*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 28, Issue 4, Ser. 1 (July. – August. 2026), PP 51-60*
*www.iosrjournals.org*

# An Empirical Evaluation Of AWS And Microsoft Azure Based On Performance Metrics

Yosef Abdulhamed Ramadan Ashibani
*Department Of Telecommunication & Aviation Information Technology, Technical College Of Civil Aviation & Meteorology, Esbea, Libya*

***Abstract:***
*The direction of the industry has shifted toward cloud computing, as this technology is now part of most online services. Cloud computing provides various advantages, such as cost savings. There are different providers available, and users need to select the most suitable ones for their applications. Therefore, it is essential that the suppliers align with the consumers' objectives and provide standard evaluation tools to assess the features of their services. Nevertheless, the tools provided may not suit the demands of users. Consequently, there should be generic evaluation categories to ensure users have some flexibility in deciding on different cloud providers. This paper presents a comparison of the evaluation between the two leading providers in cloud computing, AWS and Microsoft Azure. The comparison achieved will help users assess which supplier will be most suitable for them.*
***Keywords:*** *Cloud Computing; Security; Scalability; Availability; Usability; Amazon Web Service (AWS); Microsoft Azure.*

Date of Submission: 05-07-2026 Date of Acceptance: 15-07-2026

## I. Introduction

The National Institute of Standards and Technology (NIST) provided a concrete definition of cloud computing, which the community has accepted as the official definition. Cloud computing is a paradigm where users can access the shared resources of the information technology (IT) infrastructure like computing power, storage, networking, applications, and associated services on-demand via network connectivity. They can be provisioned and retracted easily, with minimal administrative overhead and minimal direct involvement from the cloud provider. Flexible, scalable, and efficient use of resources are a number of defining characteristics of this model that distinguish it from traditional computing [1].

There are three different service models of cloud computing, which are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

In the SaaS model, the consumers are provided with the capability of running and using their applications on the provider’s infrastructure without managing or controlling the underlying infrastructure, such as servers, networks, operating systems, and storage. In PaaS, consumers are provided with more capabilities, such as creating an application and deploying it, with control over the deployed applications and some configuration of the hosted environment. In IaaS, consumers are provided with more capabilities than in the other two models. These capabilities include provisioning networks, processing, storage, and other services [2]. Although consumers do not have control over the underlying infrastructure, they have control over deployed applications, operating systems, storage, and some limited network configurations such as storage encryption, authentication, and firewalls.

Even though cloud computing services reduce the cost, many concerns come to the surface. These concerns are presented in seven categories, each with different attributes [3]. Thus, it is not straightforward for these categories to be tested among available providers directly since there are differences between the provider models and each of them provides services with different labels. Consequently, there is a need for a method or a technique that gives consumers the capability to measure the quality of cloud services and provides potential users with the best or better choice of cloud providers. On the other hand, the criteria for selecting service providers are different and depend on the applications used and the users.

This paper provides a comparison between two of the most widely used cloud providers. These cloud providers are Amazon Web Service (AWS) and Microsoft Azure.

The comparison of these providers will be based on the Service Measurement Index (SMI), which defines important aspects of cloud providers that need to be investigated to provide an objective opinion on the provided platform. The goal of doing this is to provide developers with more information to make an informed decision on which provider is the most appropriate for them.

The remainder of the paper is organized as follows. Section II presents related work. Section III describes the evaluation indicators used to compare the cloud providers. The metrics employed are described, and the reason for their inclusion in the comparison is discussed. Section IV provides a comparative evaluation of AWS and Microsoft Azure with respect to the measures described in Section III. Section V presents the main findings of the study, along with the conclusions, future research directions, and potential improvements to the proposed work.

## II. Related Work

Cloud computing is the most popular way to run modern apps because it is flexible, scalable, and cheaper than paying for services as they are used. Amazon Web Services (AWS) and Microsoft Azure have been the dominant public cloud providers; therefore, a lot of research has been done on their capabilities, performance, cost, and quality of service. In [4], cloud-to-user latency was looked at across multiple geographical regions to compare the network performance of AWS and Microsoft Azure. It was demonstrated that latency differs significantly based on the selected data center region and user location, implying that network performance should be considered when selecting a cloud provider. It further concluded that there was no consistent outperformance of one provider over the other under all operating conditions, thus highlighting the need for workload-specific evaluations.

A full multivocal literature review on the performance evaluation of Function-as-a-Service (FaaS) platforms was presented in [5]. A total of 112 academic and industrial studies were analyzed, and AWS Lambda was found to be the most evaluated cloud platform. It was also reported that many benchmarking methodologies have limited reproducibility and are heavily dependent on micro-benchmarks rather than realistic application workloads. The study in [6] examined the cost-performance efficiency of general-purpose virtual machine instances offered by AWS, Microsoft Azure, Google Cloud Platform (GCP), and Oracle Cloud Infrastructure (OCI). The results indicated that the processor architecture has a significant effect on system performance, and ARM-based instances tend to provide better price-performance ratios across a broad range of workloads.

The authors presented a study on AWS machine learning deployment services in reference [7]. The authors compared Amazon SageMaker, AWS Lambda, and Elastic Container Service (ECS) based on scalability, deployment flexibility, and operational cost. The experimental results indicate that there is no single deployment service that is optimal for all machine learning applications, and the deployment decision should be based on workload characteristics and operational requirements. The research [8] presented a comprehensive comparison of AWS and Microsoft Azure by evaluating their storage, networking, security, pricing, and machine learning services. In addition to the review of cloud service offerings, a survey was conducted to explore users' preferences. Results showed that Azure is often chosen for machine learning development due to its integration with Microsoft's ecosystem, while AWS provides a wider range of mature cloud services. However, the comparison was largely qualitative and not backed up by empirical performance benchmarking.

The paper [9] investigated the multi-cloud deployment options of microservice-based applications. Scalability, deployment efficiency, interoperability, security, and cost reduction were analyzed and compared. The results reveal that AWS and Azure offer similar scalability; however, there are differences in resource allocation efficiency and deployment management. One study [10] examined cloud-native MLOps solutions from AWS, Microsoft Azure, Google Cloud Platform, and Databricks. The study focused on model development, deployment automation, lifecycle management, and operational complexity. We found that the selection of a platform is mostly determined by the demands of the company, the ecosystem, and operational goals, and not by isolated features of performance.

Deep learning algorithms were presented for cloud service quality prediction in [11]. We accurately predicted future Quality of Service (QoS) metrics based on historical cloud performance measurements. The study demonstrates the potential of predictive analytics as intelligent decision support for cloud resource selection and the growing importance of quantitative cloud quality assessment. The work in [12] examined the portability and elastic scalability capability of cloud-native container technologies. We measured AWS Fargate, Azure Container Instances, and Google Cloud Run against a typical set of workloads to gauge their startup latency, execution dependability, runtime variability, and portability. The results reveal that the runtime behaviour still varies significantly, although the uniformity among container-based cloud systems is rising.

The study referenced in [13] conducted a comparative analysis of serverless pricing strategies. The assessment of AWS Lambda vs Azure Functions was based on execution cost and pricing methods. The results reveal that there are considerable variances in execution costs with different cloud providers and deployment areas. We also observed that the price efficiency is highly workload characteristic- and execution frequency-dependent.

There have been extensive comparisons between AWS and Microsoft Azure in previous studies; however, there are still several research limitations. First, evaluations have been mostly limited to specific cloud services (e.g., serverless computing, virtual machines, container platforms, and machine learning services) instead of considering cloud providers as complete ecosystems. Second, qualitative comparisons of features have been chosen over standardized empirical performance evaluations. Third, most studies have not incorporated multiple cloud quality metrics such as response time, availability, scalability, reliability, cost efficiency, and resource utilization into the Service Measurement Index (SMI) framework. Therefore, this paper presents an empirical evaluation of AWS and Microsoft Azure based on the important SMI performance metrics to help developers to select the cloud platform that satisfies their technical and operational requirements.

## III. Methodology

This work focuses only on IaaS (AWS) and PaaS (Microsoft Azure). Accordingly, the definition and measurement of all metrics in this paper are accomplished regarding IaaS and PaaS service providers. Some metrics of the chosen categories are user-dependent, while others are application-dependent. User-dependent metrics are defined based on the requirements of users, such as operability and transparency, while application-dependent metrics are defined based on the requirements of users [14]. The application-dependent metrics can be measured directly in the cloud services, while the user-dependent metrics can only be measured by users' usage as well as experience and desires [15], for example, using voting surveys. Because AWS and Microsoft Azure do not use the same identifying labels for the same functions as in Table 1, direct comparison is challenging. Measuring performance will be different since hardware and software architectures are different [16]. This makes the comparison more complicated.

**Table 1**: Some of the Labeling Functions in Comparison Between AWS and Azure

| AWS | Microsoft Azure |
|---|---|
| EC2 | Virtual Machine |
| VPC | Virtual Network |
| RDS | SQL Database |
| ELB | Traffic Manager |
| Route 53 | Windows Azure Name Resolution |
| EBS/S3 | Storage |
| Direst Connect | ExpressRoute |
| IAM with MFA | Multi-Factor Authentication |
| Security Group | End Points |
| SNS/SES | Service Bus |
| EC2ConfigService | VM Agent |
| SQS/Auto Scaling | Scheduler |
| CloudWatch/Auto Scale | Automation |

**Table 2:** Selected categories with their chosen attributes

| Category | | Selected Attribute(s) |
|---|---|---|
| 1 | Agility | Elasticity |
| 2 | Assurance | Availability |
| 3 | Financial | Ongoing cost |
| 4 | Security | Security capabilities, access control and privilege management, and physical and environmental security |

This work divides metrics into two categories: application-dependent and user-dependent. The latter defines user-dependent metrics. It is not an easy task to create a measurement platform to cover all possible metrics. Accordingly, this paper focuses on the following chosen categories with selected metrics, as shown in Table 2, which are application metrics. In order to make the measurements as simple as possible for users, important measurement metrics are selected and clearly defined, which are quantitative metrics; the selected categories and the related attributes are closer to each of them.

### Agility

The demand for the usage of an application is not a constant; there will always be a change in the needs of the users, which results in either an increase or decrease in demand. The developer must ensure their system meets stakeholders' changing needs. This concern is addressed through increasing the resources available for the application, also known as the application's scalability. It is defined as the ability for the application to function adequately on increasing loads [17]. If the system were to remain constant while increasing the load, it would crack under the number of requests, and no one would be able to use the application.

There are two different types of scaling available for the applications to grow. The first is known as vertical scaling or scaling up the application. This is where additional resources are provisioned for the current instance the application is running on. Here, it is just improving the capabilities of that instance. The system usually is required to be rebooted before these new resources are available to the application. The other type of scaling is known as horizontal scaling or scaling out the application. Unlike vertical scaling, where new resources are provisioned to the existing instance, horizontal scaling provisions an identical instance, which can be used to handle requests for the application. In this case, some sort of load balancer is required to distribute requests for the application across all the available instances [18]. An example of horizontal scaling would be launching another virtual machine with the same image/snapshot as the first virtual machine.

Cloud providers create transparency with the developers while keeping the details of their infrastructure to themselves [5]. For example, the user can save data to their servers, but they don't know which machine specifically contains that data. The full automation of cloud infrastructure has led to the rise of a concept known as elasticity [19]. This refers to the system's ability to scale both up and down to meet the current demand. This way, extra resources are not provisioned and wasted when server load is minimal. This concept is traditionally associated with cloud computing; for this to exist in the traditional sense, there would have to be upfront capital costs for the new resources.

The agility of cloud computing, more specifically its elasticity, is considered to be the reason for its sudden growth in the industry. This is due to its ability to meet the application's demand on-the-fly without wasting extra resources. As this is an important aspect of cloud computing, it is key that it is a metric that should be considered in the decision of any developer when choosing a cloud service provider. The elasticity of the cloud providers was taken into consideration for the evaluation, as it is considered to be one of their most important aspects. If the application cannot grow in a timely manner, the existing servers that support it may fail or negatively impact the experience of current users. This is an undesirable experience for both the developers and the users of the application. This means that the time it takes for new resources to be provisioned is the key point of interest with regard to the cloud providers' agility.

For this metric, the test will be to determine, from the point of time when new resources are requested, how long it takes for them to take effect. Some important assumptions were made for testing this metric. Only horizontal scaling was considered, as the interest is in how long it takes for the cloud providers to supply the available resources for the application. If vertical scaling were to be considered, it would have to include the time for the system to reflect these changes, like a reboot. Also, in industry horizontal scaling is more widely used for applications than vertical scaling. Another assumption made was that the provisioning time is consistent from the provider at varying times.

**Assurance**

Assurance is the ability to provide a service that will be working when needed. The most important attributes of assurance are the expected availability of the deployed service in the cloud. In public clouds, the responsibility for any associated problems with the infrastructure lies with the service provider. Providers are responsible for monitoring their resources, for example, servers, in order to be as stable as possible. However, there is no guarantee of providing independent availability of instances. In the case of unavailable instances, it needs to restart the instances' related applications on different servers [20]. There are other problems that users might face when dealing with cloud services, such as unexpected outages of the service. Such problems should be taken into account when dealing with or using cloud services.

Cloud services' owners provide two types of support options for users: a free best-effort option offering no Service-Level Agreement (SLA), a policy that governs the use of cloud services and guarantees; and premium commercial-grade options that provide users with SLA guarantees. In order to be effective, SLA specifications need to be designed so that they meet the customers' requirements at the relevant stage [21]. A meaningful approach to compare cloud providers regarding availability assurance can be accomplished by comparing the provided terms in SLAs that are offered by each provider as well as checking the outages of each provider in the past period of time. The result will give potential customers a hint about the availability of the compared providers, and it will make it very clear for customers in choosing the best applicable one for their applications.

Evaluation of SLAs is a challenging step for users since some cloud sellers try to obtain legal protection for their future business. Moreover, most cases offer users fewer assurances. This concludes that, in order to protect their future rights, users should be aware of some possible actions that might occur, such as server outages and data backups [9]. Providers reserve the right to change SLA terms in accordance with their agreement. Additionally, there is an amount of unscheduled monthly downtime acceptable under various SLA levels [22].

**Financial**

The financial aspect (price) is an effective factor that is taken care of by most users. Price evaluation alone will not lead to actual differences among the available cloud services because there are variations in the provided services and the platforms. Nevertheless, when using the price metric along with other measured metrics, pricing will be important, which might affect the choice of the provider. In general, price is a requirement function of the used resources, such as memory and CPU. There are two kinds of pricing strategies: package pricing and unpackaged pricing. In the first type, a price is charged for each characteristic, for example, charging per 500GB of memory and each used CPU. In the second type, price is offered for a predefined plan for some characteristics [14].

As a result of the increased competition among different providers, online tools have been designed to provide comparison measurements, such as CLOUDORADO [23]. However, some of the unused services are still included in the provided list, such as EC2 Micro t1.micro, which is not provided by AWS anymore. In addition, without looking for detailed characteristics of the used comparison metrics for each provider, the user will not recognize the real differences between the compared platforms. In order to achieve a price evaluation between the two chosen cloud providers, the comparison should be conducted with the same configurations for both providers or at least with close configurations, such as, for example, 1.7GB RAM in AWS and 1.75GB RAM in Microsoft Azure. This, in truth, will give the user a meaningful comparison and show any minor difference so that users will make the right decision in choosing the required provider.

**Security**

Security is the highest concern of cloud users. To achieve security goals, security measures should be available to users in order to measure the security level of the provided services. In fact, there is no application that is completely secure [24] . One of the most secure approaches is a combined level of security. However, any weakness in any one of these layers can cause a breakdown in security [3]. Users have different levels of visibility and control over the underlying infrastructure of cloud models (SaaS, PaaS, IaaS) that they can configure.

The security responsibilities differ from platform to platform. In IaaS, for example, in AWS EC2 infrastructure as a service, securing the underlying infrastructure, such as environmental security, virtualization security, and physical security, is the responsibility of the provider. The consumer's responsibility is the security of IT systems, which are the deployed instances. This means that the security responsibility of the customers is above the hypervisor layer, securing operating systems and applications as well as data. In PaaS, the security is a balance between providers and consumers. Securing the platform is related to the provider, while developing secure applications is the customer's mission [3].

However, in IaaS, the provider still has significant responsibilities for the provided cloud service regarding access to data, such as shared network devices and security devices like firewalls. The provider should deliver such security services and management. On the other hand, users should understand the offered security services in order to protect their data and applications that are deployed in the cloud.

Although some cloud vendors provide customers with the related services, including the ability to handle data processing with managed security [3], the offered security services knowledge differs among the adopted services, such as in IaaS, PaaS, or SaaS. Providing details of such security mechanisms will help users evaluate the offered services and realize if they meet their needs or not, or at least exploit the available security services to protect their data. The already provisioned security capabilities are different from provider to provider and from model to model. In order for consumers to select the best option, they should first know the offered security services in detail. Through the literature review, no direct criteria for security comparison have been found. Some of the adopted methods are based on users' feedback, such as using surveys.

In addition to this, some of the provided security services are with different labels, such as Security Group in AWS, while it is labelled as Endpoint in Microsoft Azure. Furthermore, the available comparison metrics are general, such as checking if the Firewall option is provided or not, while not comparing the entire feature details such as in [23]. The best way to achieve an effective comparison is by implementing a direct comparison with further details for each feature if there is a difference in the specifications.

## IV. Evaluation

**Agility**

The scalability of the cloud provider was measured by determining the time it takes to provision an instance from the moment it is requested. This was done through using the cloud provider's command-line interface (CLI). To accurately measure the provision time, software tools had to be developed specifically for each cloud provider. This is due to each provider having different APIs used to control their platforms. These differences go beyond just different parameters and endpoints. For example, provisioning an instance on Microsoft Azure was a synchronous blocking operation where control was not returned until an instance was provisioned, which made it trivial to determine the time, whereas Amazon Web Services (AWS) was an asynchronous non-blocking operation. This meant that to get accurate provision times for AWS, a polling mechanism was required. These details will be elaborated in their respective breakdowns.

To make a fair comparison between the capabilities of Microsoft Azure and AWS, we needed to consider comparable hardware specifications. As Azure and AWS tailor to different cloud models (platform vs. infrastructure), it is difficult to impossible to get identical systems on both providers. For the tests on AWS, the M3 Medium machine was used, whereas on Azure the Standard D1 was used. Table 3 presents a comparison of the specifications. Each of the cloud providers was tested on Ubuntu 14.04 LTS.

**Table 3:** Agility Platform Test Specifications

| | Microsoft Azure | Amazon Web Services |
|---|---|---|
| CPU Type | Intel Xeon E5-2660 | Intel Xeon E5-2670 |
| CPUs | 1 | 1 |
| RAM | 3.5 GB | 3.75 GB |

| | | |
|---|---|---|
| Storage Type | SSD | SSD |
| Storage Amount | 50 GB | 4 Gb |

Manually provisioning new instances in Microsoft Azure through their CLI required the development of a template. Azure uses templates to describe all the setup information of the system outside of the machine. Examples of information included in this template include resource groups, deployment regions, and access control information. The template also specifies the image used if a virtual machine is employed. The image can be either a base image like Ubuntu Server or a snapshot of an instance containing everything preconfigured. To accurately measure the provision time, timestamps were used upon requesting the instance to be provisioned and upon finishing the provisioning. The end was determined when control was returned to the testing suite developed. This is because the provision operation is a blocking call, meaning that control is not returned until the resources have been provisioned.

Manually provisioning new instances in AWS through its CLI is more straightforward compared to Azure. Setup preferences are passed with the provisioning operation as arguments. These preferences are similar to Azure, such as security groups, deployment image, and access control. Unlike Azure, though, the provision operation on AWS for a new virtual machine is non-blocking. This means that the timing information would have to be acquired differently. AWS records the time that the instance started to be provisioned. This was used, as it is more accurate than the starting timestamp, as it removes the latency induced through the network by about a second. The time the instance was provisioned was determined through polling for information about the new instance. AWS provides states for the instance, which specify when it is being provisioned and when it is running. The problem with this polling mechanism is that it introduces a deviation in the result, as it can't continuously check. This deviation is considered to be within a couple of seconds, so it is considered acceptable to achieve a closely accurate result.

With the specified tools and conditions, 100 tests were performed on each of the cloud providers to determine the relative provisioning time. Due to network latency, a couple of outliers were removed from the data set due to being well beyond the standard deviation of that provider. Table 4 displays the results of the provisioning testing. There is a significant difference in provision time between the cloud providers with comparable specifications. On average, Microsoft Azure took 3 more minutes to provision a new instance than AWS.

**Table 4:** Agility Platform Test Results

| Provision Time | Microsoft Azure | Amazon Web Services |
|---|---|---|
| Mean | 3.6 Minutes | 43 Seconds |
| Best Case | 3.3 Minutes | 49 Seconds |
| Worst Case | 3.9 Minutes | 40 Seconds |

**Assurance**

In order to provide a meaningful result, the comparison will be achieved in two parts. The first part will present directly provided SLAs for the most important services, such as storage and backup, as in Table 5. The second part will compare the outages that each provider faced previously and the availability of the chosen providers during the implementation part of this paper. The following table shows a comparison of some provided SLAs' terms between AWS and Microsoft Azure.

**Table 5:** A comparison of provided SLAs' terms between AWS and Microsoft Azure

| Characteristic | AWS | Azure |
|---|---|---|
| Provide Service Level Agreement | Yes [25] | Yes |
| Provide a Service Credit for Region Unavailable | Yes, EC2 and EBS [15] | - |
| Active Directory | - | 99.9% [26] |
| API Management | - | 99.9% [26] |
| App Service | - | 99.95% [26] |
| SLA for Mobile Apps | - | 99.9% [26] |
| Backup | - | 99.9% [26] |
| Storage [27] | 99.9% (S3) | 99.99% |
| SQL Database [26] | (RDB) 99.9% | 99.9% |
| VPN Gateway | Yes [28] | 99.9% [26] |
| Cloud services and virtual machines [27] | 99.5% | 99.5% |
| Traffic Manager [27] | 100% (Route 53) | 99.99% |
| CDN [26] | 99.9% (CloudFront) | 99.9% (CDN) |

Early studies reported considerably higher outage frequencies for both cloud providers. For example, in 2014 AWS EC2 experienced approximately 2.41 hours of downtime (99.9974% availability), whereas Microsoft Azure experienced about 39.77 hours of downtime across multiple outages. Similarly, AWS storage services experienced approximately 2.69 hours of downtime compared with 10.97 hours for Azure storage. In 2015, AWS recorded approximately 2.5 hours of downtime, while Microsoft Azure experienced approximately 10.8 hours [26][27]. Although these results suggested that AWS provided higher availability during that period, they no longer represent the current state of cloud computing. Recent studies indicate that both AWS and Microsoft Azure have substantially improved their service reliability through multi-region architectures, availability zones, automated failover mechanisms, and continuous infrastructure enhancements. Modern outages are typically localized to specific regions or services rather than affecting the entire cloud platform, and current research emphasizes resilience, fault tolerance, and rapid recovery instead of annual downtime comparisons [29]. The availability duration of the two providers in consecutive 30-day periods is shown in Table 6.

**Table 6:** AWS and Microsoft Azure Availability Comparison in 30 days

| **AWS** | | | | | |
|---|---|---|---|---|---|
| Service Name | Availability/Month | Outages | Regions | Downtime/Region | Total/ hours |
| EC2 | 100% | 0 | 10 | - | - |
| S3 | 100% | 0 | 10 | - | - |
| **Microsoft Azure** | | | | | |
| VMs | 99.9997% | 14 | 13 | 16.32 mins | 3.54 |
| Cloud Storage | 100% | 0 | 13 | - | - |

Although the two providers offer similar SLAs, detailed specifications differ and are unavailable to the public. However, the result will give a general idea to users about what is provided. In addition to the previous statistics, this project has not encountered any outages.

**Financial**

Pricing comparison is not an easy task since the cost is subject to change based on periodic updates. However, the comparison can assist users in choosing the required service based on affordability or desire. In this paper, the package price plan is adopted in the evaluation process.

In order to provide a meaningful price comparison to users and to show the effect of application requirements, hosted place, and configuration on the cost [30][31], the comparison procedure will be applied with two different criteria on both AWS and Microsoft Azure. In the first comparison, a real-world case is considered where the costs are determined to host the solution on each platform. The comparison results can be seen in Tables 7 and 8. The second comparison looks into the change in prices depending on the region [23] used. In each of the assumptions, the selected options are as close as possible for the purpose of comparison, and the specifications are chosen as similar as possible in both Windows Azure and AWS.

**Comparison 1**

This comparison investigates the cost for deploying a web application with certain requirements. For this comparison, the following assumptions are made:

- SQL Server (On-Demand License Included RDS) database, Enterprise Edition, 2 GB or more of RAM, 2 CPU cores
- 5 servers with 1 CPU and at least 2 GB of RAM
- Elastic IP addresses
- The hosted area is in the eastern US.

**Table 7:** Windows Servers on AWS Assigned Cost

| **Chosen Service** | **No.** | **Price/Hour** | **Period/Month** | **Total Cost($)** |
|---|---|---|---|---|
| **Windows with SQL Enterprise** (db.r3.2xlarge) | 1 | $5.810 | 720 | 4183.2 |
| – Memory (GB) | 15 | | | |
| – vCPU | 2 | | | |
| **Windows with SQL Web** (m4.4xlarge) (for websites) | 5 | 1.844 | 720 | 6638.4 |
| – Memory (GB) | 2 | | | |
| – vCPU | 1 | | | |
| – Elastic IP address (*) | 1 | 0 | 0 | 0 |
| Total | | | | 10821.6 |

(*) $0.005 per additional Elastic IP address associated with a running instance per hour on a pro rata basis

**Table 8:** Windows Servers on Microsoft Azure Assigned Cost

| **Chosen Service** | **No.** | **Price/Hour** | **Period/Month** | **Total Cost($)** |
|---|---|---|---|---|
| **Windows SQL Server Enterprise** (standard) | 1 | 2.225 | 720 | 1602 |
| – Memory (GB) | 14 | | | |
| – vCPU | 2 | | | |
| **Windows with SQL Web** (basic) (for websites) | 5 | 0.580 | 720 | 2088 |

| | | | | |
|---|---|---|---|---|
| – Memory (GB) | 1.75 | | | |
| – vCPU | 1 | | | |
| **Instance-level IP Addresses** | 1 | | 2.88 | 2.88 |
| **Total** | | | | 3690 |

**Comparison 2**

The main goal of this evaluation part is to compare the price in different hosted areas. To achieve this, the following assumptions are adopted:

- vCPU: 1
- Storage Size (GB): 500 HDD
- RAM: AWS=3.75GB; Microsoft Azure=3.5GB
- Transfer Out (GB): 100, GET Requests (KB): 10, PUT Requests (KB): 500

The result of this comparison can be seen in Table 9.

**Table 9:** Cost comparison between AWS and Microsoft Azure regarding the selected area

| Area | Price per month ($) | |
|---|---|---|
| | AWS | Microsoft Azure |
| North USA | 98.24 | 75.06 |
| South USA | 163.40 | 106.56 |
| Europe | 130.56 | 75.06 |
| Asia | 107.56 | 72.18 |
| Australia | 126.96 | 99.36 |

**Security**

The best way to evaluate the security services offered by cloud providers is a direct comparison of their security services and tools provided on their platform, which are available to developers. A direct comparison of the most important security attributes has been carried out on the two providers, Microsoft Azure and AWS web services. The comparison will be focused on the most important security capabilities that are provided by the two mentioned providers [32]. In the case of any difference in the provided security features, the difference is presented in the comparison as Table 10 shows. This comparison has been accomplished by reviewing most of the published documents by each provider and other academic resources. From the comparison result as shown in Table 10, it is clear that both providers provide some common security services, such as Firewall service. However, the provided security services are not identical, and some of the provided security services are labelled differently, such as Security Group in AWS, while it is labelled as Endpoint in Microsoft Azure.

**Table 10:** Comparison of security between Azure and AWS

| Security Capability | AWS | Microsoft Azure |
|---|---|---|
| Isolation of Hypervisors [33] | Yes | Yes |
| Firewall [33][34] for Inbound and Outbound | Yes (Security Group) SSH, RDP, or HTTP | Yes (Endpoints only) (**) |
| Packet filtering [35][33] | Yes (firewall) | Yes |
| API access [35][36] | HTTPS within an SSL/TLS | Client certificates |
| Data access control( Secret Keys) [35] | User access policies | A secret key for each user's account access |
| Remote access to individual cloud servers (SSH, RDP) [34] | Yes | Yes |
| VPN access and identity-based access management and rules [35][33] | Yes | Yes (dynamic client-based VPN access) |
| Mutual authentication [35][33] | Yes | Yes |
| Shared Private Cloud network [33][34] | Yes(VPC) | Yes (only VPN) |
| Private IP [35][36] | Yes | Yes |
| Multiple IP per cloud server [35][36][23] | Yes | No |
| IPv6 [23] | No | No |
| Secure Execution Using IPsec [34] | Yes | Yes |
| VLAN per customer [23] | Yes | No |
| VPN [35][36] | Yes | Only to create a virtual network |
| PCI DSS [35][36] | Yes | Yes |
| FISMA Compliant [23] | Yes | Yes |
| Encrypted storage [35] | Yes | Yes |
| Resource usage monitoring (CPU, RAM, disc, network, ...) [35][36] | Yes | Yes |
| Compliance [36] | Yes | Yes |

(**) Cannot control internet traffic, does not provide any grouping, and provides VPN gateway capabilities.

## V. Conclusion And Future Work

Although there are differences in the provided configurations and performance among the two providers, both cloud providers provide comparable performance on similar systems. The evaluation process shows that AWS has a significant edge in terms of scalability. Regarding availability, no significant outages were observed during the work on this project. This indicates that the availability of both providers is very high. Despite there is a similarity between the two providers regarding the offered services, in some instances, Microsoft Azure offers a more desirable or lower price, while AWS offers more options at a higher price, which is reflected in the performance. This means that the cost is diverging and would be greatly changed when choosing different features.

The two providers provide security tools and configurations, and users need to be aware of the following security issues when dealing with any cloud service: which security standard is followed by each provider, communication protocols that are used to communicate with other data centers, clear text administration protocols, the ability to monitor communication in and out of the cloud as well as within the cloud, the ability for end-user devices that can download data from the cloud, and how to configure such provided services.

For this evaluation, there are more challenges and evaluation criteria that can be addressed in future work. For the elasticity measurement, it was applied on a single day throughout the day in a single region. It is possible that at different times of day or different days of the week in different regions the provision time could vary. One criterion that was not investigated in the study was the usability of the cloud provider. This can be considered to be the most important criterion above the functional abilities of the platform, as the developer would be more comfortable developing on a platform that is easier to use and well designed. This is a difficult criterion to specify in a quantitative manner, as usability is a very qualitative and subjective measurement.

## References

[1] Peter Mell And T. Grance, "The NIST Definition Of Cloud Computing," Pp. 1–3, 2011.
[2] Z. Haoxing And C. System, Cloud Computing: Concepts, Technology & Architecture, 4th Ed. Pearson Education, 2024.
[3] F. Khoda Parast, C. Sindhav, S. Nikam, H. Izadi Yekta, K. B. Kent, And S. Hakak, "Cloud Computing Security: A Survey Of Service-Based Models," Comput. Secur., Vol. 114, Pp. 1–18, 2022.
[4] F. Palumbo, G. Aceto, A. Botta, D. Ciuonzo, V. Persico, And A. Pescapé, "Characterization And Analysis Of Cloud-To-User Latency: The Case Of Azure And AWS," Comput. Networks, Vol. 184, No. November, 2021.
[5] J. Scheuner And P. Leitner, "Function-As-A-Service Performance Evaluation: A Multivocal Literature Review," J. Syst. Softw., Vol. 170, P. 110708, 2020.
[6] J. Tharwani And A. Purkayastha, "Cost-Performance Evaluation Of General Compute Instances: AWS, Azure, GCP, And OCI," Int. J. Comput. Trends Technol., Vol. 72, No. 11, Pp. 248–255, 2024.
[7] R. Bagai, "Comparative Analysis Of AWS Model Deployment Services," Int. J. Comput. Trends Technol., Vol. 72, No. 5, Pp. 102–110, 2024.
[8] S. Hameed, S. H. R. Bukhari, S. U. E. Abiha Rizvi, M. Tahir, And F. Aadil, "A Comparative Analysis On The Services Offered By Amazon Web Services And Microsoft Azure," Concurr. Comput. Pract. Exp., Vol. 37, No. 21–22, 2025.
[9] O. Article, "Multi-Cloud Deployment Strategies For Microservices: A Comparative Study Of AWS And Azure," 2025.
[10] S. Moreschini, D. Hastbacka, And D. Taibi, "Comparing Cloud-Native Mlops Platforms: AWS, Azure, GCP, And Databricks," IEEE Softw., Vol. 43, No. 2, Pp. 136–142, 2026.
[11] X. Guerron, M. Fernández-Diego, S. Abrahão, E. Insfran, And · Sira Vegas, "Building And Validating Deep Learning Models For Forecasting The Quality Of Cloud Services," 2026.
[12] D. T. T. Nguyen And T. De Souza-Daw, "Evaluating Multi-Cloud Portability And Elastic Scaling Across Serverless-Container Platforms," Int. J. Cloud Appl. Comput., Vol. 16, No. 1, Pp. 1–22, 2026.
[13] N. Ekwe-Ekwe, "Priceless: An Examination Of Serverless Functions-As-A-Service (Faas) Pricing Models," No. 1, 2026.
[14] G. B. D. P. Vidyarthi And A. Authors, "A Framework For Selection Of Best Cloud Service Provider Using Ranked Voting Method," In IEEE International Advance Computing Conference (IACC), 2014, Pp. 831–837.
[15] Y. A. R. Ashibani, "Enhancing Multi-Factor Authentication Efficiency For Mobile Applications Using Machine Learning And Behavioural Biometrics," In International Conference On Intelligent And Fuzzy Systems, 2025, Pp. 154–161.
[16] W. Van Winkle, "Microsoft Azure Vs. Amazon Web Services: Cloud Comparison," Pp. 1–6, 2015.
[17] A. Algiffary And T. Sutabri, "Harnessing The Power Of Distributed Systems For Scalable Cloud Computing A Review Of Advances And Challenges," Indones. J. Comput. Sci., Vol. 12, No. 2, Pp. 284–301, 2024.
[18] L. M. Vaquero, L. Rodero-Merino, And R. Buyya, "Dynamically Scaling Applications In The Cloud," Comput. Commun. Rev., Vol. 41, No. 1, Pp. 45–52, 2011.
[19] R. K. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph And And M. Z. Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, "Above The Clouds: A Berkeley View Of Cloud Computing," 2009.
[20] T. Benson, S. Sahu, A. Akella, And A. Shaikh, "A First Look At Problems In The Cloud," 2nd USENIX Work. Hot Top. Cloud Comput. Hotcloud, 2010.
[21] N. Subramanian And A. Jeyaraj, "Recent Security Challenges In Cloud Computing," Comput. Electr. Eng., Vol. 71, No. July, Pp. 28–42, 2018.
[22] D. Data, "Comparing Public Cloud Service Level Agreements," Dimens. Data, 2013.
[23] Cloudpricecheck, "Cloud Pricing Comparison." [Online]. Available: Https://Cloudpricecheck.Com/. [Accessed: 27-Apr-2026].
[24] Y. Ashibani And Q. H. Mahmoud, "A Behavior-Based Proactive User Authentication Model Utilizing Mobile Application Usage Patterns," In 32nd Canadian Conference On Artificial Intelligence, 2019, Pp. 284–295.
[25] Amazon Web Services, "Amazon EC2 Service Level Agreement." [Online]. Available: Https://Aws.Amazon.Com/Ec2/Sla/. [Accessed: 08-Oct-2025].
[26] Microsoft Azure, "Service Level Agreements (Slas) For Microsoft Online Services." [Online]. Available: Https://Azure.Microsoft.Com/En-Us/Support/Legal/Sla/. [Accessed: 18-Oct-2025].

[27] Techtarget, “AWS Vs. Azure Face-Off: Cloud Costs, Commitments And Slas.” [Online]. Available: Https://Www.Techtarget.Com/Searchcloudcomputing/. [Accessed: 08-Oct-2025].
[28] Amazon Web Services, “Amazon Virtual Private Cloud (Amazon VPC) Faqs.” [Online]. Available: Https://Aws.Amazon.Com/Vpc/Faqs/. [Accessed: 12-Dec-2025].
[29] O. Bibartiu, F. Dürr, K. Rothermel, B. Ottenwälder, And A. Grau, “Availability Analysis Of Redundant And Replicated Cloud Services With Bayesian Networks,” Qual. Reliab. Eng. Int., Vol. 40, No. 1, Pp. 561–584, 2024.
[30] Amazon Web Services, “Amazon Relational Database Service (Amazon RDS) Pricing.” [Online]. Available: Https://Aws.Amazon.Com/Rds/Pricing/. [Accessed: 14-Jan-2026].
[31] Amazon Web Services, “AWS Pricing Calculator.” [Online]. Available: Https://Calculator.Aws/. [Accessed: 14-Jan-2026].
[32] J. A. Mahatme And M. R. Satpute, “A Comprehensive Review On Cloud Computing: Challenges, Architectures, And Future Directions,” Iarjset, Vol. 12, No. 7, Pp. 285–295, 2025.
[33] Network Computing, “Cloud Storage Security: AWS Vs. Azure.” [Online]. Available: Https://Www.Networkcomputing.Com/. [Accessed: 20-May-2026].
[34] Network Computing, “Comparing Iaas Providers.” [Online]. Available: Https://Www.Networkcomputing.Com/. [Accessed: 24-May-2026].
[35] Amazon Web Services, “AWS Security Services And Features.” [Online]. Available: Https://Aws.Amazon.Com/Security/. [Accessed: 25-May-2026].
[36] Microsoft Learn, “Azure API Management Documentation.” [Online]. Available: Https://Learn.Microsoft.Com/Azure/Api-Management/. [Accessed: 25-May-2026].