International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

Extensive Experience in Aws Services in Develop and Deploying and Highly Available, Scalable and Fault Tolerant Systems

Prabu Ravichandran,
Sr. Data Architect, AWS, Amazon, Raleigh, NC, USA
Prabu.ravichandran07@gmail.com

ABSTRACT

Distributed systems' reliability and high availability have long been essential concerns. If you want to keep your customers happy and keep money coming in, you need to make sure your cloud services are always available and trustworthy. There have been numerous suggestions for improving the cloud's availability and reliability, but no thorough studies that address the whole issue have been carried out. A system's fault-tolerance level determines how well it can continue operating even if some part of it fails. We need a lot more software choices, and mission-critical systems must be up at all times. To buy something, for instance, we might like to check out an online store. It doesn't matter if it's 3 in the morning on a holiday or 9 in the morning on a Monday; what matters is that the site is accessible and prepared to take our payment. Many companies can't afford to fail if these expectations aren't met. Thousands of dollars could be lost for every minute that an active e-commerce site is down, even with the most careful estimations. Companies and groups invest much in developing software systems that can deal with errors for reasons like these. The infrastructure offered by Amazon Web Services (AWS) is perfect for developing software systems that can withstand failures. Having said that, this feature is not exclusive to our platform. Any platform can host a fault-tolerant software system if enough effort and time are put into it. Amazon Web Services (AWS) stands out from the competition because it lets you construct resilient systems with little to no initial investment and human intervention.

Key words: AWS, Cloud, Scalable, Fault Tolerant system.

1. INTRODUCTION

In the past, companies would build their own applications from the ground up, sometimes using third-party components or platforms. Not long ago, this was the norm. On the other hand, a new era began with the introduction of service-oriented architecture (SOA), which allowed applications to delegate some tasks to third-party services that were already in place [1]. Companies need to allocate more resources, both in terms of time and money, in order to scale up their information technology infrastructures in order to satisfy the ever-evolving demands of their businesses. However, while it is true that achieving this goal through one's own premises and investments is a cost-effective option, it also hinders enterprises from maximising their resource utilisation [2]. Companies have been compelled to look for new alternative technological solutions as a result of these issues. Cloud computing is a relatively new technology that aims to increase processing capacity so that millions of instructions per second can be executed. Currently, the term "cloud computing" and the services it provides are highly regarded in the IT industry.The provision of information technology and computing resources through the network is a relatively new development in the field of information technology that can be regarded as a paradigm shift. The 2009 and 2011 revisions to the National Institute of Standards and Technology's definition of cloud computing are among the most famous and generally agreed upon. According to this definition, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (such as networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [3]. Users are able to tap into a common pool of computing resources through the cloud computing architecture. New developments in cloud computing are amplifying the trend towards cloud virtualization. To rephrase, cloud computing's services are a major development in the direction of the utility computing paradigm's actualization [4]. Users are able to access services even if they are hosted in a different location or supplied in a different manner, according to this computing model.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

Computing technologies have evolved over the years, with examples including cluster and grid computing, virtualization, service-oriented computing, and more. Cloud computing, on the other hand, is a relatively new phenomenon that is currently grappling with the absence of thorough standards and solutions [5-8]. Consequently, cloud business models and technologies bring with them a number of important challenges, such as load balancing [9-10], security, energy efficiency, workflow scheduling [10-15], data/service availability, licence management, data lock-in, and API design [16-18]. Concerns about service reliability and high availability (HA) are among the most important ones when it comes to cloud computing. A system's reliability can be defined as the probability that it will remain operational throughout a given time frame without experiencing any faults. However, the likelihood that a system is available and operating properly at time 't' is what we mean when we talk about its availability [19-20]. Maintaining client trust and avoiding revenue losses due to service level agreement (SLA) violation fines are both made impossible without high availability (HA) in cloud services. Businesses and government agencies worldwide have been increasingly interested in cloud computing environments in recent years as a means to support mission-critical technologies. [21-29] However, cloud services' unreliability and low availability is quickly becoming an issue. Cloud service failures and an availability rate of about 99.91% have resulted in losses of about $285 million per year, according to studies.[30-31]

2. AMAZON MACHINE IMAGES

Amazon Elastic Compute Cloud (EC2) is a web service provided by Amazon Web Services that enables the construction and hosting of software systems through the provision of computing resources, namely as server instances. Amazon Elastic Compute Cloud (EC2) is an excellent entry point for those who are new to building apps on Amazon Web Services. A reliable and durable system can be built using many instances and two of the EC2 supplemental services, Auto Scaling and Elastic Load Balancing. A initial impression might be that an instance of Amazon Elastic Compute Cloud is no different from a traditional server. Linux, Windows, and OpenSolaris are among the most popular operating systems utilised by instances in Amazon Elastic Compute Cloud. This allows for the accommodation of software that is interoperable with those OSes. Because Amazon EC2 instances have IP addresses, you can use the standard techniques for connecting to distant machines, including SSH or RDP.

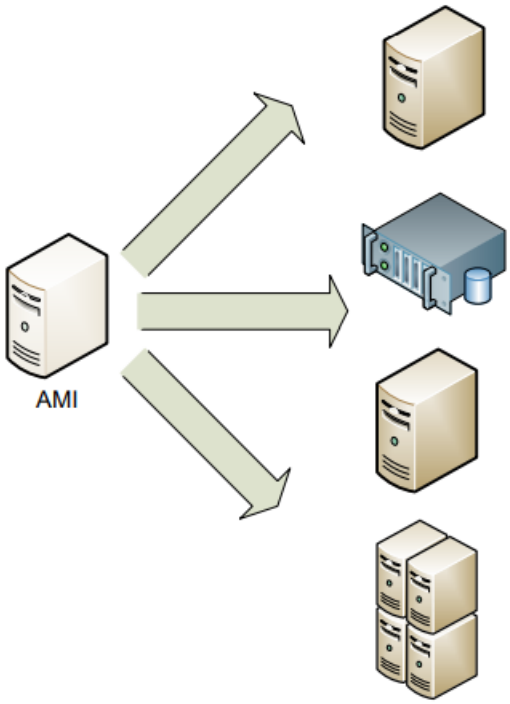

Figure 1 - Amazon Machine ImageOne can describe your service instances using an AMI, or Amazon Machine Image. The operating system, application server, and programmes are all part of the software configuration that is delivered to an instance type using this template. Similar to hardware archetypes, the amount of RAM and number of CPUs required by your application determine the instance type you choose in Amazon Elastic Compute Cloud (EC2) when creating an account. The following diagram shows the link between AMIs and the various instance types of server resources that can be created using them.

Many AMIs with typical software setups are published by Amazon. Not only that, but a number of AWS developers have released their own unique AMIs. You may locate all of these AMIs on the AWS website's Amazon Machine Image resources page2. However, if you want to construct AWS apps that can withstand failure, you need start by compiling your own AMI library. At minimum, one AMI that you have developed should make up your application. Then, starting your application is as easy as launching the AMI.

If your app runs on the web, for instance, your application manifest file (AMI) should include the web server configuration, any related static material, and the code for any dynamic pages. Internet Information Server (IIS) and Apache are two popular web servers. You can also configure your AMI to run instances with all the required software components and content pre-installed. This might be achieved by running a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

bootstrap script. That is why your web server and application can start taking requests the moment the AMI is launched. After you've made an AMI, it's easy to replace an unhealthy instance; all you have to do is begin a new instance using the AMI as a blueprint.

2.1 Elastic IP Addresses

Public IP addresses, known as Elastic IP Addresses, can be "routed" to any instance of Amazon Elastic Compute Cloud (EC2) in a given region. To make it easier to build fault-tolerant systems, these addresses are associated with an AWS account instead of an instance or the length of time an instance is running. If an instance fails, an elastic IP address can be easily transferred to a new instance. Elastic IP addresses, like Amazon Elastic Block Storage volumes and all other EC2 resources, can be programmatically accessed through the AWS Management Console or the API.

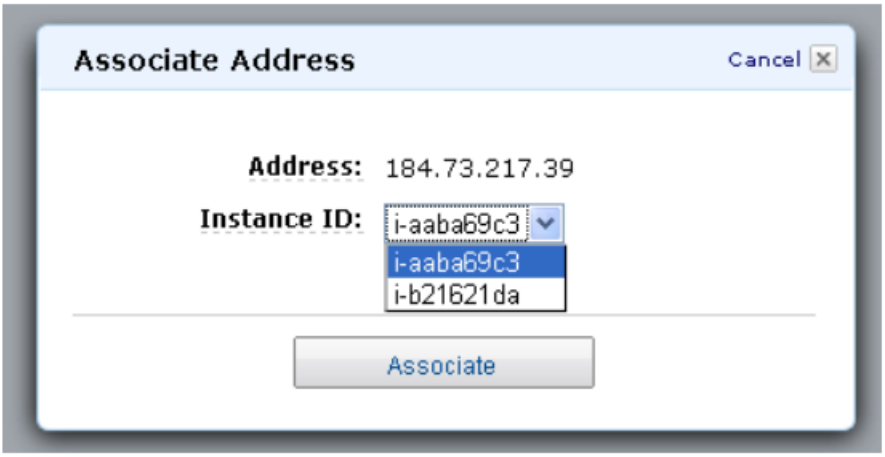

Figure 2 - Elastic IP addresses

3. FAILURES CAN BE USEFUL

Although it's not always easy to accept, the truth is that the majority of software systems will eventually become obsolete. This might be attributable to a combination of the following factors:

There will be memory and resource leaks in the software. All software, whether you've created it yourself or rely on it (such as operating systems, device drivers, and application frameworks), falls under this category.

Fragmentation of file systems over time will have an effect on performance.

Physical deterioration is inevitable with hardware, especially storage systems.

Although these issues can be partially reduced through disciplined software engineering, in the end, even the most advanced software system relies on several components (such as the operating system, firmware, and hardware) over which it has no influence. At some point, your application's availability will be disrupted due to a failure that was caused by a mix of hardware, system software, and your own software. There are practical and economical constraints that limit the aggressiveness of hardware maintenance and servicing in a traditional IT system. Amazon Elastic Compute Cloud, on the other hand, lets you terminate and re-create resources whenever you need them. At regular intervals, additional server instances can be added to an application that fully utilizes the AWS infrastructure. This will make sure that your system is not negatively impacted by any possible degradation. To revive this resource, you are effectively leveraging an event that would normally be seen as a failure, like a server shutdown. According to this point of view, the actual definition of an AWS application should focus on the service it provides to clients rather than the server instance(s) that comprise it. According to this way of thinking, the server instances are no longer important and can be thrown away.

3.1 Auto Scaling

For any application to be well-designed and run on the AWS platform without errors, the idea of automatically adding and adjusting computing resources is essential. With Auto Scaling3, you have a tremendous tool at your fingertips. Using Auto Scaling, you may effortlessly increase or decrease the capacity of your Amazon EC2 instance. When deciding

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

whether to use more or fewer server instances, you can set criteria like:

1. Start new server instances or stop existing ones when the count reaches a specific threshold. Launch or terminate server instances when their resource use exceeds or falls below a specified threshold, whether it be CPU, network, or disk. The Amazon CloudWatch service will gather these metrics since it keeps tabs on Amazon EC2 instances. With Auto Scaling, you can easily launch new server instances whenever you need them, even when you terminate an existing one. With Auto Scaling, you can create more instances as needed to handle growing workloads, and they will be automatically terminated when they are no longer needed.

4. ELASTIC LOAD BALANCING

To evenly distribute incoming traffic to your application, Elastic Load Balancing utilises many Amazon EC2 instances, an AWS offering. The process of Elastic Load Balancing begins with the assignment of a DNS host name and continues with the assignment of a group of Amazon Elastic Compute Cloud instances to process requests made to this host name.

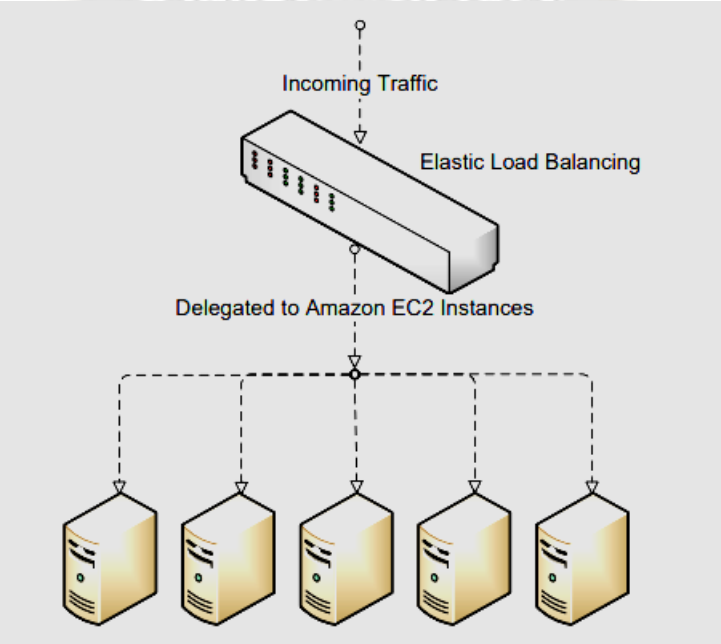

Figure 3 - Elastic Load Balancing

Elastic Load Balancing is able to identify low-health instances inside the Amazon Elastic Compute Cloud (EC2) pool and intelligently redirect traffic to those instances. This process is repeated until all instances that caused harm have been restored. Elastic Load Balancing lets you utilise a single DNS name for addressing, and Auto Scaling makes ensuring that there are enough running Amazon Elastic Compute Cloud instances to handle incoming requests. The two features work hand in hand and are a perfect suite.

4.1 Regions and Availability Zones

Distributing your application on a regional basis is another crucial step in gaining improved fault tolerance. Make sure your app is safe by running it in many datacenters across different locations in case any of Amazon Web Services'sdatacenters go down. Amazon Web Services is accessible from several locations. Data storage, instance operation, queue establishment, and database instantiation can all be done according to your specifications when utilising AWS. There are five regions that make up AWS infrastructure services, including Amazon Elastic Compute Cloud (EC2). These regions are Northern Virginia (US East), Northern California (US West), Ireland (EU), Singapore (Asia Pacific), and Japan (Asia Pacific). One minor variation in Amazon S3's region structure is the US Standard. This includes datacenters everywhere from Northern California in the US to Ireland in the EU to Singapore and Japan in Asia Pacific and beyond. Different regions have different availability zones (AZs). In a Region, there are several physical locations called Availability Zones. These zones are designed to provide low-latency, cost-effective network access to other zones in the same region, and they can also withstand failures in other zones. To safeguard your applications from a potential (albeit improbable) failure that impacts a whole Availability Zone, it is recommended to

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

create instances in distinct zones. Geographically distributed and including many areas or nations, regions are made up of one or more Availability Zones. Each Amazon Elastic Compute Cloud (EC2) region is guaranteed an availability rate of 99.95% according to the service level agreement.

5. BUILDING MULTI-AZ ARCHITECTURES TO ACHIEVE HIGH AVAILABILITY

One way to ensure your application is always available is to distribute it across different Availability Zones. To create a multi-site solution, it is possible to deploy redundant instances to different Availability Zones for each layer of an application, such as the web, application, and database. Our aim is to ensure that every application stack is replicated in at least two separate Availability Zones. Utilizing Elastic Load Balancing can further increase fault tolerance while reducing the need for manual intervention.

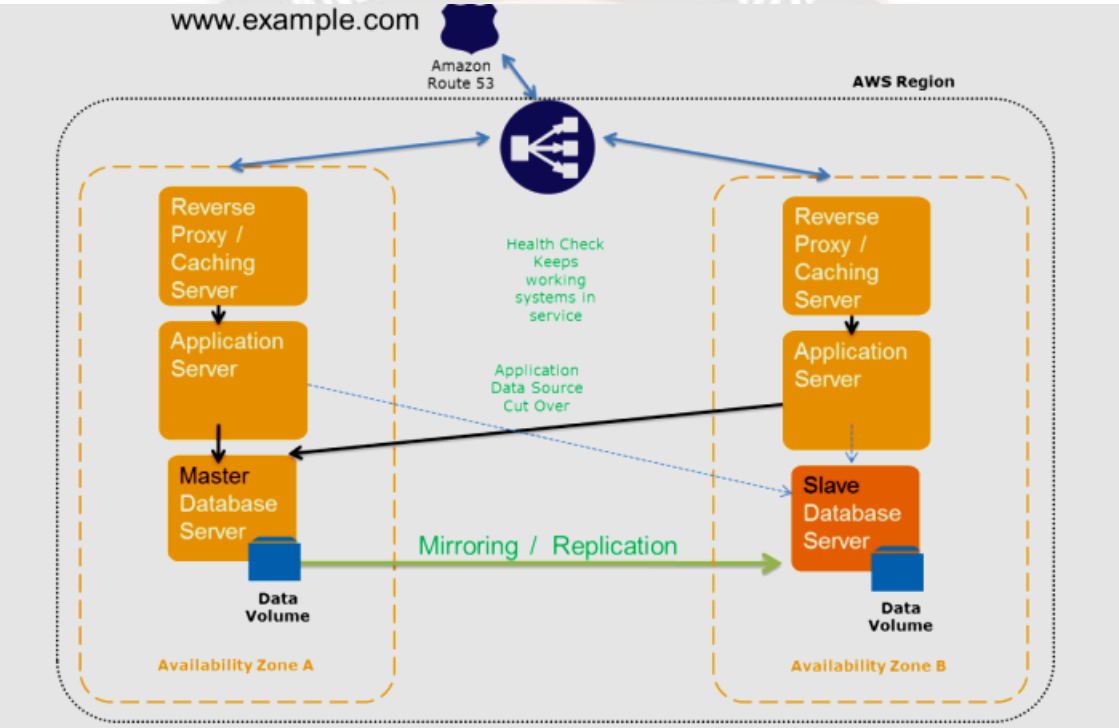

Figure 4: Employ Multiple Availability Zones and Elastic Load Balancers.

Installing an Elastic Load Balancer behind your compute instances is one approach to enhance fault tolerance. This is because, in addition to limiting traffic to only stable Amazon EC2 instances, the load balancer may automatically divide it across many instances and Availability Zones. You can equally divide incoming application traffic among Amazon Elastic Compute Cloud instances situated in one or more Availability Zones with the help of Elastic Load Balancers. Instance health can be detected using Elastic Load Balancing on Amazon Elastic Compute Cloud. It stops routing traffic to unhealthy Amazon EC2 instances the moment it finds them. Instead, it distributes the workload across the remaining instances that are in good health. Elastic Load Balancing will redirect requests to available instances in other availability zones in the event that all of your Amazon EC2 instances in one zone fall down. Make sure you have instances configured in more than one availability zone. Once the original Amazon EC2 instances are back to normal, it will start balancing the load again.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

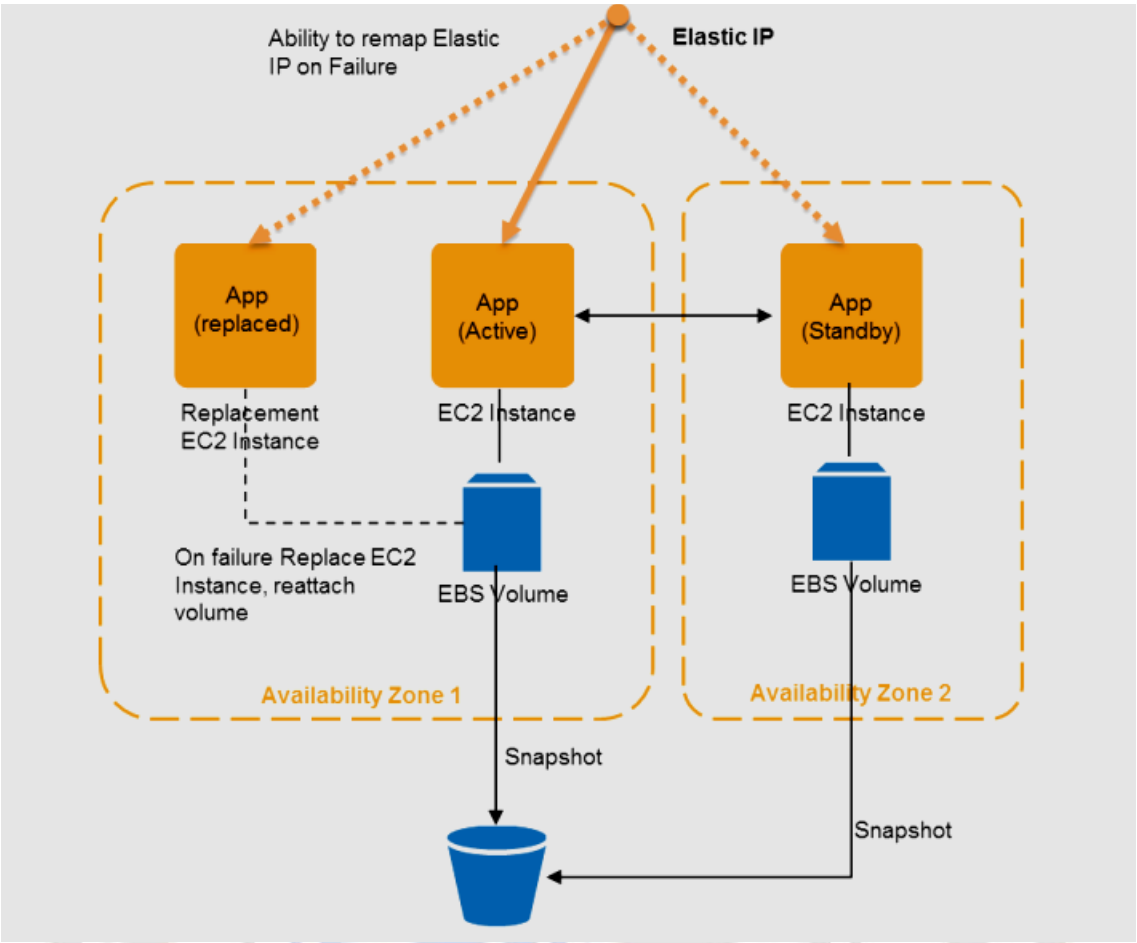

Figure 5: Harness the Power of Elastic IPs and MAZs

In an AWS Region, Auto Scaling makes it simple to automate the addition or removal of capacity by coordinating across multiple Availability Zones. Two database solutions provided by AWS, SimpleDB and Amazon Relational Database Service (Amazon RDS), may make managing a multi-site system easier and cheaper.

6. FAULT-TOLERANT BUILDING BLOCKS

Amazon Elastic Compute Cloud (EC2) and its capabilities allow you to build and launch apps on a robust and affordable platform. Still, they're not the only thing that Amazon Web Services offers. A wide variety of Amazon Web Services products are available for integration with application development workflows. You can improve your own apps' fault tolerance by making use of these web services, which are inherently fault-tolerant.

6.1 Amazon Simple Queue Service

An extremely dependable distributed messaging system, Amazon Simple Queue Service (SQS) can support your application's fault tolerance. You can store messages in queues you construct. Since each queue is defined as a URL, any server with Internet connection can access them, subject to the queue's connection Control List (ACL). With Amazon Simple Queue Service (SQS), you can make sure that your queue is available at all times; messages sent to a queue are kept for four days (or until your application reads and deletes them). Below is an example of a canonical system architecture that makes use of Amazon Simple SQL.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 November 2023* ***Revised:*** *12 December 2023* ***Accepted:*** *30 January 2024*

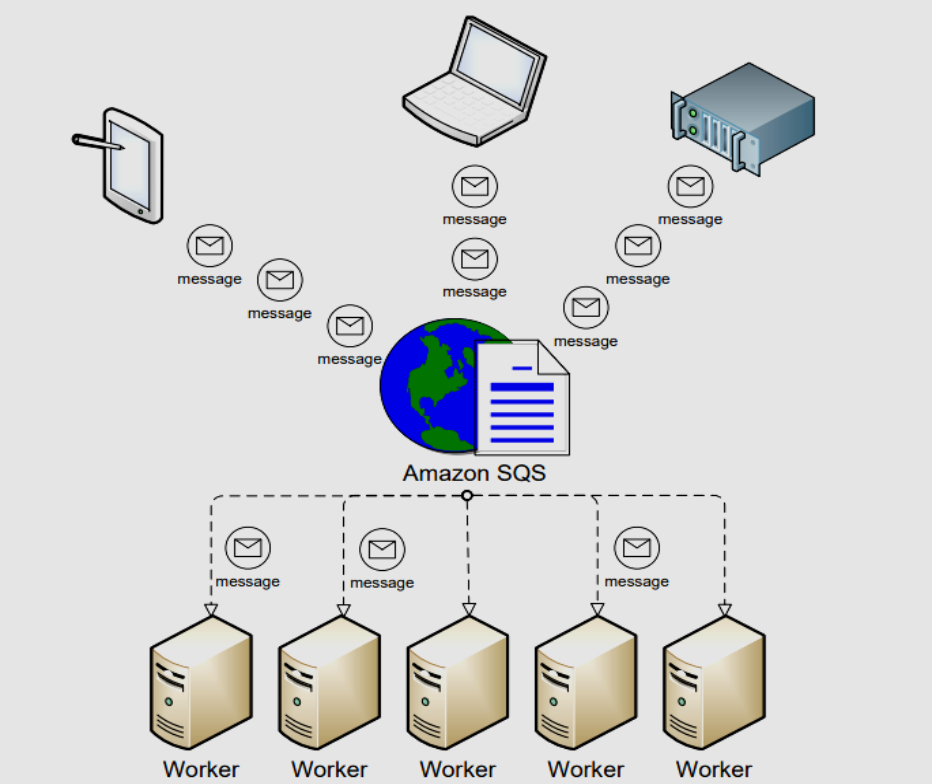

Figure 6 - Amazon SQS System Architecture

To handle requests, this sample makes use of an Amazon SQS queue. Many Amazon Elastic Compute Cloud instances are always scanning the queue for new requests. One of these Amazon Elastic Compute Cloud instances will receive a request and process it. It returns to polling as soon as that instance finishes processing the request. Adding more workers to more instances of Amazon Elastic Compute Cloud allows you to scale upwards as the average message processing time or queue size increases. Typically, Auto Scaling is used to manage Amazon Elastic Compute Cloud (EC2) instances in such a way that there are always sufficient instances acting as "workers" to process the messages that are queued. Even if all worker processes die, Amazon Simple Queue Service (SQS) will keep messages in its queue. You can create replacement Amazon EC2 instances with plenty of time to spare because messages are kept for up to four days.

CONCLUSION

The learning curve for developing cloud applications is greatly reduced with Amazon Elastic Compute Cloud (EC2) because it delivers server instances that are conceptually quite comparable to conventional servers. But that's only the start; you won't get very far in terms of cost, performance, or fault tolerance if you approach Amazon EC2 server instances similarly to normal hardware server instances. By utilising the additional capabilities of Amazon Web Services (AWS) products, such as Elastic Compute Cloud (EC2), the platform's true capabilities can be realised. Elastic IP addresses, multiple Availability Zones, and the ability to quickly commission replacement instances are some of the best practices for building fault-tolerant applications on Amazon Elastic Compute Cloud (EC2). With Auto Scaling, you may automate the launch of a replacement server in the event of a failure, drastically reducing the time and resources needed for server monitoring. When a server isn't working properly, Auto Scaling might automatically start a new one after you terminate the old one. You can make your application's endpoint public with the help of Elastic Load Balancing. Users will not be able to see the ebb and flow of instances being started, terminated, or re-launched on Amazon Elastic Compute Cloud.

REFERENCES

1. Ardagna D (2015) Cloud and multi-cloud computing: current challenges and future applications. In: 7th international workshop on principles of engineering service-oriented and cloud systems (PESOS) 2015. IEEE/ACM, Piscataway, pp 1–2
2. Rastogi G, Sushil R (2015) Cloud computing implementation: key issues and solutions. In: 2nd international conference on computing for sustainable

global development (INDIACom). IEEE, Piscataway, pp 320–324
3. Mell P, Grance T (2011) The NIST definition of cloud computing. Commun ACM 53(6):50 Google Scholar
4. Buyya R et al (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener ComputSyst 25(6):599–616 Article Google Scholar
5. Puthal D et al (2015) Cloud computing features, issues, and challenges: a big picture. In: International conference on computational intelligence and networks (CINE). IEEE, Piscataway, pp 116–123
6. Mesbahi M, Rahmani AM (2016) Load balancing in cloud computing: a state of the art survey. Int J Mod EducComput Sci 8(3):64 Article Google Scholar
7. Mesbahi M, Rahmani AM, Chronopoulos AT (2014) Cloud light weight: a new solution for load balancing in cloud computing. In: International conference (ICDSE) on data science and engineering. IEEE,
8. Piscataway Saab SA et al (2015) Partial mobile application offloading to the cloud for energy-efficiency with security measures. Sustain Comput Inf Syst 8:38–46 Google Scholar
9. Keegan N et al (2016) A survey of cloud-based network intrusion detection analysis. Hum cent Comput Inf Sci 6(1):19 Article MathSciNet Google Scholar
10. Younge AJ et al (2012) Providing a green framework for cloud data centers. Handbook of energy-aware and green computing-two, vol set. Chapman and Hall, UK, pp 923–948 Google Scholar
11. Yuan H, Kuo C-CJ, Ahmad I (2010) Energy efficiency in data centers and cloud-based multimedia services: an overview and future directions. In: Green computing conference, 2010 international. IEEE,
12. Piscataway Zakarya M, Gillam L (2017) Energy efficient computing, clusters, grids and clouds: a taxonomy and survey. Sustain Comput Inf Syst 14:13–33 Google Scholar
13. Zhang Q et al (2014) RESCUE: an energy-aware scheduler for cloud environments. Sustain Comput Inf Syst 4(4):215–224 Google Scholar
14. Bielik N, Ahmad I (2012) Cooperative game theoretical techniques for energy-aware task scheduling in cloud computing. In: Proceedings of the 2012 IEEE 26th international parallel and distributed processing symposium workshops and Ph.D. forum. IEEE Computer Society,
15. Piscataway Zhu X et al (2016) Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. IEEE Trans Parallel DistribSyst 27(12):3501–3517.
16. Nadella, Geeta Sandeep. Validating the Overall Impact of IS on Educators in U.S. High Schools Using IS-Impact Model – A Quantitative PLS-SEM Approach, DAI-A 85/7(E), Dissertation Abstracts International, Ann Arbor, ISBN 9798381388480, 189, (2023).
17. Gonaygunta, Hari, Factors Influencing the Adoption of Machine Learning Algorithms to Detect Cyber Threats in the Banking Industry, DAI-A 85/7(E), Dissertation Abstracts International, Ann Arbor, United States, ISBN 9798381387865, 142, 2023.
18. Ramya Manikyam, J. Todd McDonald, William R. Mahoney, Todd R. Andel, and Samuel H. Russ. 2016.Comparing the effectiveness of commercial obfuscators against MATE attacks. In Proceedings of the 6th Workshop on Software Security, Protection, and Reverse Engineering (SSPREW'16)
19. R. Manikyam. 2019.Program protection using software based hardware abstraction.Ph.D. Dissertation.University of South Alabama.
20. GPB GRADXS, N RAO, Behaviour Based Credit Card Fraud Detection Design And Analysis By Using Deep Stacked Autoencoder Based Harris Grey Wolf (Hgw) Method, Scandinavian Journal of Information Systems 35 (1), 1-8.
21. R Pulimamidi, GP Buddha, Applications of Artificial Intelligence Based Technologies in The Healthcare Industry, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 4513-4519.
22. R Pulimamidi, GP Buddha, AI-Enabled Health Systems: Transforming Personalized Medicine And Wellness, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 4520-4526.
23. GP Buddha, SP Kumar, CMR Reddy, Electronic system for authorization and use of cross-linked resource instruments, US Patent App. 17/203,879.
24. Hari Gonaygunta (2023) Machine Learning Algorithms for Detection of Cyber Threats using Logistic Regression, 10.47893/ijssan.2023.1229.
25. Hari Gonaygunta, Pawankumar Sharma, (2021) Role of AI in product management automation and effectiveness, https://doi.org/10.2139/ssrn.4637857
26. Sri Charan Yarlagadda, Role of Artificial Intelligence, Automation, and Machine Learning in Sustainable Plastics Packaging markets: Progress, Trends, and Directions, International Journal on Recent and Innovation Trends in Computing and Communication, Vol:11, Issue 9s, Pages: 818–828, 2023.
27. Sri Charan Yarlagadda, The Use of Artificial Intelligence and Machine Learning in Creating a Roadmap Towards a Circular Economy for Plastics, International Journal on Recent and Innovation Trends in Computing and Communication, Vol:11, Issue 9s, Pages: 829-836, 2023.

28. Amol Kulkarni, Amazon Athena Serverless Architecture and Troubleshooting, International Journal of Computer Trends and Technology, Vol, 71, issue, 5, pages 57-61, 2023.
29. Amazon Redshift Performance Tuning and Optimization,International Journal of Computer Trends and Technology, vol, 71, issue, 2, pages, 40-44, 2023.
30. B. Nagaraj, A. Kalaivani, S. B. R, S. Akila, H. K. Sachdev, and S. K. N, "The Emerging Role of Artificial intelligence in STEM Higher Education: A Critical review," International Research Journal of Multidisciplinary Technovation, pp. 1–19, Aug. 2023, doi: 10.54392/irjmt2351.
31. D. Sivabalaselvamani, K. Nanthini, Bharath Kumar Nagaraj, K. H. Gokul Kannan, K. Hariharan, M. Mallingeshwaran, Healthcare Monitoring and Analysis Using ThingSpeak IoT Platform: Capturing and Analyzing Sensor Data for Enhanced Patient Care, IGI Global eEditorial Discovery, Pages: 25, 2024. DOI: 10.4018/979-8-3693-1694-8.ch008.