security	ami-0bb75a49cd70b9a63	t2.micro		sg-5734223c
<input checked="" type="checkbox"/> webserver	ami-0add30a975378ff11	t2.micro		sg-5734223c
<input type="checkbox"/> lpw	ami-976152f2	t2.small		sg-d02234bb

Cancel [Next Step](#)

We can start with 2 or more instances.

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

### Create Auto Scaling Group

[Cancel and Exit](#)

**Group name** ⓘ asg

**Launch Configuration** ⓘ webserver

---

**Group size** ⓘ Start with  instances

---

**Network** ⓘ vpc-ceb76ba6 (172.31.0.0/16) (default) [Create new VPC](#)

**Subnet** ⓘ 

subnet-fca1bd87(172.31.16.0/20) | Default in us-east-2b x  
|  
subnet-0152024c(172.31.32.0/20) | Default in us-east-2c  
subnet-10e34678(172.31.0.0/20) | Default in us-east-2a

[Create new subnet](#) a public IP address. ⓘ

► Advanced Details

Select the classic load balancers.

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

### Create Auto Scaling Group

[Cancel and Exit](#)

▼ Advanced Details

**Load Balancing** ⓘ ☒ Receive traffic from one or more load balancers [Learn about Elastic Load Balancing](#)

**Classic Load Balancers** ⓘ

**Target Groups** ⓘ

---

**Health Check Type** ⓘ ☐ ELB ☒ EC2

**Health Check Grace Period** ⓘ  seconds

**Monitoring** ⓘ Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration webserver. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency. [Learn more](#)

**Instance Protection** ⓘ

**Service-Linked Role** ⓘ AWSServiceRoleForAutoScaling [View Role in IAM](#)

[Cancel](#) [Next: Configure scaling policies](#)

## Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

### Auto Scaling Group Details

Group name	asg
Group size	2
Minimum Group Size	2
Maximum Group Size	2
Subnet(s)	subnet-fca1bd87, subnet-0152024c, subnet-10e34678
Load Balancers	webserverelb
Target Groups	
Health Check Type	EC2
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	Protect From Scale In
Service-Linked Role	AWSServiceRoleForAutoScaling

### Scaling Policies

### Notifications

[Cancel](#) [Previous](#) [Create Auto Scaling group](#)

Launch Instance	Connect	Actions								
Filter by tags and attributes or search by keyword										
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4		
	i-049e94edd880088...	t2.micro	us-east-2b	stopped		None		-		
	i-0da32bd7b44a83fcc	t2.micro	us-east-2c	stopped		None		-		
	i-0eb7ffabf8af7270f	t2.micro	us-east-2c	stopped		None		-		
	i-074c146e5439f56a5	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-18-220-180-24.us-...	18.2.		
	i-0a06cd6d60534cca0	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-18-222-164-186.us...	18.2.		

Description	Status Checks	Monitoring	Tags
Instance ID	i-074c146e5439f56a5	Public DNS (IPv4)	ec2-18-220-180-24.us-east-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	18.220.180.24
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-24-48.us-east-2.compute.internal
Availability zone	us-east-2b	Private IPs	172.31.24.48
Security groups	<a href="#">launch-wizard-2</a> , <a href="#">view inbound rules</a> , <a href="#">view outbound rules</a>	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-ceb76ba6
AMI ID	prc (ami-0d3c793a5cbf074e5)	Subnet ID	subnet-fca1bd87
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	autoscaling	T2/T3 Unlimited	Disabled
Owner	547838237415	EBS-optimized	False

## Benefits of auto scaling

- Setup scaling quickly
- Automatically maintain performance
- Pay only for what you need

Maintain optimal application performance and availability, even when workloads are periodic, unpredictable, or continuously changing.

## 5.2 Install jupyter in AWS

The process of installing jupyter in AWS is similar to auto scaling group till putty. After that open the terminal and type `sudo su` for root access. After that run all command which is shown below:

```
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-21-221 ~]$ sudo su
[root@ip-172-31-21-221 ec2-user]# jupyter notebook --generate-config --allow-root
Overwrite /root/.jupyter/jupyter_notebook_config.py with default config? [y/N]y
Writing default config to: /root/.jupyter/jupyter_notebook_config.py
[root@ip-172-31-21-221 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-21-221 ec2-user]# ls
Anaconda3-4.4.0-Linux-x86_64.sh  certs
[root@ip-172-31-21-221 ec2-user]# mkdir certs
mkdir: cannot create directory 'certs': File exists
[root@ip-172-31-21-221 ec2-user]# ls -lrt
total 511024
-rw-r--r-- 1 root root 523283080 May 30 2017 Anaconda3-4.4.0-Linux-x86_64.sh
drwxr-xr-x 2 root root 4096 Apr 15 06:10 certs
[root@ip-172-31-21-221 ec2-user]# cd certs/
[root@ip-172-31-21-221 certs]# sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout jupyterpython1.pem -out jupyterpython1.pem
Generating a 1024 bit RSA private key
.....+++++
...+++++
writing new private key to 'jupyterpython1.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:india
string is too long, it needs to be less than 2 bytes long
Country Name (2 letter code) [XX]:in
State or Province Name (full name) []:guj
Locality Name (eg, city) [Default City]:vado
Organization Name (eg, company) [Default Company Ltd]:nu
Organizational Unit Name (eg, section) []:ce
Common Name (eg, your name or your server's hostname) []:sr
Email Address []:abc@gmail.com
[root@ip-172-31-21-221 certs]# sudo su
[root@ip-172-31-21-221 certs]# exit
```

Figure 5.8: install jupyter

```
root@ip-172-31-21-221:/home/ec2-user
glibc.x86_64 0:2.17-260.175.amzn1
glibc-common.x86_64 0:2.17-260.175.amzn1
krb5-libs.x86_64 0:1.15.1-34.44.amzn1
libcurl.x86_64 0:7.61.1-7.91.amzn1
openssl.x86_64 1:1.0.2k-16.150.amzn1
perl-Getopt-Long.noarch 0:2.40-3.6.amzn1
python27.x86_64 0:2.7.16-1.125.amzn1
python27-botocore.noarch 0:1.12.92-2.69.amzn1
python27-devel.x86_64 0:2.7.16-1.125.amzn1
python27-libs.x86_64 0:2.7.16-1.125.amzn1
python27-urllib3.Noarch 0:1.24.1-1.6.amzn1

Complete!
[root@ip-172-31-21-221 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-21-221 ec2-user]#
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Apr 15 04:49:37 2019 from 42.106.27.80

 _ _ _ _ _
| | | | |
|_|_|_|_|_| Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-21-221 ~]$ sudo su
[root@ip-172-31-21-221 ec2-user]# wget https://repo.continuum.io/archive/Anaconda3-4.4.0-Linux-x86_64.sh
--2019-04-15 05:10:15-- https://repo.continuum.io/archive/Anaconda3-4.4.0-Linux-x86_64.sh
Resolving repo.continuum.io (repo.continuum.io)... 104.18.200.79, 104.18.201.79, 2606:4700::6812:c94f, ...
Connecting to repo.continuum.io (repo.continuum.io)|104.18.200.79|:443... connect: Connection refused
HTTP request sent, awaiting response... 200 OK
Length: 523283080 (499M) [application/x-sh]
Saving to: 'Anaconda3-4.4.0-Linux-x86_64.sh'

Anaconda3-4.4.0-Lin 100%[=====>] 499.04M 66.9MB/s in 6.8s

2019-04-15 05:10:22 (73.2 MB/s) - 'Anaconda3-4.4.0-Linux-x86_64.sh' saved [523283080/523283080]

[root@ip-172-31-21-221 ec2-user]#
```

```
[root@ip-172-31-21-221 ec2-user]# ls -lrt
total 0
```

```
[root@ip-172-31-21-221 ec2-user]# yum update all
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main | 2.1 kB 00:00
amzn-updates | 2.5 kB 00:00
No Match for argument: all
No package all available.
No packages marked for update
[root@ip-172-31-21-221 ec2-user]# yum update -y
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
--> Package amazon-ssm-agent.x86_64 0:2.3.68.0-1.amzn1 will be updated
--> Package amazon-ssm-agent.x86_64 0:2.3.274.0-1.amzn1 will be an update
--> Package aws-cfn-bootstrap.noarch 0:1.4-30.21.amzn1 will be updated
--> Package aws-cfn-bootstrap.noarch 0:1.4-31.22.amzn1 will be an update
--> Package aws-cli.noarch 0:1.15.83-1.49.amzn1 will be updated
--> Package aws-cli.noarch 0:1.16.102-1.50.amzn1 will be an update
--> Package bind-libs.x86_64 32:9.8.2-0.68.rc1.58.amzn1 will be updated
--> Package bind-libs.x86_64 32:9.8.2-0.68.rc1.59.amzn1 will be an update
--> Package bind-utils.x86_64 32:9.8.2-0.68.rc1.58.amzn1 will be updated
--> Package bind-utils.x86_64 32:9.8.2-0.68.rc1.59.amzn1 will be an update
--> Package ca-certificates.noarch 0:2017.2.14-65.0.1.17.amzn1 will be updated
--> Package ca-certificates.noarch 0:2018.2.22-65.1.20.amzn1 will be an update
--> Package curl.x86_64 0:7.53.1-16.84.amzn1 will be updated
--> Package curl.x86_64 0:7.61.1-7.91.amzn1 will be an update
--> Package file.x86_64 0:5.30-11.34.amzn1 will be updated
--> Package file.x86_64 0:5.34-3.37.amzn1 will be an update
--> Package file-libs.x86_64 0:5.30-11.34.amzn1 will be updated
--> Package file-libs.x86_64 0:5.34-3.37.amzn1 will be an update
```

```
[root@ip-172-31-21-221 ec2-user]# ls -lrt
total 511020
-rw-r--r-- 1 root root 523283080 May 30 2017 Anaconda3-4.4.0-Linux-x86_64.sh
[root@ip-172-31-21-221 ec2-user]#
```

```
[root@ip-172-31-21-221 ec2-user]# bash Anaconda3-4.4.0-Linux-x86_64.sh

Welcome to Anaconda3 4.4.0 (by Continuum Analytics, Inc.)

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
=====
Anaconda End User License Agreement
=====

Copyright 2017, Continuum Analytics, Inc.

All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.
* Neither the name of Continuum Analytics, Inc. ("Continuum") nor the names
of its contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
--More--
```

```
[root@ip-172-31-21-221 ec2-user]# which python
/usr/bin/python
[root@ip-172-31-21-221 ec2-user]# source .bashrc
[root@ip-172-31-21-221 ec2-user]#
```

root@ip-172-31-21-221:~

login as: ec2-user

Authenticating with public key "imported-openssh-key"

Last login: Mon Apr 15 05:07:04 2019 from 42.106.27.80

```
  _|  _|_ )
 _| (  /   Amazon Linux AMI
__|\___|___|
```

<https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/>

[ec2-user@ip-172-31-21-221 ~]\$ sudo su

[root@ip-172-31-21-221 ec2-user]# cd ~

[root@ip-172-31-21-221 ~]# ls -al

total 40

dr-xr-x--- 4 root root 4096 Apr 15 05:20 .

dr-xr-xr-x 25 root root 4096 Apr 15 04:42 ..

drwxr-xr-x 20 root root 4096 Apr 15 05:19 **anaconda3**

-rw-r--r-- 1 root root 18 Jan 15 2011 .bash\_logout

-rw-r--r-- 1 root root 176 Jan 15 2011 .bash\_profile

-rw-r--r-- 1 root root 254 Apr 15 05:20 .bashrc

-rw-r--r-- 1 root root 176 Apr 15 05:20 .bashrc-anaconda3.bak

-rw-r--r-- 1 root root 100 Jan 15 2011 .cshrc


drwx----- 2 root root 4096 Apr 15 04:42 **.ssh**

-rw-r--r-- 1 root root 129 Jan 15 2011 .tcshrc

[root@ip-172-31-21-221 ~]# vi .bash profile

2 files to edit

**█**

 root@ip-172-31-21-221:/home/ec2-user

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Apr 15 05:29:03 2019 from 42.106.27.80

      _|_  _|_  )
     _|_  ( _|_ /   Amazon Linux AMI
    _|_ \ _|_ \

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-21-221 ~]$ sudo su
[root@ip-172-31-21-221 ec2-user]# which python
/root/anaconda3/bin/python
[root@ip-172-31-21-221 ec2-user]# █
```

```
[root@ip-172-31-21-221 ec2-user]# source .bashrc
[root@ip-172-31-21-221 ec2-user]# which python
/root/anaconda3/bin/python
[root@ip-172-31-21-221 ec2-user]# python
Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:09:58)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

```
[root@ip-172-31-21-221 ec2-user]# ipython
Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:09:58)
Type "copyright", "credits" or "license" for more information.

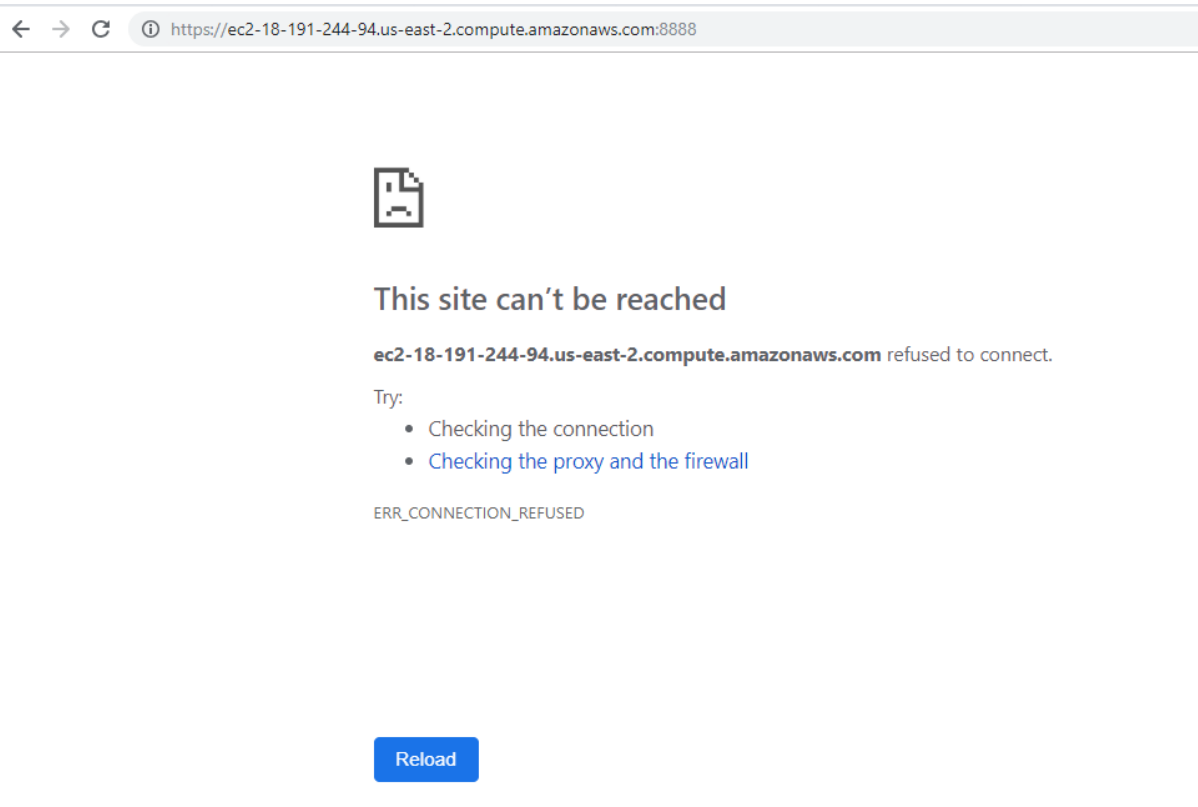
IPython 5.3.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: from IPython.lib import passwd

In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'shal:ab9ffb5fdd47:dd27be688efce95d5c4dec9295c53a4fde705859'

In [3]: █
```

After installing python jupyter, using the out [2] (password) run the command and paste the public DNS of instance in the chrome. But in this, the error occurs, this site can't be reached.



This may be happen because of, AWS doesn't support linux and ubuntu.

## 5.3 Using AmazonSageMaker

Amazon SageMaker has the ability to build, train, and deploy machine learning models quickly. Machine learning offers a variety of benefits for enterprises, such as advanced analytics for customer data or back-end security threat detection. Amazon SageMaker supports Jupyter notebooks, which are open source applications that help developers share live code. For SageMaker users, these notebooks include drivers, packages, and libraries for common deep learning platforms and frameworks. SageMaker can pull data from Amazon Simple Storage Service (S3), and there is no practical limit to the size of the data set.

### How Amazon SageMaker works:

step-1 Open amazon sagemaker console and launches a notebook instance.

step-2 Specifies the location of the data in S3 and the preferred instance type, then initiates the training process.

step-3 When the model is ready to deploy, the service automatically operates and scales cloud infrastructure, using a set of SageMaker instance types.

SageMaker stores code in volumes, which are protected by security groups and offer encryption.

Limitation of Amazon SageMaker.

AWS charges each SageMaker user for the compute, storage and data processing resources used to build, train, perform and log machine learning models and predictions.

## 5.4 Using Linear Regression

Linear regression is Machine Learning algorithm. There are two types: 1) Simple Linear Regression 2) Multiple linear regression In this implementation, we have used Multiple linear regression. Multiple regression means two or more features. Multiple Linear Regression is a simple and common way to analyze linear regression.

In this implementation, we have created one dummy dataset which is used for availability.

Availability depends on the mean-time-between-failure (MTBF) and mean-time-to-repair(MTTR). MTBF means service uptime and MTTR means service downtime.

The formula of Availability is:

$$AV = \frac{MTBR}{MTBF+MTTR}$$

Table 5.1: Dataset

Monitoring hours	Downtime hours	Uptime hours	AV
24	0.79	23.2	0.96
24	1.6	22.3	0.92
24	2.4	21.6	0.9
24	3.19	20.8	0.86
24	4	19.9	0.82
24	4.7	19.2	0.8
24	5.59	18.4	0.76
24	6.4	17.5	0.72
24	7.2	16.8	0.7
24	7.9	16	0.66
24	8.8	15.1	0.62
24	9.6	14.4	0.6
24	10.3	13.6	0.56
24	11.2	12.7	0.52
24	12	12	0.5
24	12.7	11.2	0.46
24	13.6	10.3	0.42
24	14.4	9.6	0.4
24	15.1	8.8	0.36
24	16	7.9	0.32
24	16.8	7.2	0.3
24	17.5	6.4	0.26
24	18.4	5.5	0.22
24	19.2	4.8	0.2
24	19.9	4	0.16
24	20.8	3.1	0.12
24	21.6	2.4	0.1
24	22.3	1.6	0.06
24	23.2	0.79	0.03
24	24	0	0

From this formula, we can say that MTBF+MTTR is equal to the monitoring. So that the availability equation is:

$$AV = \frac{Uptime}{Monitoring}$$

Using this equation we can calculate the availability. So, the coding of linear regression for availability.

Step-1: Import the packages.

Step-2: Check current directory.

Step-3: Set the dataset.csv file directory.

step-4: Read that dataset.csv file.

Step-5: Convert the values into the array.

Step-6: Apply regression algorithm.

step-7: Train the model using linear regression and set the x-label and y-label.

Step-8: The graph represents the availability ratio with respect to monitoring and Uptime. These three dots indicate the achieved availability and only three dots because of test size is equal to 0.3 (we can change the test size).

Step-9: Set the if-else condition for high and low availability. In the output, True means high availability and False means low availability.

```
In [45]: import pandas as pd
import operator
import csv
import matplotlib as mp
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

```
In [46]: pwd
```

```
Out[46]: 'C:\\Users\\Reetlu\\LR for Availability'
```

```
In [48]: r'C:\\Users\\Reetlu\\Documents\\sem-3-4\\Dataset.csv'
```

```
Out[48]: 'C:\\Users\\Reetlu\\Documents\\sem-3-4\\Dataset.csv'
```

```
In [112]: PATH = "C:\\Users\\Reetlu\\Documents\\sem-3-4\\Dataset.csv"
```

```
In [114]: import pandas as pd
df = pd.read_csv("C:\\Users\\Reetlu\\Documents\\sem-3-4\\Dataset.csv")
df.iloc[:25]
```

```
Out[114]:
```

	Monitoring_hrs	Downtime_hrs	Uptime_hrs	AV	High Available
0	24	0.79	23.2	0.96	NaN
1	24	1.60	22.3	0.92	NaN
2	24	2.40	21.6	0.90	NaN
3	24	3.19	20.8	0.86	NaN
4	24	4.00	19.9	0.82	NaN
5	24	4.70	19.2	0.80	NaN
6	24	5.59	18.4	0.76	NaN
7	24	6.40	17.5	0.72	NaN
8	24	7.20	16.8	0.70	NaN
9	24	7.90	16.0	0.66	NaN
10	24	8.80	15.1	0.62	NaN
11	24	9.60	14.4	0.60	NaN
12	24	10.30	13.6	0.56	NaN
13	24	11.20	12.7	0.52	NaN
14	24	12.00	12.0	0.50	NaN

14	24	12.00	12.0	0.50	NaN
15	24	12.70	11.2	0.46	NaN
16	24	13.60	10.3	0.42	NaN
17	24	14.40	9.6	0.40	NaN
18	24	15.10	8.8	0.36	NaN
19	24	16.00	7.9	0.32	NaN
20	24	16.80	7.2	0.30	NaN
21	24	17.50	6.4	0.26	NaN
22	24	18.40	5.5	0.22	NaN
23	24	19.20	4.8	0.20	NaN
24	24	19.90	4.0	0.16	NaN

```
In [115]: X = df.iloc[:, 0].values.reshape(-1, 1) # values converts it into a numpy array
Y = df.iloc[:, 1].values.reshape(-1, 1)
```

```
In [116]: from sklearn import linear_model
reg=LinearRegression()
```

```
In [117]: reg.fit(X, Y)
```

```
Out[117]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

```
In [118]: Y_pred =reg.predict(X)
```

```
In [121]: av=df.AV.loc[:25]  
av
```

```
Out[121]: 0      0.96  
1      0.92  
2      0.90  
3      0.86  
4      0.82  
5      0.80  
6      0.76  
7      0.72  
8      0.70  
9      0.66  
10     0.62  
11     0.60  
12     0.56  
13     0.52  
14     0.50  
15     0.46  
16     0.42  
17     0.40  
18     0.36  
19     0.32  
20     0.30  
21     0.26  
22     0.22  
23     0.20
```

```
In [122]: X = df.iloc[:10][['Monitoring_hrs', 'Uptime_hrs']]
y = df.iloc[:10]['AV']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

from sklearn.linear_model import LinearRegression
lm = LinearRegression()

lm.fit(X_train, y_train)

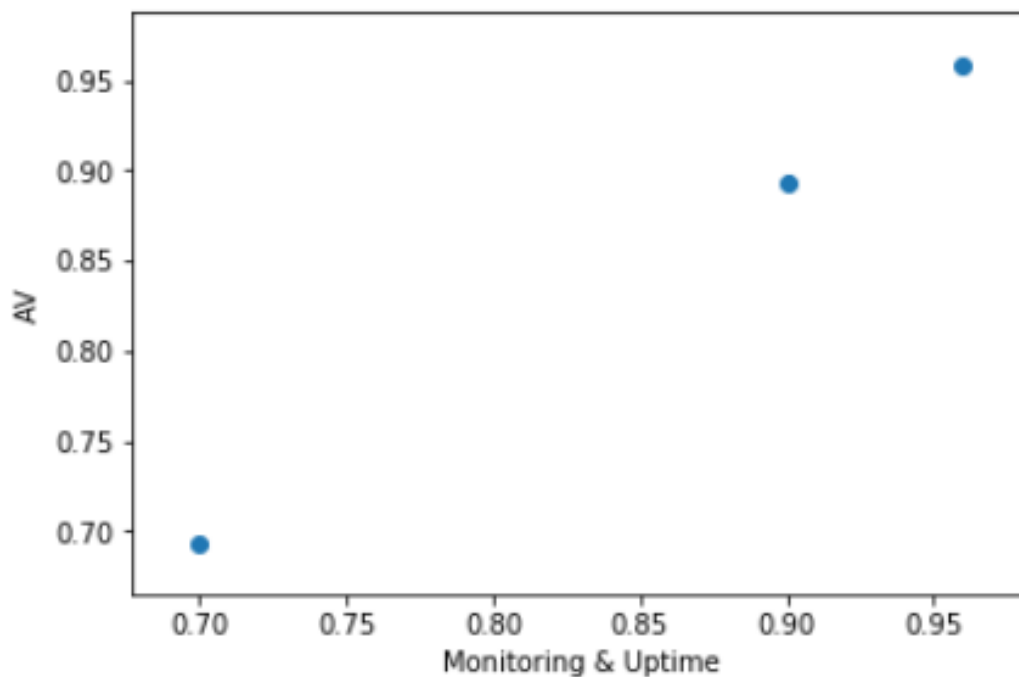
print (lm.coef_)

predictions = lm.predict(X_test)

#Create a scatterplot of the real test values versus the predicted values.
from matplotlib import pyplot as plt
plt.scatter(y_test, predictions)
plt.xlabel('Monitoring & Uptime')
plt.ylabel('AV')
```

```
[0.          0.04146658]
```

```
Out[122]: Text(0,0.5,'AV')
```



```
In [128]: av=df.AV.loc[:25]
print(av)
df['High Available'] = df['AV'].apply(lambda x: 'True' if x >=0.80 else 'False')
#if av >= 0.85:
#    print("high availability")
#else:
#    print("Low availability")
#av.all()
print(df)
```

```
0      0.96
1      0.92
2      0.90
3      0.86
4      0.82
5      0.80
6      0.76
7      0.72
8      0.70
9      0.66
10     0.62
11     0.60
12     0.56
13     0.52
14     0.50
15     0.46
16     0.42
17     0.40
18     0.36
19     0.32
20     0.30
21     0.26
22     0.22
23     0.20
24     0.16
25     0.12
Name: AV, dtype: float64
```

	Monitoring_hrs	Downtime_hrs	Uptime_hrs	AV	High Available
0	24	0.79	23.20	0.96	True
1	24	1.60	22.30	0.92	True
2	24	2.40	21.60	0.90	True
3	24	3.19	20.80	0.86	True
4	24	4.00	19.90	0.82	True
5	24	4.70	19.20	0.80	True
6	24	5.59	18.40	0.76	False
7	24	6.40	17.50	0.72	False
8	24	7.20	16.80	0.70	False
9	24	7.90	16.00	0.66	False
10	24	8.80	15.10	0.62	False
11	24	9.60	14.40	0.60	False
12	24	10.30	13.60	0.56	False
13	24	11.20	12.70	0.52	False
14	24	12.00	12.00	0.50	False
15	24	12.70	11.20	0.46	False
16	24	13.60	10.30	0.42	False
17	24	14.40	9.60	0.40	False
18	24	15.10	8.80	0.36	False
19	24	16.00	7.90	0.32	False
20	24	16.80	7.20	0.30	False
21	24	17.50	6.40	0.26	False
22	24	18.40	5.50	0.22	False
23	24	19.20	4.80	0.20	False
24	24	19.90	4.00	0.16	False
25	24	20.80	3.10	0.12	False
26	24	21.60	2.40	0.10	False
27	24	22.30	1.60	0.06	False
--	--	-- --	-- --	-- --	--

# Chapter 6

## Conclusion & Future Work

### 6.1 Conclusion

- According to the report of the semantic corporation, we can say that the cloud is getting more vulnerable day by day.
- Our evaluation shows the high availability and low availability. So, we can say that, when the availability is high then the trust is achieved. Achieve trust using a supervised learning method (linear regression).Also discuss cloud security and various cloud parameters.

we can calculate availability using MTTR and MTBF.

### 6.2 Future work

- If AmazonSagemaker service is freely available then we can run our algorithm or code using this service. And this service provides the output quickly, and it's easy to develop.
- This research work is on progress. In the future, Reinforcement Learning techniques will use for trust using other parameters.

# References

- [1] Manuel, Paul. (2013). A trust model of cloud computing based on Quality of Service. *Annals of Operations Research*. 233. 1-12. 10.1007/s10479-013-1380-x.
- [2] M. H. Ling and K. A. Yau, "Reinforcement Learning-Based Trust and Reputation Model for Spectrum Leasing in Cognitive Radio Networks," 2013 International Conference on IT Convergence and Security (ICITCS), Macao, 2013, pp. 1-6.
- [3] Lin Ye, Hongli Zhang, Jiantao Shi and Xiaojiang Du, "Verifying cloud Service Level Agreement," 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, 2012, pp. 777-782.
- [4] H. Wang, C. Yu, L. Wang and Q. Yu, "Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences," in *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 644-657, 1 July-Aug. 2018.
- [5] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman and D. Woods, "Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds," in *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 523-536, 1 July-Sept. 2017. doi: 10.1109/TCC.2015.2415794
- [6] Varadharajan and U. Tupakula, "On the Design and Implementation of an Integrated Security Architecture for Cloud with Improved Resilience," in *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 375-389, 1 July-Sept. 2017.[6]
- [7] S. Deshpande and R. Ingle, "Trust assessment in cloud environment: Taxonomy and analysis," 2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, 2016, pp. 627-631..
- [8] R. Neisse, D. Holling and A. Pretschner, "Implementing Trust in Cloud Infrastructures," 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Newport Beach, CA, 2011, pp. 524-533.
- [9] M. Fugini and G. Hadjichristofi, "Security and trust in Cloud scenarios," 2011 1st International Workshop on Securing Services on the Cloud (IWSSC), Milan, 2011, pp. 22-29.
- [10] N. Bohlol and Z. Safari, "Systematic parameters vs. SLAs for security in cloud computing," 2015 9th International Conference on e-Commerce in Developing Countries: With focus on e-Business (ECDCC), Isfahan, 2015, pp. 1-8.
- [11] El Makkaoui, Khalid Ezzati, Abdellah beni hssane, Abderrahim Motamed, Cina. (2016). Data confidentiality in the world of cloud. 84. 305-314.

- [12] S. B. Hosseini, A. Shojaei and N. Agheli, "A new method for evaluating cloud computing user behavior trust," 2015 7th Conference on Information and Knowledge Technology (IKT), Urmia, 2015, pp. 1-6.
- [13] S. Dey and S. K. Sen, "SVM- A novel trust measurement system in cloud service," 2018 Emerging Trends in Electronic Devices and Computational Techniques (EDCT), Kolkata, 2018, pp. 1-5.
- [14] M. K. Tripathi and V. K. Sehgal, "Establishing trust in cloud computing security with the help of inter-clouds," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, 2014, pp. 1749-1752.
- [15] M. I. M. Saad, K. A. Jalil and M. Manaf, "Achieving trust in cloud computing using secure data provenance," 2014 IEEE Conference on Open Systems (ICOS), Subang, 2014, pp. 84-88.
- [16] W. Abderrahim and Z. Choukair, "Trust Assurance in Cloud Services with the Cloud Broker Architecture for Dependability," 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, New York, NY, 2015, pp. 778-781.
- [17] A. S. Horvath and R. Agrawal, "Trust in cloud computing," SoutheastCon 2015, Fort Lauderdale, FL, 2015, pp. 1-8.
- [18] S. Saadat and H. R. Shahriari, "Towards a process-oriented framework for improving trust and security in migration to cloud," 2014 11th International ISC Conference on Information Security and Cryptology, Tehran, 2014, pp. 220-225.
- [19] A. Hoeller and R. Toegl, "Trusted Platform Modules in Cyber-Physical Systems: On the Interference Between Security and Dependability," 2018 IEEE European Symposium on Security and Privacy Workshops (EuroSPW), London, 2018, pp. 136-144.