Cluster Computing (2019) 22:S12459–S12471
https://doi.org/10.1007/s10586-017-1657-y

Monitoring IaaS using various cloud monitors

Absa Stephen[1] · **Shajulin Benedict**[2] · **R. P. Anto Kumar**[3]

Received: 6 September 2017 / Revised: 10 December 2017 / Accepted: 27 December 2017 / Published online: 9 January 2018

Abstract
In cloud computing, the performance of infrastructure as a service is critical because of its divergence in area. The cloud providers guarantee that the resources will be available around the clock. The providers assure that the period of unavailability of resources is very less. Recently the cloud users have increased rapidly; therefore the providers also have increased, basically increasing the complexity of the infrastructure. This complex infrastructure should be allocated properly to the users and the availability of resources should be notified to the users. So monitoring of these resources constantly is critical. In this analysis, comparison of various monitoring tools in terms of SLA parameters are measured and tabulated. For the comparison, the Amazon cloud instances are monitored with three different monitoring tools like CloudWatch monitoring, IDERA uptime cloud monitor and ManageEngine applications manager. The SLA parameters of IaaS are CPU utilization, network in, network out, disk read, disk write, response time and memory usage. In addition with Amazon instances, servers like Tomcat and data base like PostgreSQL are also monitored and their performance parameters are also analyzed. The instances monitored by cloudwatch monitoring gives twice the range of CPU Utilization than the others. The network data transfer is also high using cloudwatch.

Keywords Performance analysis · Availability · Monitoring · Software as a service · Servers

1 Introduction

Cloud computing is a developing field which permits the users to acquire benefits on all types of resources it provide. The main aim of cloud is to cut cost, and facilitates the users focus on their trade as an alternative of being delayed by IT complications. The foremost technology used in cloud computing is virtualization. Cloud computing is a developing area where three types of services are provided to customers. The service providers provide infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) to the customers.

Infrastructure as a service (IaaS) refers to the particulars of infrastructure like physical computing resources such as compute, storage, networking and networking services. IaaS-cloud providers contribute these resources based on their need from their huge content of resources presents any where throughout the universe. The PaaS dealers present a development environment to the developers. The development environment includes the operating system, databases etc. In software as a service (SaaS) model, users have admittance to any purpose software and databases without installing it, on their own computer.

In this context the infrastructure service parameters are taken and its performances are analyzed. Customers can procure the resources and reimburse for the resources that are utilized similar to utility billing. When the customer request for the IaaS resources, they could sign a bond with service providers if they could offer the mandatory resources to users after concession. The agreement that is signed involving client and contributor is called service level agreement (SLA).

✉ Absa Stephen
absa@sxcce.edu.in

Shajulin Benedict
shajulin@iiitkottayam.ac.in

R. P. Anto Kumar
antokumar@sxcce.edu.in

[1] Department of Electronics and Communication Engineering, St.Xavier's Catholic College of Engineering, Anna University, Nagercoil 629003, India

[2] Indian Institute of Information Technology Kottayam, Kottayam, India

[3] Department of Computer Science and Engineering, St.Xavier's Catholic College of Engineering, Anna University, Nagercoil 629003, India

An SLA is the measurement of a service bond where the precise uniqueness of the resources being contributed are properly stated [1]. In cloud computing, a SLA verifies the service necessities that have to be assured by the contributor in turn to hold out a users cloud service agreement. The SLA IaaS parameters of cloud computing are the CPU utilization (compute), disk read, disk write and network capability. As the extended part of our work [2], to analyze the performance metrics in IaaS resources, monitoring is indispensable. Monitoring the cloud environment becomes essential to analyze the performance and IaaS parameters of cloud environment on the root of SLA agreed. The deviation of value from the monitored and agreed parameters could provide information to consumers and providers the degree of violation.

There are various cloud providers like Amazon web services (AWS), Go Grid, Microsoft Azure etc. The AWS infrastructure considered for performance analysis here are AWS EC2 [3] instances and EBS [3–5]. Amazon EC2 offers virtual compute units, identified as instances. Reorganized outlines for the instances, recognized as Amazon machine images (AMIs),which enclose the fragments essential for our server configurations. The fragments are processing unit, memory, storage, and network ability of the instances, known as *instance types*. Amazon elastic block store supplies lump memory space for use with elastic compute instances. Elastic block store capacities are extremely accessible and consistent memory capacity that is attached with every operating instance. The EC2 and EBS will be available in the similar zone. AWS EBS are also a part of utility computing that pay for what you use.

In this work, the main objective is to choose the cloud monitoring tools that are capable of monitoring Amazon web services and to monitor IaaS SLA metrics. The monitoring tools used to monitor AWS are IDERA (formerly Copperegg), CloudWatch and ManageEngine applications manager. The analysis was based on SLA metrics and its graphical representations by various cloud monitors are presented. For the analysis, ten (10 numbers) instances are launched using Amazon web services and they are connected to IDERA and ManageEngine applications manager and their output was compared with CloudWatch monitor. The main focus of this research is to guide the users to know the status of the resources and to select the best service provider.

This paper is put in order as follows: In Sect. 2 we discussed the related work on IaaS monitoring. The parameters considered for monitoring are analyzed. In Sect. 3 we have presented the cloud provider, monitoring tools used and also described how to integrate the instances running with monitoring tools. In Sect. 4 the real time graphical output of the AWS instances monitored with various monitoring tools are presented.

2 Related work

In cloud computing several service providers are there to provide various services based on customers request. Whatever the services provided by service providers, there is a need to monitor the SLA metrics that is signed by both and whenever there is deviation in the signed metrics and monitored metrics that should to be notified to both parties. So the contributors, based on monitoring Infrastructure as a service are summarized in this context.

In [6,7] Guilherme Da Cunha Rodrigues et al. presented a sketch of cloud monitoring and also assessed the extent of cloud monitoring based on management. They divided the monitoring configuration into three modules as cloud model (i.e., based on service-IaaS, PaaS and SaaS), examining inspection (whether infrastructure providers or service providers or customers have to receive the information), and monitoring center (based on service and type of resource). Also they have presented some cloud specific monitoring solutions such as Amazon CloudWatch, Monitis, Copperegg, Zennoss and RackSpace cloud monitoring.

In [8–14] Giuseppe Aceto et al. presented that monitoring is an indication of overseeing and controlling hardware and software. They also discussed the properties of cloud monitoring tool and monitoring solutions. They also have discussed all cloud platforms.

In [1] Jesús Montesa et al. discussed and analyzed different types of cloud service models. They have also proposed a cloud monitoring design which leads to the development of cloud monitoring tool. They have implemented the proposed architecture as a tool named GMonE. The authors have assesed the performance of tool by using Yahoo cloud serving benchmark (YCSB), and OpenNebula cloud middleware on the Grid'5000 experimental test bed.

In [13,15,16] Ruben Trapero et al. presented how to assure security in cloud using SLA'S. They framed the security SLA life cycle and also defined the metrics essential to be monitored. The authors also provided a relationship between metrics and measurement. They automated the remediated SLA violation whenever there was deviation in the detected value over threshold.

In [17–19] Vincent C. Emeakaroha et al. proposed an architecture that monitors and detects violation. They presented that the service level agreement is formed at the application layer and once the SLA is been violated they suggest that penalty can be issued. They also suggest it is tough to match the monitored metrics with the required metrics.

In [13,16,20] Lars Larsso et al. suggested a scheduling model for cloud federations. This model also helps to minimize the degree of SLA violation. They also have proposed an architecture which is aided for monitoring of resources.

In [21] Rima Grati et al. presented that cloud providers are responsible to preserve quality of service intensity that they have be in agreement with users. In this research paper the authors have presented that understanding of monitoring tools becomes important to maintain the agreed SLA. The authors have undergone a survey of different monitoring tools and how they fit for monitoring various providers based on applications.

In [22–24] Vijayakumar et al. presented various continues security assessment in cloud environment using NLP in SDLC, that provides continues security for enterprises applications in cloud environment.

In [25–27] Wesam Dawoud et al. presented that IaaS is the basic layer for the cloud service models. The different service models provide various types of security challenges. Since the other two layers are built up over this layer security of IaaS becomes crucial. They proposed a security model for the IaaS layer.

The IaaS provides information on resource usage of low level monitoring data for CPU, network and storage parameters [28,29]. The primary purpose of monitoring is to accumulate data in view with some routine metrics. The authors also have authorized and estimated the architectural uniqueness and the functionalities intended. They have confined the concert of the resources in terms of CPU, network utilization and response time.

In this work, the authors have analyzed the IaaS parameters to find out the degree of violation by estimating the difference between measured SLA parameter and agreed SLA. To evaluate the degree of violation, monitoring of IaaS resources becomes mandatory. The IaaS parameters like CPU utilization, storage and network interface of AWS are displayed using three different types of monitoring tools and its output variations are tabulated.

3 Monitoring Amazon web services infrastructure

In cloud computing, where resources are selected based on infrastructure as a service due its vast diversity, monitoring of this allocated infrastructure becomes essential. In this part, we have presented the various cloud monitoring tools, we have used to monitor the instances that we have launched in Amazon web services. Monitoring of infrastructure is necessary to ensure that agreement is not violated by both provider and user. The service providers guarantee 99.98% of availability of resources round the clock. In order to confirm that monitoring of infrastructures is approved, which indirectly directs the users and providers to know the status of resources. The resources of IaaS considered in this context are core (AWS instances), memory, disk read, disk write, network in and network out.

3.1 Amazon web services

The monitoring tools used to monitor the resources are Amazon CloudWatch, Idera (formerly Copperegg) and ManageEngine. In this section we explained all the monitoring tools and how they are integrated with AWS. Amazon CloudWatch is an integrated tool with Amazon web services for monitoring the resources such as EC2, EBS etc. This is a pay as you go pricing tool. CloudWatch is capable of monitoring 1233 metrics. The metrics considered here to monitor EC2 launched instances are CPU utilization, disk read bytes, disk write bytes, network in and network out. In Amazon web services 10 instances are launched and the above metrics are monitored using AWS CloudWatch.

3.2 Idera

Idera (formerly Copperegg) has been mainly designed for managing and monitoring servers and applications for smaller and mid-size concerns. Idera is capable of monitoring Vm's, servers, network devices and applications. It's also capable of reporting SLA's and also to track the performances of resources. The metrics involved in Idera are CPU usage, memory usage and disk metrics. Idera is also a tool that we have to pay for the metrics monitored.

3.3 ManageEngine

ManageEngine applications manager is designed for managing application performance management, fault management, SLA management and reporting. ManageEngine uses a single console to monitor physical, virtual and cloud applications. The performance metrics are CPU utilization, memory and disk read and writes.

3.4 Block diagram

The service provider is integrated with the monitoring tools using the credentials of the service provider. The service provider used here is Amazon service provider and the monitoring tools used are Idera, ManageEngine and Copperegg. The block diagram shown in Fig. 1 demonstrates the integration of the monitoring tools.

In Fig. 1, the EC2 instances launched in AWS service provider are integrated with cloud monitoring tools using the application peripheral interface.

After integrating the AWS instances with the tools, all the instances launched will be visible in the dashboards of the monitoring tools. Each and every instance launched will be displayed with its own ID and it shows the status of all instances.

4 Monitoring performance metrics

In this section we present the real time outputs we got using the monitoring tools Idera, ManageEngine and CloudWatch. For the analysis 10 AWS EC2 instances are launched in the Oregon region. These 10 AWS instances are monitored and their CPU utilization, memory usage, disk read, disk write, network in and network out. The running instances in Amazon web services are shown in Fig. 2.

4.1 CloudWatch monitoring

For the analysis, we first present the performance metrics using CloudWatch monitoring. AWS CloudWatch monitoring is a tool within Amazon web services to monitor the infrastructure like launched instances and memory.

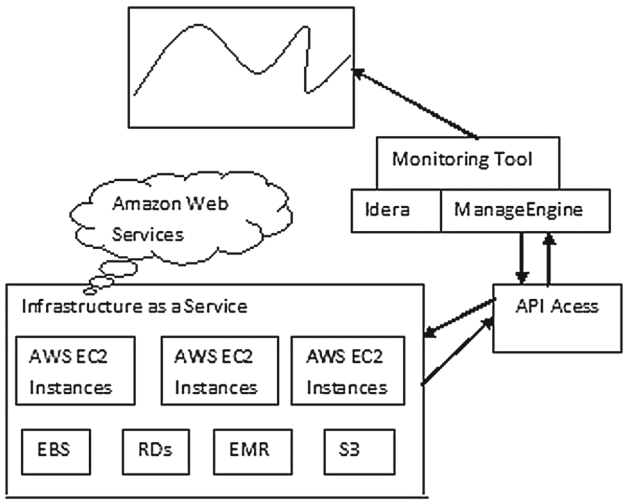


Fig. 1 AWS infrastructure integrated with monitoring tools

The performance metrics like CPU utilization, network in, network out, disk read, disk write are shown in Figs. 3, 4, 5 and 6 respectively Fig. 2, shows the ID'S of all the 10 instances running and also shows the region where the instances are present. All these instances CPU utilization are displayed using various colors in the output.

Figure 3 shows the CPU utilization [30] of all the instances which are launched. The fraction of owed EC2 computes unit that is at present in utilization on the launched instance. This parameter recognizes the actual power essential to execute an function on a chosen instance. In Y-axis the percentage of utilization is plotted with respect to time in X-axis. The range of utilization ranges from 2.5 to 40% using CloudWatch. The duration of the time plotted is for every 5 min. For every 5 min, the gradual variation in utilization can be noted.

Figures 4 and 5 shows the network in and network out [30] performance metrics monitored using CloudWatch respectively. Network in defines the amount of bytes accepted on all the network boundarys by the instance. This metric discovers the degree of arriving network transfer to an application on a single instance. This output plots the number of bytes along its Y-axis and time along its X-axis. The maximum amount of byte transfer to an instance is around 700,000,000 bytes and a minimum less than 100,000,000 bytes.

In a similar way network out defines the quantity of bytes transferred out on all network boundary by the instance. This parameter discovers the degree of network transfer to an application on an instance. The output shown in Fig. 5, plots the amount of bytes in Y-axis and time along its X-axis. The output plotted in both network in and network out are for each 5 min interval. From the output shown the maximum amount of data transferred out from an instance is approximately 24,000,000 bytes and minimum amount is less than 70,000,000 bytes of data.

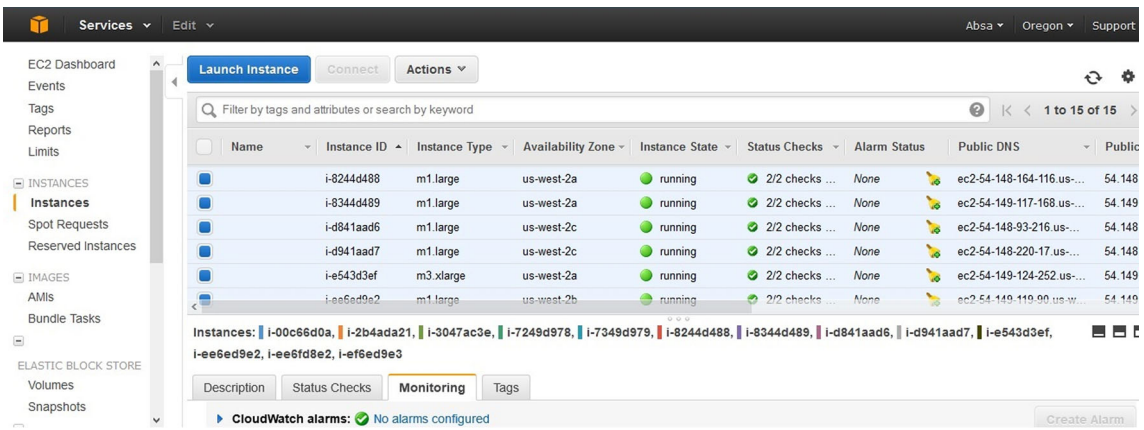


Fig. 2 All the instances (10) launched in Amazon web services

Fig. 3 CPU Utilization of 10 launched instances using CloudWatch

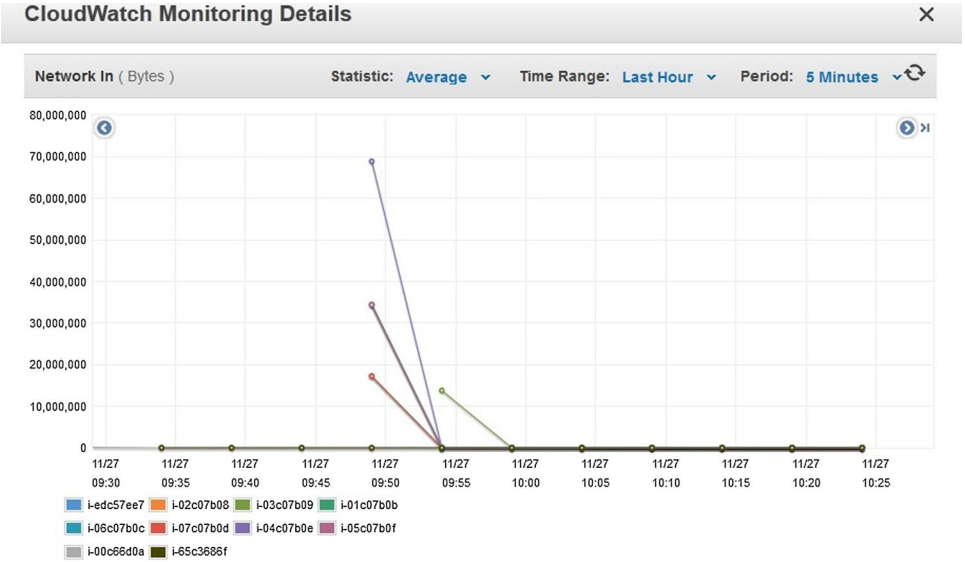


Fig. 4 Network in for 10 launched instances using CloudWatch

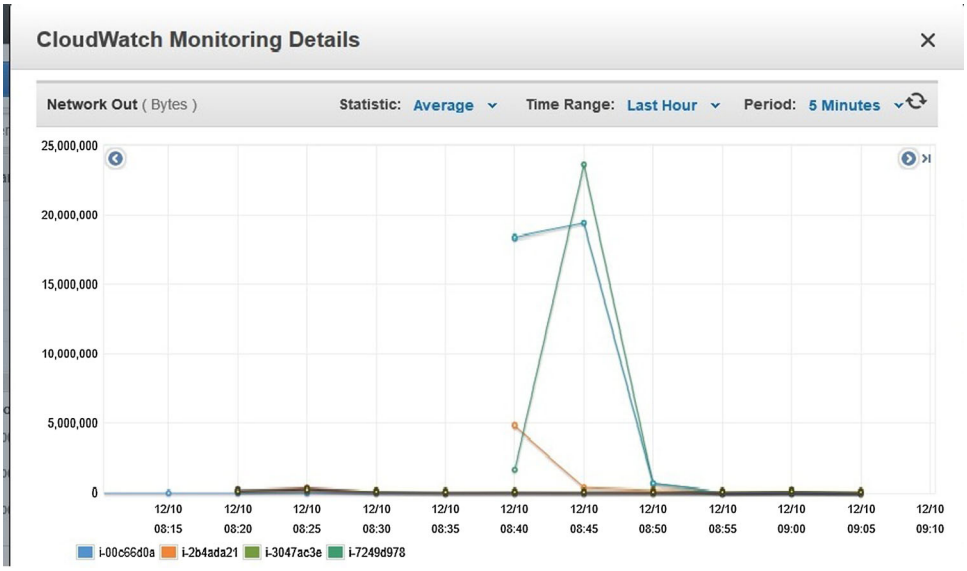


Fig. 5 Network out for 10 launched instances using CloudWatch

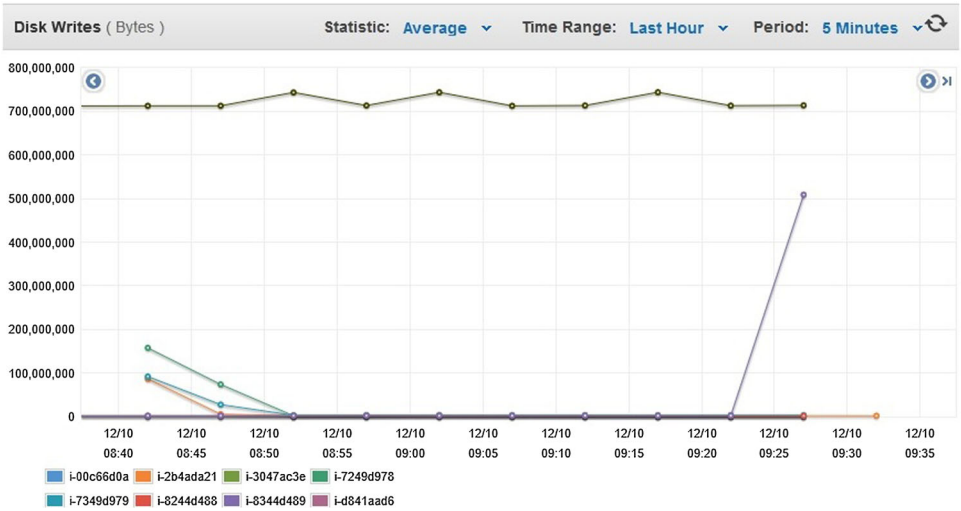


Fig. 6 Disk writes for 10 launched instances

Figure 6 shows the disk write for all the 10 AWS instances launched. Bytes written to the store volumes available to the instance are called Disk Write [30].

In a similar manner, bytes read from store volumes of all the instances are called disk read. Disk write is the metric used to find out the capacity of the information the function reads onto the hard disk of the instance. Disk read is the metric used to find out the capacity of the information the function reads from the hard disk of the instance.

4.2 Idera

Idera acquired the copperegg which is software as a service tool used to monitor the performance metrics of both public and private cloud environments [31,32].

The products of copperegg tool are integrated with public clouds like Amazon and Rackspace. Copperegg is a tool which supports in system management and involved in monitoring the AWS instances availability. The metrics used here for analysis are similar to CloudWatch metrics such as CPU utilization, network in, network out, disk read and disk write.

For the same number of 10 instances launched in AWS, the performance metrics using Copperegg are presented in this section. In Idera tool, all the instances with their IDs are displayed as shown in Fig. 7.

Once all the 10 instances are attached to uptime cloud monitor, AWS option is clicked and the performance parameters like CPU utilization, network in and network out and memory usage are displayed against each instance ID.

Figure 8 shows the CPU utilization and network of a single AWS instance with ID i-1469cc0c. The CPU utilization for the instance shown is 9.9% and network data transfer is 13.8 bytes per sec peak value. In a similar way the Idera is able to measure the same performance metrics of any server which is attached to it. In Fig. 9, Disk I/O of the server which is attached to Idera is shown. The Copperegg also monitors the EBS attached with EC2. The amount of volume read and write are displayed as shown in Fig. 10.

Amazon elastic block store (Amazon EBS) tranfers information to CloudWatch for several parameters. The metrics for Amazon EBS are volume read bytes and volume write bytes. These metrics provide data on the I/O operation for a particular phase of instance. The added value reports the entire number of bytes transmited during that period. Information is only accounted to Amazon CloudWatch when the volume is active. If the volume is inactive, no data is accounted to Amazon CloudWatch.

4.3 ManageEngine applications manager

ManageEngine applications manager is the monitoring software used to monitor heterogeneous resources such as application servers, web servers, databases, network services, virtual systems and cloud resources [33]. It also supports monitoring Amazon web service instances, EBS volumes and Amazon S3 buckets. All the 10 AWS instances launched are displayed in the ManageEngine application manager tool. It gives information about the availability of particular instances directly compared to other monitoring tools. It also gives details about downtime, mean time between failure (MTBF) and mean time to repair directly (MTTR) as shown in Fig. 11.

These ManageEngine tool displays the uptime and down time directly compared with other tools. If the instance was not available for the acquired period of time, the user could know directly the violation of the agreement. From down time data displayed the user can also understand the duration of the time the instances are unavailable. From the unavailable period the user can aver penalty from service provider. Also details like MTTR and and MTBF can also be got direcly using ManageEngine. MTTR is the normal time needed to

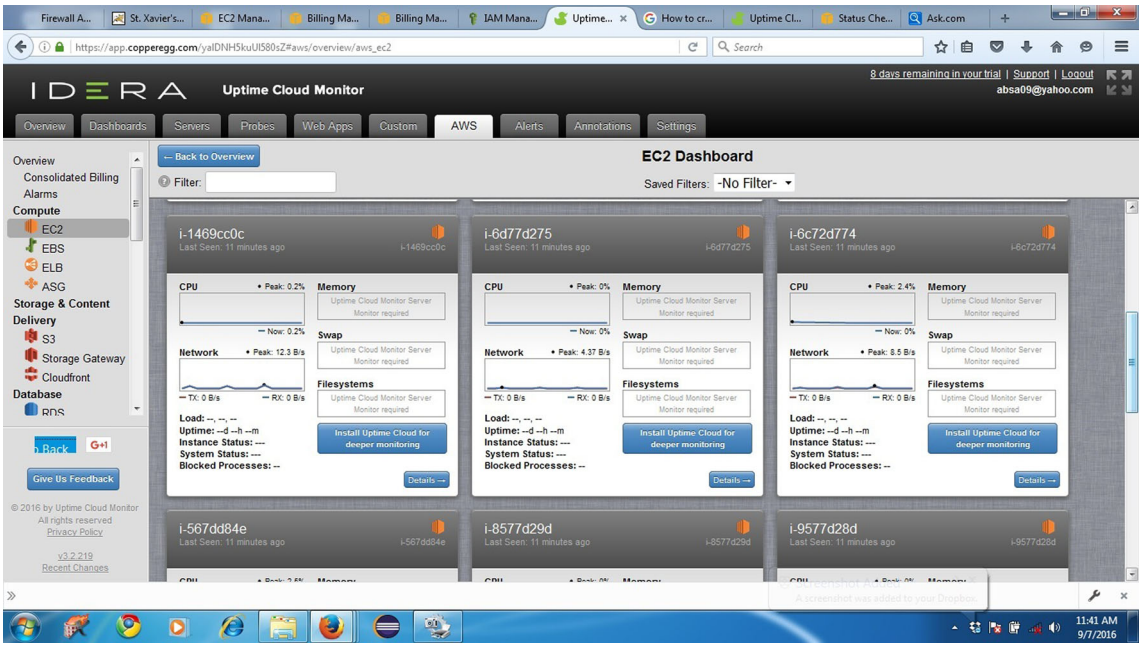
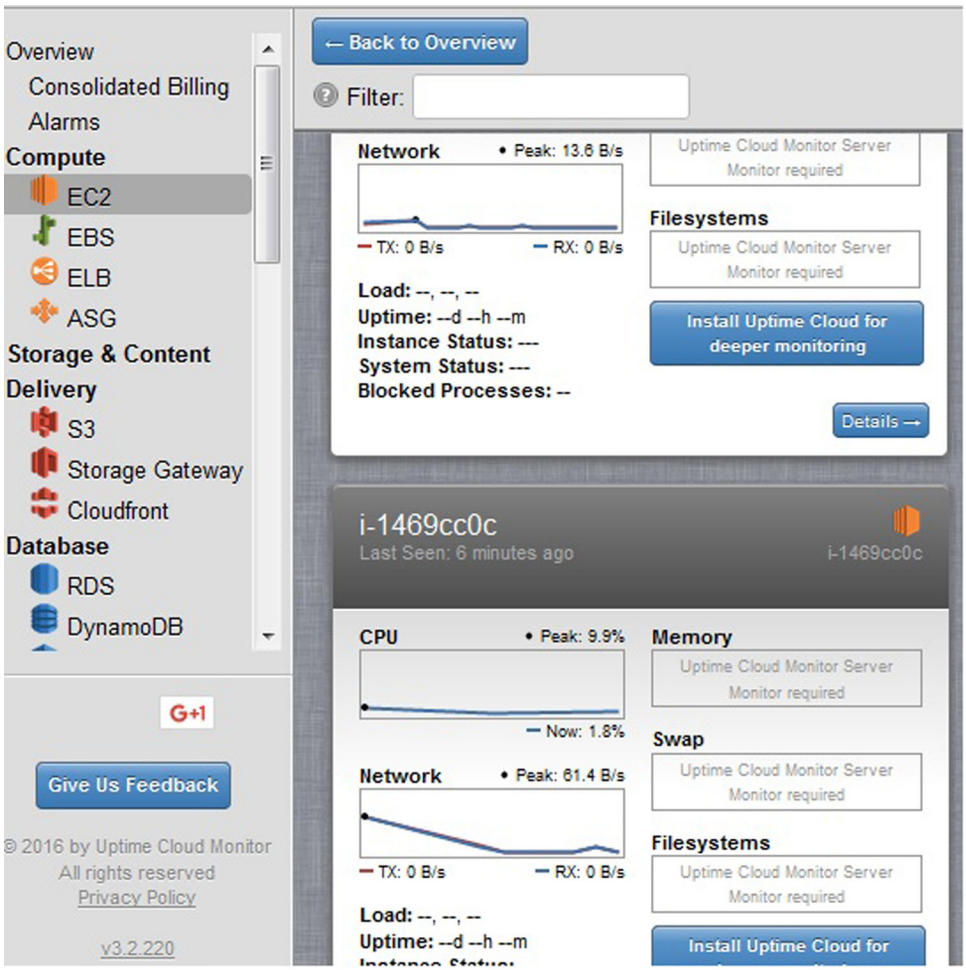


Fig. 7 All the 10 instances attached with Idera uptime cloud monitor

Fig. 8 CPU utilization and network parameter of a single instance with ID i-1469cc0c



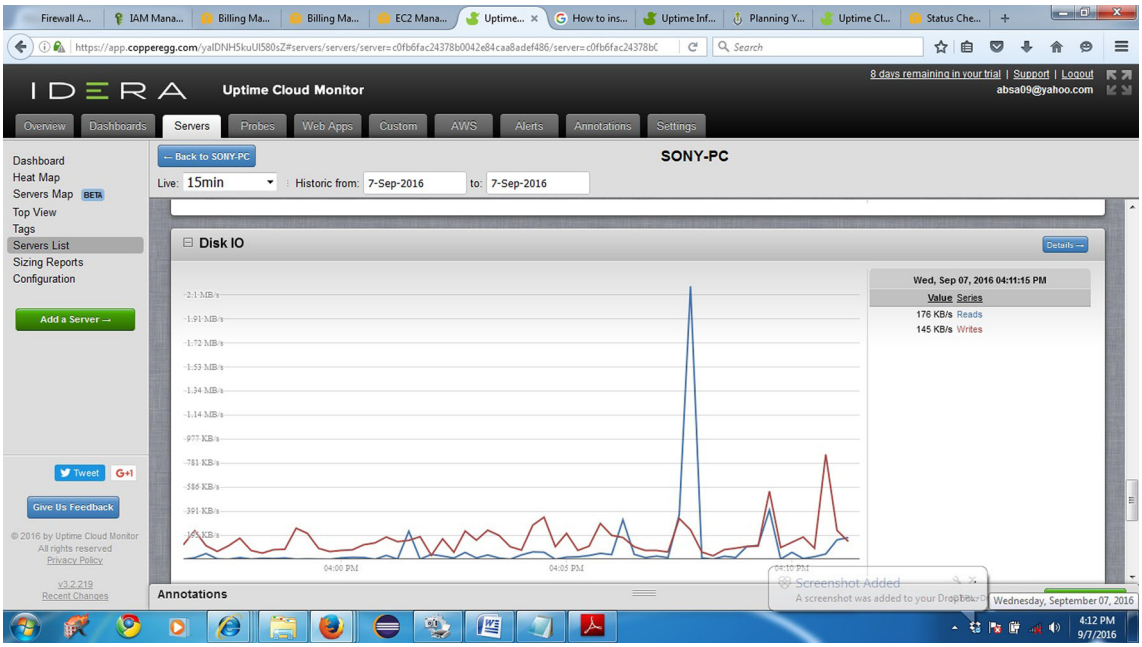


Fig. 9 Disk I/O of server attached with Idera

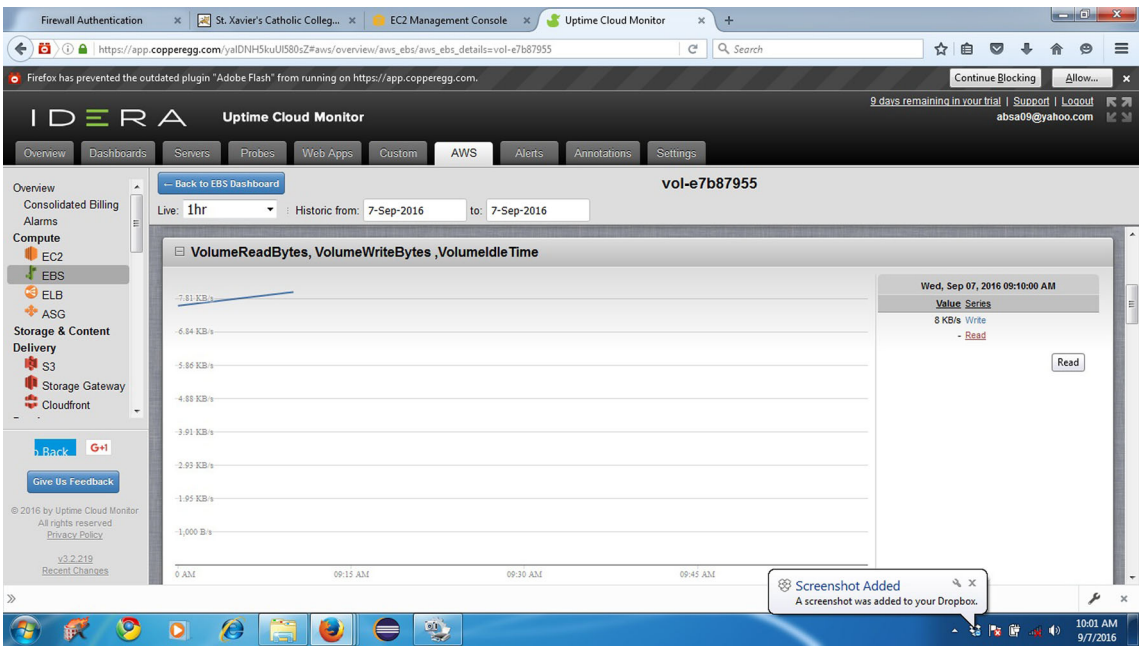


Fig. 10 Volume read/write on AWS EBS

repair a failed instance and return it to running status. MTBF is the standard quantity of time that an instance works before failing. This unit of measurement contains only prepared time involving failures and does not take in renovate times.

The CPU utilization of all the launched instances using ManageEngine is shown in Fig. 12. The maximum utilization of compute unit is 20%. It has displayed the CPU utilization output for every 5 min like CloudWatch.

The network in and network out of all the launched AWS instances are displayed using ManageEngine as shown in Fig. 13. Just like CloudWatch the time interval chosen is 5 min. The maximum amount of byte transfer to an instance is around 7800 bytes. In a similar way network out the quantity of bytes transferred out on all network interfaces by the instance is around 4500 bytes. The advantage of ManageEngine is it gives the time duration of up-time and

Fig. 11 Instances displayed with uptime and down time using ManageEngine

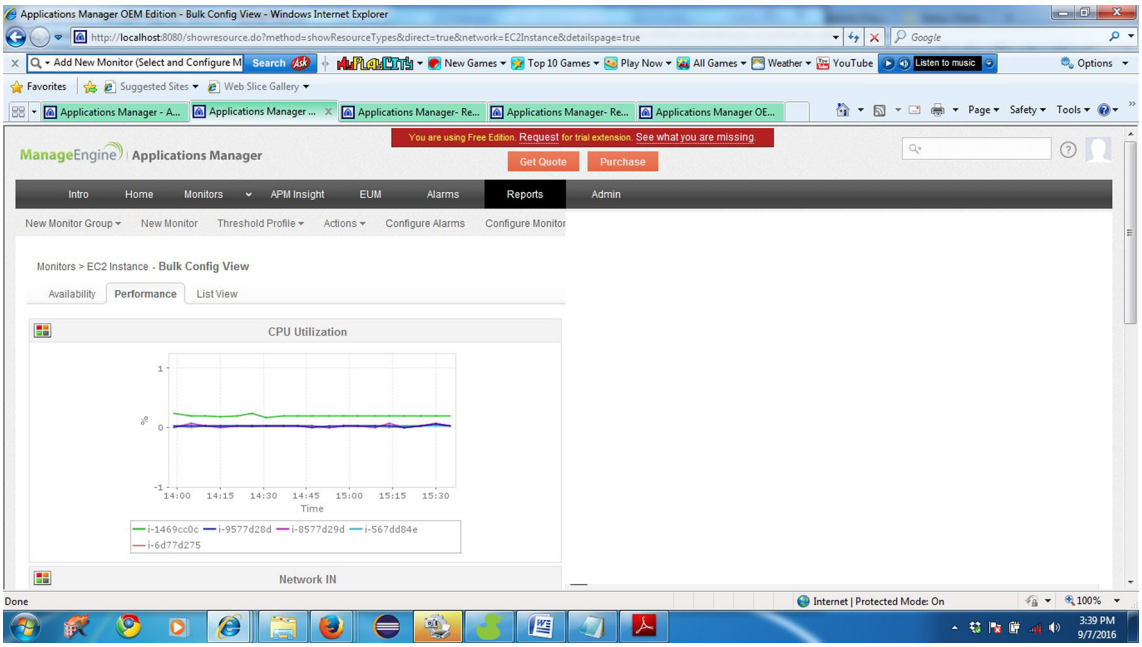


Fig. 12 CPU utilization of all 10 instances using ManageEngine.

down-time where users can directly view availability of each and every instance running.

4.4 Servers

Using Copperegg and ManageEngine the users can monitor any number of added servers. Added servers are Sony pc and an AWS instance. The attached servers CPU utilization and data transfer performance metrics are shown in Fig. 14.

In a similar way the ManageEngine supports number of application servers like Tomcat server, Jetty, Glassfish etc. and also a number of database servers like PostgreSQl, MySQL etc. For analysis we have monitored PostgreSQL and Tomcat server and their performance metrics are also shown in Fig. 15.

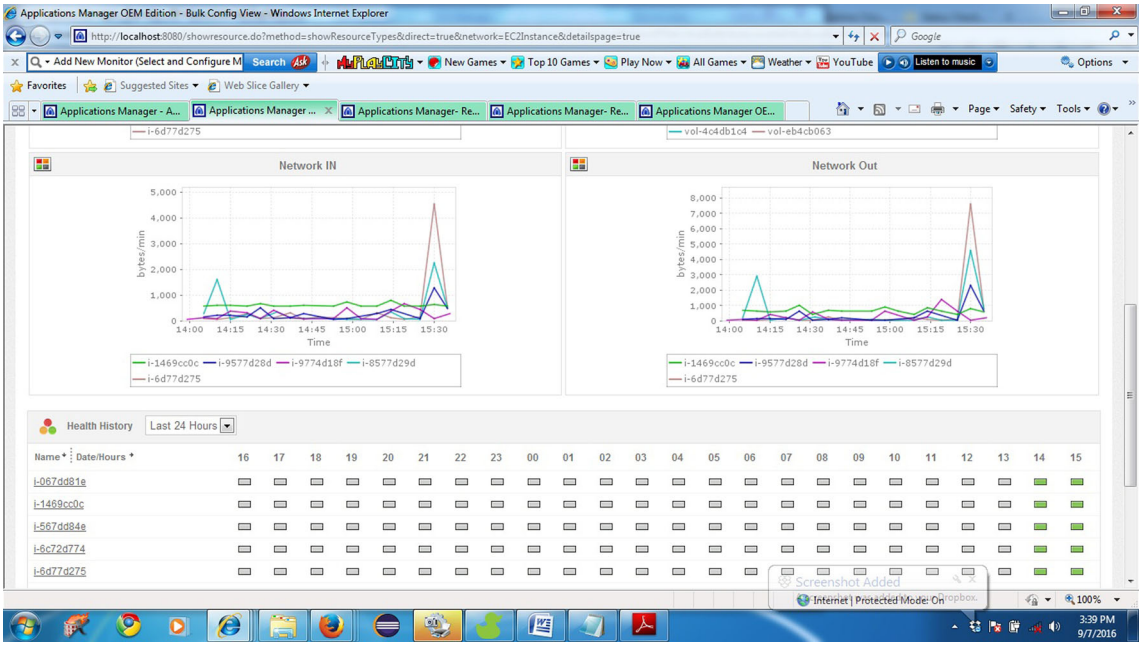


Fig. 13 Network in and out for all 10 instances using ManageEngine

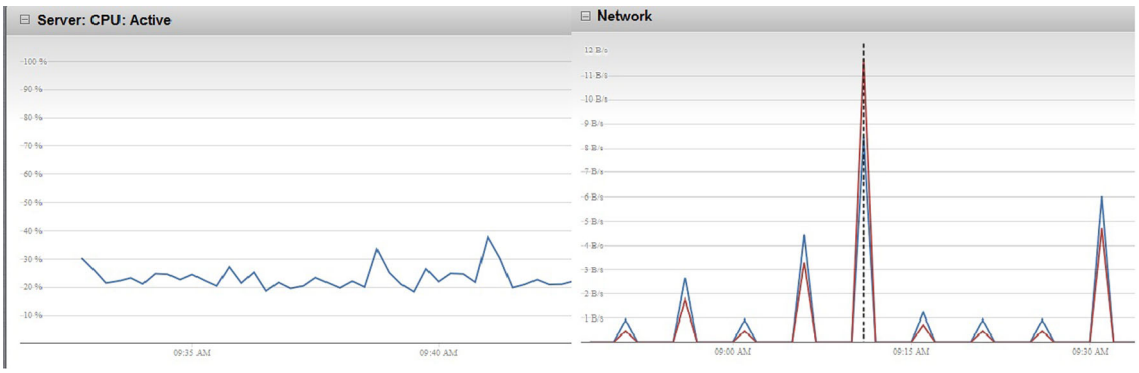


Fig. 14 Performance metrics of attached servers

The Tomcat server's ID is 192.168.56.1_Tomcat _ server and the PostreSQL database server's Id is 192.168.56.1_ PGSQL. The availability of these servers, and their response time are also displayed on this ManageEngine tool as shown in Fig. 16. The availability is 100% for all the servers and the response time of the launched servers varies as shown in Fig. 16.

From the above results, the monitors used to monitor infrastructure as a service give the performance metrics that directs the user directly or indirectly to be familiar with availability. Availability is an important IaaS metrics that all service providers guarantee to be 99.9%. Here we have tested the availability of Amazon instances and its values using the three monitoring tools. But from the results we can understand only ManageEngine could display the availability and response time directly whereas the other two monitors could not as tabulated in Table 1.

Based on the percentage of availability we can come to the conclusion that the resources are available for the agreed period of time. If the percentage is less, it shows that the agreement is violated and the user can claim penalty. Thus monitoring and performance analysis using various monitoring tool becomes essential to check SLA agreement. The Monitoring tools CloudWatch and Idera do not provide direct information regarding availability but status can be monitored using the performance metrics. In Fig. 16, the uptime shows the availability. The instance with uptime 0% represents that it is unavailable during the time period.

In a similar way, the performance metrics of the AWS instances running are also tabulated as shown in Table 2.

From the performance analysis it is significant to assess the performance of cloud environments to recognize the degree of violation. In the table 2 we have tabulated the CPU Utilization and Network transfer in the interface which are the

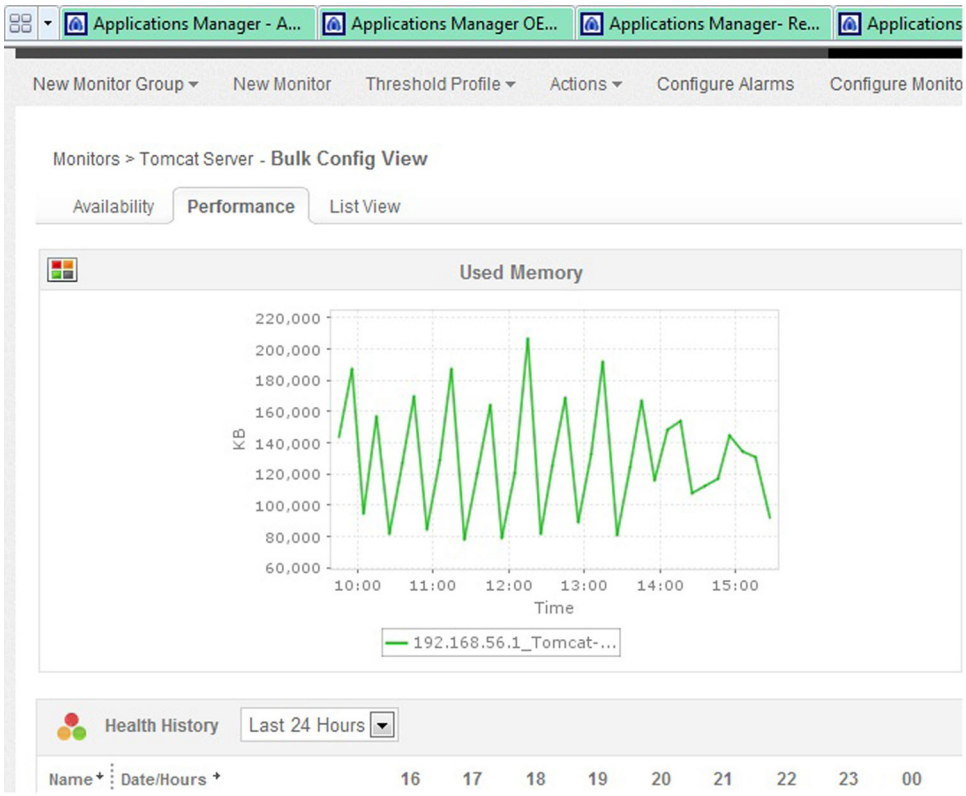


Fig. 15 Memory used performance metrics of Tomcat server

Fig. 16 Direct availabilty chart for all servers and instances using ManageEngine

Table 1 Availability

S.No	Service provider	Monitoring tool	Availability in %
1	Amazon	CloudWatch	-
2	Web	Idera	-
3	Services	ManageEngine	100

Table 2 Performance Metrics

S.No	Service provider	Monitoring tool	CPU utilization in % (Max value)	Network data transfer in bytes (peak Value)
1	Amazon	CloudWatch	40	1166667 Bytes /Sec
2	Web	Idera	10	71.1 Byte/sec
3	Services	Manage Engine	20	12.3Bytes/sec

Table 3 Disk Read/Write

S.No	Service provider	Monitoring tools	Disk IO
1	Amazon web services	CloudWatch	3.7 MB
2		Idera	2.2 MB
3		ManageEngine	7000 KB

modules of compute units (AWS EC2). Table 3 shows the status ofs disk read/write for the instances launched.

This analysis is carried out to check whether the infrastructure guaranteed by the service provider is available throughout the time duration that is guaranteed.

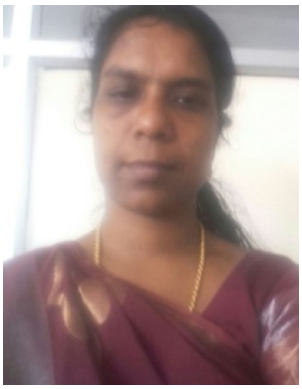
5 Conclusions and future scope

To analyze the performance of IaaS in cloud the service level agreement (SLA) signed between the provider and the customer plays a significant role. In SLA the metric availability of the resources is the major issue to carry on computation. In our research contribution we have taken availability as the major metric and carried on the analysis. Here the cloud monitoring tool ManageEngine displays the availability in percentage whereas the other tools CloudWatch and Idera do not. But they display the computation parameters like CPU utilization, network in and out, disk read and write etc. When there is no availability of resources, the user and provider can conclude it as agreement violation. Cloud computing is a utility computing where penalty can be issued to the provider since they guarantee 99.9% availability. This research is now focused only on Amazon web services and in future it is to be extended to a number of service providers and its availability details can be tabulated. The number of monitoring tools that supports IaaS can also be increased and a mathematical formulation for penalty in case of unavailability can be framed. This analysis, if extended for number of service providers, based on availability, it may be ranked and the user who looks for high availability can choose the service provider without any difficulty.

References

1. Montesa, J., Sánchez, A., Memishi, B., Pérez, M.S., Antoniu, G.: GMonE: a complete approach to cloud monitoring. Futur. Gener. Comput. Syst. **29**(8), 2026–2040 (2013). https://doi.org/10.1016/j.future.2013.02.011
2. Absa, S., Benedict, S.: A survey on SLA based cloud architectures. J. Converg. Inf. Technol. **11**(1), 1–12 (2016)
3. Kertesz, A., Kecskemeti, G., Brandic, I.: An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous cloud environments. Futur. Gener. Comput. Syst. **32**, 54–68 (2014). https://doi.org/10.1016/j.future.2012.05.016
4. http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.htm
5. Cloud computing management. http://www.channelfutures.com/private-cloud
6. Da Cunha Rodrigues, G., Calheiros, R.N.: Monitoring of cloud computing environments: concepts, solutions, trends, and future directions. ACM 2016, SAC 2016 (2016). https://doi.org/10.1145/2851613.2851619
7. Weingärtner, R., Bräscher, G.B., Westphall, C.B.: Cloud resource management: a survey on forecasting and profiling models. J. Netw. Comput. Appl. **47**, 99–106 (2015). https://doi.org/10.1016/j.jnca.2014.09.018
8. Alecsandru, P., Patriciu, V.V.: Digital forensics in Cloud computing. Adv. Electr. Comput. Eng. **14**(2), 101–108 (2014). https://doi.org/10.4316/AECE.2014.02017
9. Aceto, G., Botta, A., de Donato, W., Pescapè, A.: Cloud monitoring: a survey. Comput. Netw. **57**(9), 2093–2115 (2013). https://doi.org/10.1016/j.comnet.2013.04.001
10. Alhamazani, K.: An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art. Computing **97**(4), 357–377 (2015)
11. Computing Performance issues and performance analysis tools for HPC cloud applications: a survey. **95**(2), 89–108 (2013). https://doi.org/10.1007/s00607-012-0213-0
12. Giannakou, A., Rillingy, L., Pazatz, J.-L., Majorczyky, F., Morin, C.: Towards self adaptable security monitoring in IaaS clouds,"15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CC-GRID 2015) (2015)
13. de Chaves, S.A., Uriarte, R.B., Westphall, C.B.: Toward an architecture for monitoring private clouds. IEEE Commun. Mag. **49**(12), 130–137 (2011)
14. Petcu, D., Crăciun, C.: Towards a security SLA-based cloud monitoring service. In: CLOSER 2014 - 4th International Conference on Cloud Computing and Services Science, pp. 593-603
15. Trapero, R., Modic, J., Stopar, M., Taha, A., Suri, N.: A novel approach to manage cloud security SLA incidents. Futur. Gener. Comput. Syst. **72**, 193–205 (2017). https://doi.org/10.1016/j.future.2016.06.004
16. Ghosha, R., Longo, F., Naik, V.K., Trivedi, K.S.: Modeling and performance analysis of large scale IaaS Clouds. Futur. Gener. Comput. Syst. **29**(5), 1216–1234 (2013). https://doi.org/10.1016/j.future.2012.06.005
17. Stantchev, V., Schröpfer, C.: Negotiating and enforcing QoS and SLAs in grid and cloud computing. GPC 2009, LNCS 5529, 25–35 (2009)

18. Casalicchio, E., Silvestri, L.: Mechanisms for SLA provisioning in cloud-based service providers. Comput. Netw. **57**(3), 795–810 (2013). https://doi.org/10.1016/j.comnet.2012.10.020
19. Vincent, C., Emeakaroha, V.C., Ferreto, T.C., Netto, M.A., Brandic, I., De Rose, C.A.: CASViD: application level monitoring for SLA violation detection in clouds. In: IEEE 36th Annual Conference on Computer Software and Applications (COMPSAC) (2012). https://doi.org/10.1109/COMPSAC.2012.68
20. Larsson, L., Henriksson, D., Elmroth, E.: Scheduling and monitoring of internally structured services in cloud federations. In: IEEE Symposium on Computers and communications (ISCC) (2011). https://doi.org/10.1109/ISCC.2011.5984012
21. Grati, R., Boukadi, K., Ben-Abdallah, H.: Overview of IaaS monitoring tools. In: IEEE/ACS 12th International Conference on Computer Systems and Applications (AICCSA) (2015). https://doi.org/10.1109/AICCSA.2015.7507146
22. Vijayakumar, K., Arun, C.: Automated risk identification using NLP in cloud based development environments. J. Ambient Intell. Hum. Comput., 1–13 (2017). https://doi.org/10.1007/s12652-017-0503-7
23. Vijayakumar, K., Arun, C.: Continuous security assessment of cloud based applications using distributed hashing algorithm in SDLC. Clus. Comput. 1–12 (2017). https://doi.org/10.1007/s10586-017-1176-x
24. Vijayakumar, K., Arun, C.: Analysis and selection of risk assessment frameworks for cloud based enterprise applications. Biomed Res ISSN: 0976-1683 (Electronic) (2017)
25. Dawoud, W., Takouna, I., Meine, C.: Infrastructure as a service security: challenges and solutions. In: The 7th International Conference on Informatics and Systems (INFOS) (2010)
26. Varatharajan, R., Manogaran, G., Priyan, M.K., Sundarasekar, R.: Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm. Clus. Comput. https://doi.org/10.1007/s10586-017-0977-2
27. Varatharajan, R., Hariharan, N., Perumal, S., Sankar, A.: A Novel Method to Increase the coupling efficiency of laser to single mode fibre. Wirel. Pers. Commun. **87**, 419–430 (2016). https://doi.org/10.1007/s11277-015-3028-4
28. Katsaros, G., et al.: A self-adaptive hierarchical monitoring mechanism for clouds. J. Syst. Softw. **85**(5), 1029–1041 (2012)
29. Fatema, K., Emeakaroha, V.C., Healy, P.D., Morrison, J.P., Lynn", T.: A survey of cloud monitoring tools: taxonomy, capabilities and objectives. J. Parallel Distrib. Comput. **74**(10), 2918–2933 (2014). https://doi.org/10.1016/j.jpdc.2014.06.007
30. http://docs.aws.amazon.com
31. https://en.wikipedia.org/wiki/CopperEgg
32. https://en.wikipedia.org/wiki/Idera_Software
33. ManageEngine application manager user guide. www.manageengine.com/ServiceDeskPlus



Absa Stephen was graduated in Electronics and Communication Engineering in 1999 from Manonmaniam Sundaranar University, Tirunelveli India. In 2006, she received M.E degree in Embedded System Technologies from Anna University' Chennai, India. From January 2000 to July 2001 she worked as a lecturer at S.A. Raja's polytechnic college. From August 2001 to July 2004 she worked as lecturer at Sun College of Engineering and Technology. At present she is working as Assistant Professor at St. Xavier's Catholic College of Engineering, India. Her research interest includes Cloud Computing, Networking and Embedded Technologies. She is a Life member of ISTE.

Shajulin Benedict graduated in 2001 from Manonmaniam Sunderanar University, India, with Distinction. In 2004, he received M.E. Degree in Digital Communication and Computer Networking from A.K.C.E, Anna University, Chennai. He is the University second rank holder for his masters. He did his Ph.D. degree in the area of Grid scheduling under Anna University, Chennai (Supervisor - Dr. V. Vasudevan, Director, Software Technologies Group of TIFAC Core in Network Engineering). He was affiliated towards the same group and published more papers in Int. Journals. After his Ph.D. award, he joined a research team in Germany to pursue PostDoctorate under the guidance of Prof. Gerndt. Later, he worked as Professor at SXCCE Research Centre of Anna University-Chennai and he established the HPCCLoud Research Laboratory. Now, he works at the Indian Institute of Information Technology Kottayam, an institute of national importance, India. He is also a Guest Scientist in TUM, Germany. His research interests include Grid scheduling, Performance Analysis of parallel applications (including exa-scale), IoT Cloud Computing, and so forth.

R.P. Anto Kumar was graduated in Electronics and Communication Engineering in 1998 from Bharathiar University Coimbatore, India. He obtained his M.E. degree from Government College of Engineering Tirunelveli in 2000, specializing in Computer Science and Engineering. He completed his Ph.D. from Bharathiar University Coimbatore in 2014 under the specialization of Computer Science. He is in teaching profession for the past fifteen years. Currently he is working as a Professor in the Department of Computer Science at St. Xavier's Catholic College of Engineering, India. His area of interest are Image processing, Biometrics and multimedia. He is a Life member of ISTE, CSI and professional member of IET.