International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Leveraging AWS Full Stack Development Platform for Scalable and Reliable Enterprise Applications

[1]Anaswara Thekkan Rajan, [2]Sneha Aravind, [3]Sagar Shukla, [4]Ranjit Kumar Gupta

Submitted: 03/01/2024 **Revised**: 15/02/2024 **Accepted**: 20/02/2024

Abstract: Cloud computing has become an essential innovation in the quickly changing digital change landscape, giving enterprises unprecedented operational flexibility and scale. The emergence of cloud computing has brought about a significant transformation in contemporary technological frameworks, with Amazon Web Services, also known as AWS, developing as a prominent cloud player. In order to realise AWS's maximal potential and showcase its revolutionary advantages for both individuals and enterprises, the research delves into its subtleties. A method is shown for automating and expanding business application cloud installations using contemporary tools and techniques. Conventional infrastructure installations have lost ground to modern cloud-based alternatives for businesses. A detailed explanation is given of the AWS services utilised and the unique tasks they have in relation to the suggested design. To demonstrate the suggested architecture's capacity to handle and absorb data, a stress test was carried out. Numerous commercial and open-source automating and scaling options are accessible. Performance, usability, and sustainability issues with automated the application's deployment in the cloud exist. In earlier studies, we outlined specific standards for selecting a service that offers cloud storage and a method for growing cloud-based applications to thousands of users. Our objective is to provide a set of standards for selecting an appropriate scaling and automate strategy for applications used by businesses.

Keywords: *Digital Transformation, Modern Cloud, Amazon Web Services (AWS), Enterprise Applications, Automation, Scaling Solutions, Algorithms, Enterprise Applications, Cloud-Based Solutions.*

I. INTRODUCTION

In today's ever-changing corporate environment, maintaining competitiveness and relevance has come to be associated with the idea of digital transformation. Scalable and operational versatility have become critical success factors for organisations as they endeavour to adjust to changing market demands and technology improvements [1]. Modern cloud computing technology have revolutionised how organisations function and expand by making it possible for them to accomplish these goals. More than simply a catchphrase, "digital transformation" actually refers to a fundamental shift in how companies use technology to improve consumer experiences, simplify processes, and spur innovation [1, 2]. This transition is primarily driven by the requirement for organisations to be adaptive and nimble in the face of change, which calls for scalable technology solutions and infrastructures. Scalability is the capacity of an organisation to grow or shrink in reaction to shifting customer needs [2]. This is especially important given the quick-changing nature of the industry in today's fast-paced corporate climate [2, 3].

[1]*Independent Researcher,USA.*
[2]*Independent Researcher,USA.*
[3]*Independent Researcher, USA.*
[4]*Independent Researcher, USA.*

These days, public clouds provide a greater choice of services than co-location or on-premises data centres. These cloud-based offerings includes storage, virtualised and physical hardware, the form of databases, server-less lambda functions, IoT, AI, storage, virtualised containers environments, and data bases [3]. Because of its many benefits, cloud computing services are becoming a vital resource for businesses, organisations, teams, and entrepreneurship [3].

We list a few of these Cloud's Most Notable Advantages below:

- Flexibility in terms of pay-per-use pricing structures, upward/downward elastic capabilities, and scalability/elasticity characteristics; [3],
- Flexibility through the highest level Service Level Agreements (SLA), global regions/Availability Regions, monitoring tools, backup, and failover capabilities.

- Advanced features for automating and modernisation that make use of programmed access or cloud-native software creation capabilities, as well as features for privacy and governance, requirements for compliance, and credentials, are offered [2, 3].
- Balancing expenses and environmental effects.
- There are many platforms in the area of cloud services provisioning industry: Some examples are Google Cloud and Microsoft Azure, although Amazon Web Services (AWS) stands out as having more expertise offering these kinds of services [3].

1.1 Amazon History

"Amazon.com Web Service" is the name of the service that Amazon began testing in 2002. This functioned similarly to a tool used by developers of software to retrieve data from Amazon's list of products using particular computer code techniques. A year later, in 2003, Amazon's CEO, Jeff Bezos, called a gathering with his senior staff at his home. His goal was to identify Amazon's areas of expertise. They saw that Amazon had a distinct edge when it came to infrastructure and computing system management. They therefore had a really great idea [3]. They considered developing an Internet-based operating system. Assume that Windows or macOS, or an equivalent operating system, lets your machine function flawlessly. In a comparable manner, they sought to develop tools for controlling many aspects of the Internet, such as data storage, database management, and computer job execution [3, 4].

Deployments of cloud applications raise the demand for security and performance efficiency. These are the areas of study that the global cloud community is most engaged in. These phrases are already included in the Quality of Service for the cloud service providers [4].

These phrases are also a result of the most recent "transforming" trend in the IT sector. Businesses must adapt their approaches to the development, deployment, maintenance, and even storage of end-user data of new apps [4, 5]. Faster mobile networks and the use of cloud computing have made faster access to information essential. The most alluring aspects of the contemporary cloud architecture offered by the major providers appear to be scalability and productivity. Cloud infrastructure distribution solutions should be scalable and accessible to optimisation [5]. There are several approaches available for scaling the deployment of cloud applications based on network data like latency to estimate performance. Our present research suggests standards for creating scalable settings for corporate application deployments in the cloud.

The following general layers comprise the centralised cloud computing paradigm, which the survey accurately defines:

- **Centralised cloud computing layer (I):** Encompasses the cloud datacentre providers. Long-term preservation and application-level processing of information processes, which are usually less time-sensitive, are done at this layer [5]. Depending on user needs, applications under this layer may contain many service components, each serving a distinct role for high-level processing of information.
- **SDN/NFV technology layer (layer II):** Modern approaches for developing creative network services from architecture to implementation and operating include Software Defined Networking (SDN) and Network Functions Virtualisation (NFV) [5]. They facilitate the transfer of data between cloud datacentres and edge nodes.
- **Layer (III) Edge computing layer:** The edge computing layer is made up of gateways as well as information gathering services that process raw data by filtering, encoded data, encrypting, and aggregating local internet-based information streams. The distribution and proximity of cloud resources to end consumers and end gadgets occurs at this tier. Another term for computing at the edge is ubiquitous computing [5]. This layer can assist in lowering traffic coming from data centres and the largest network.

The remainder of the document is structured as follows: The cloud's options for selecting measurements of performance are explained in section 2. The scaling conditions are explained in section 3. Findings of the experiment are presented in Section 4.

II. METHOD AND MATERIALS

2.1 Cloud-based corporate application scalability concept

The idea behind cloud computing is that its scalability is theoretically endless. It is only once the software infrastructure is scalable that one may take use of and profit from the cloud architecture. In light of this, [5] requires the detection of monolithic elements and roadblocks in the current structure, the identification of areas in which the architecture's immediate provisioning capacities are not feasible, and the reworking of the application to make use of the scalable framework and cloud. The process of creating scalable applications is explained in terms of its key characteristics, which are as follows:

- Services that are scalable can manage heterogeneity.
- Functionally, scalable services are effective.
- An increase in resources would lead to a corresponding improvement in performance.
- Resilient offerings are scalable.
- Scalable businesses have the potential to become increasingly cost-effective as they expand, as the cost per unit decreases as the number of units grows.

2.2 Tracking performance metrics and contexts

The items from the various centralised cloud model layers are described in this section along with the corresponding metrics. Analysing the use of the virtual machine's (VM) CPU, Memory, and Disc resource can help scale an enterprise cloud program [5, 6]. The system runs out of processing power if the CPU utilisation approaches 100% and the CPU run queued are almost full. Predictive analysis should be done before that happens to ensure that the system is ready for the next difficulty. Memory use displays the portion of the machine's memory that is in use. Once more, adaptability is required if the memory usage is excessive. The quantity of data that a certain Virtual Machine (VM) reads or writes determines its disc use.

2.3 Cloud-based corporate application scaling

The scaling technique's main procedure is shown in Fig. 1. The corresponding application instances provide the performance metrics. The customer's request determines how the scaling parameters are adjusted. Based on the modified thresholds, the performance is assessed [5], after which scaling is carried out to increase the number of instances and enhance performance or decrease the number of instance dynamically [6, 7]. The present state of the surroundings is maintained if scaling is not required. A performed efficiency estimation and tweaking of a rather large automated Amazon Web Services (AWS) setup is used as an example [7].

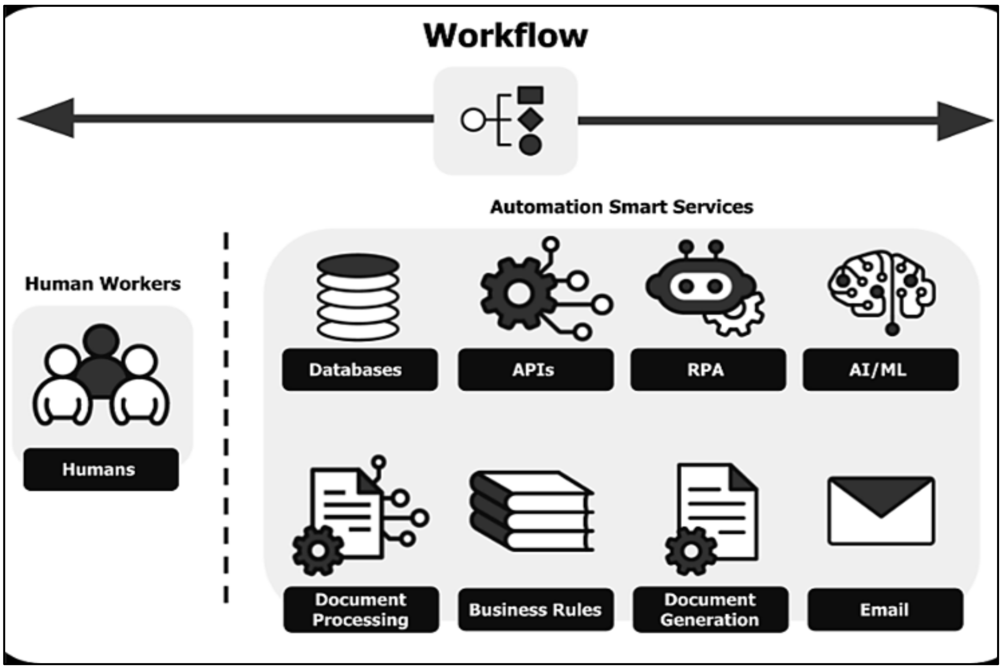

Fig. 1 Process of the suggested automation and scalability. [7, 8]

Certain tasks may be completed synchronously and Rabbit MQ and Celery are utilised to do this. Static material and media files are kept on Amazon S3 [8]. A number of tools for handling and automating the infrastructure's creation and deployment have been assessed. Additional details on the environment components (as seen in Fig. 2) and the assessed instruments used in this study is provided below [8, 9].

A. **HAProxy:** HAProxy is an open-source and free application that may act as a proxy server and a load balancer for TCP and HTTP-based applications with high availability. It is known to be quick and effective, is written in C, and can distribute queries among several backend servers [8, 9].

B. **Django:** Model-view-template architecture is used by the Python web framework Django, which is open-source and free to download [9]. The Django Software Society is a separate entity that was founded as a 501 non-profit to support and advance the framework in question.

C. **Puppet:** An open-sourced server automating program called Puppet assists in automating the maintenance and configuration of contemporary IT infrastructure. Puppet is set up in an agent-master architecture framework [8, 9], where all configurable data for several managed configurations (nodes) is stored on a master node executing an agent [9].

D. **Chef:** Chef is a potent coding language and automating tool for infrastructure. It uses a client-server architecture to function. Every managed node has an agent called Chef Client running on it. It establishes a connection with a Chef Server on a regular basis in order to download and assess recipes, which are configuration codes [9, 10]. The system wouldn't be altered if there was no need for it [10].

E. **Ansible:** Red Hat® Ansible® is an IT automation tool that creates predictable, scalable and straightforward procedures out of the repetitive, wasteful chores associated with releasing software cycles [10]. Red Hat's Ansible can automate cloud provisioning, orchestration of services, installation of applications, and management of configurations.

F. **Terraform:** An automation framework called Terraform from Hashi Corp. is focused on building, modifying, and enhancing IT infrastructure [10, 11]. It converts APIs into files for configuration that can be shared, handled like code, revised, reviewed, and versioned by members of the team. It is open-source and free of charge [11]. It permits the reuse of infrastructure deployment and architecture between various providers of cloud services [12, 11].

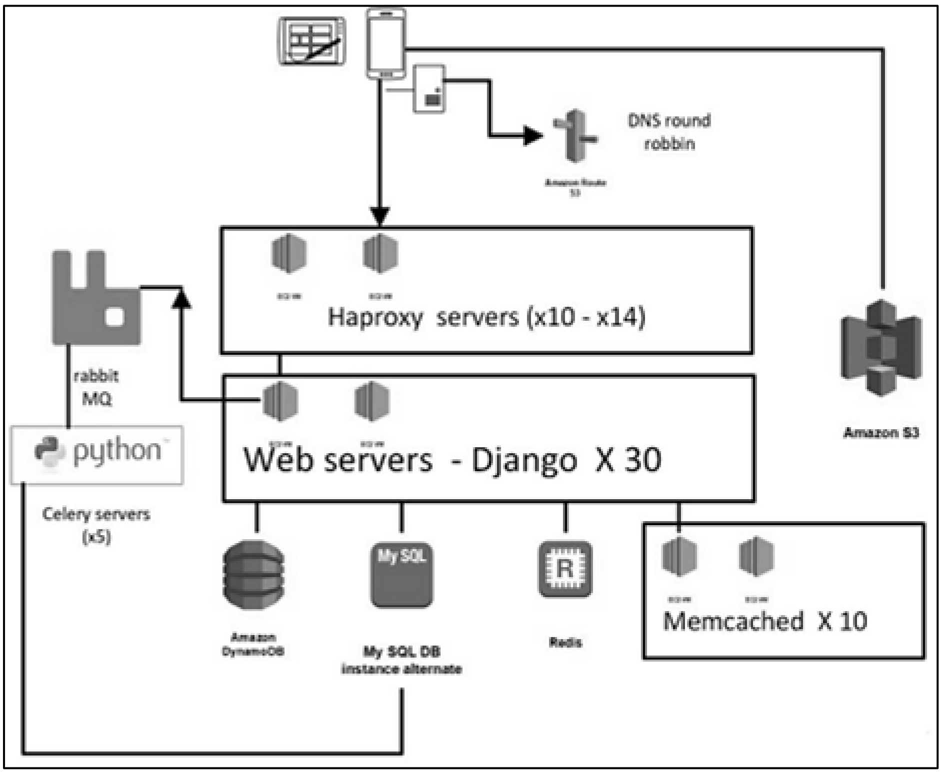

Fig. 2 Architecture of environments. [13, 14]

Utilising Chef handling configurations, the backend limitations are dynamically evaluated [15]:

$N_{web-server-instances}/N_{HAProxy-instances.}$ …………1

2.4 Modifying the scaling limitation

The increase in the threshold is a dynamical metric that is determined by the environment as it exists right now [17]. KAMA is used to modify the coefficient [16, 17].

III. RESULT AND DISSUASION

Images from two distinct HAProxy load balancers taken at the same time interval are included in Table 1 [18, 19]. Unevenly dispersed connections across the HAProxy instances are shown in Sessions Cur (current count), which represents the current total number of sessions [20, 21]. The statistics from a server with few open connections are displayed in Table 1 [22, 23].

Table 1 Session distribution among HAProxy instances (lower-tono load). [24]

	Queue			Session Rate			Sessions						Bytes
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	Lb Tot	Last	In
Webserver –tb-06	0	0	1700	51	471		1	425	425	127.921	121.508	Os	68.649.589
Webserver –tb-07	0	0	1700	51	470		1	425	425	127.921	121.512	Os	68.519.149
Webserver –tb-08	0	0	1700	51	485		1	425	425	127.962	121.516	Os	68.954.258
Webserver –tb-09	0	0	1700	51	488		2	425	425	127.955	121.521	Os	68.941.962
Webserver –tb-010	0	0	1700	51	472		1	425	425	127.941	121.549	Os	68.952.625
Webserver –tb-11	0	0	1700	51	421		2	425	425	127.968	121.522	Os	68.986.215
Webserver –tb-12	0	0	1700	51	472		1	425	425	127.914	121.526	Os	68.512.625
Webserver –tb-13	0	0	1700	51	745		1	425	425	127.932	121.596	Os	68.799.541

Webserver –tb-14	0	0	1700	51	761		1	425	425	127.941	121.580	Os	68.648.965
Webserver –tb-15	0	0	1700	51	720		1	425	425	127.963	121.529	Os	68.648.965
Webserver –tb-16	0	0	1700	51	411		2	425	425	127.978	121.596	Os	68.948.589
Webserver –tb-17	0	0	1700	51	470		1	425	425	127.925	121.599	Os	68.648.215
Webserver –tb-18	0	0	1700	51	450		1	425	425	127.983	121.496	Os	68.284.659
Webserver –tb-19	0	0	1700	51	480		2	425	425	127.965	121.451	Os	68.641.896
Webserver –tb-20	0	0	1700	51	479		1	425	425	127.981	121.496	Os	68.641.989

This is also hinted at in Fig. 3 over the years in the Data Dog monitoring. One HAProxy session count is represented by each colur [25, 26]. This process seems to have allowed us to identify that AWS Route 53 balances in a certain manner, providing the first proxy IP to every client and subsequently the second IP to every client in a matter of seconds [27, 28]. In this manner, for a predetermined amount of time, every client connects to the same HAProxy. Although they are gradually balancing, peaks are only growing up on one HAProxy, which is vulnerable to overloading [18, 28].

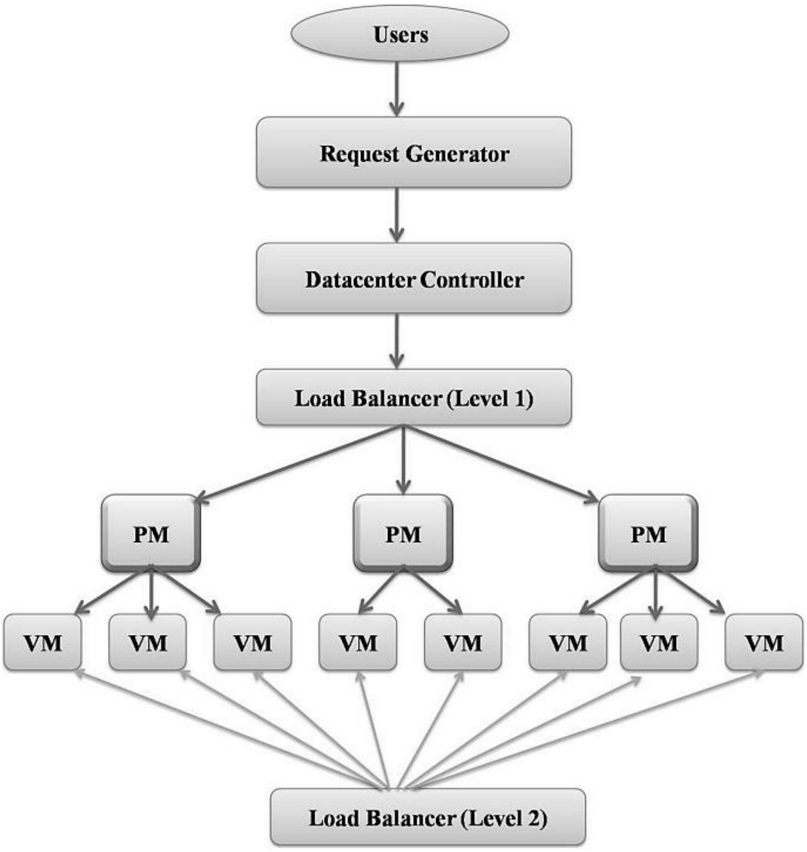

Fig. 3 Efficiency coefficient computed over time for each balancer server. [25, 28]

IV. CONCLUSION

The results of our continuing experiments are as follows:

— Throughout an online streamed event, scaling the HAProxy instances to a maximum number of eight is suggested, preferably four. The advantage of this is that the per backend, per proxy restrictions will be increased even if customers connect to the same HAProxy balancer. The information demonstrates that despite being highly loaded, the proxy servers are able to process a remarkably high volume of requests. The active connections via TCP limit of around 65000 TCP sessions per HAProxy is represented by a single constraint. Accordingly, four instances ought to be able to handle the 200K open connections with ease.

- It is recommended to utilise a single DNS record for multivalued responses. Improved request distribution has been demonstrated through load testing.
- Efficiency in scaling may be attained by employing automation technologies and putting the recommended auto scaling strategy into practice. Future developments of this study will include expanding the scaling technique to hybrid clouds and enhancing the neural network-based adaptive thresholds computation.

References

[1] Milhem H, Harrison NB (2022) The Quality Attributes and Architectural Tactics of Amazon Web Services (AWS) 2022 Intermountain Engineering, Technology and Computing (IETC) 1-6.

[2] Williams, Dana M (2020) Power Accrues to the Powerful: Amazon's Market Share, Customer Surveillance, and Internet Dominance. The Cost of Free Shipping: Amazon in the Global Economy, edited by Jake Alimahomed-Wilson and Ellen Reese, Pluto Press 35-49.

[3] Pelle I, Czentye J, Dóka J, Kern A, Gerő BP, et al. (2021) Operating Latency Sensitive Applications on Public Serverless Edge Cloud Platforms. In IEEE Internet of Things Journal, vol. 8: 7954-7972.

[4] Kewate N (2022) A Review on AWS - Cloud Computing Technology. International Journal for Research in Applied Science and Engineering Technology. AWS basic 10: 258- 563.

[5] Hashemipour S, Ali M (2020) Amazon Web Services (AWS) – An Overview of the On-Demand Cloud Computing Platform. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 332: 40-47.

[6] Priyadarshini P, Veeramanju KT (2022) A Systematic Review of Cloud Storage Services- A Case Study on Amazon Web Services. International Journal of Case Studies in Business, IT and Education (IJCSBE) 6: 124–140.

[7] Adebayo, A. O., Ogunbiyi, E. O., Adebayo, L. O., & Adewuyi, S. (2021). Schiff Base Modified Chitosan Iron (III) Complex as new Heterogeneous Oxidative Catalyst. Journal of Chemical Society of Nigeria, 46 (2).

[8] Adelakun, B. O. (2023). How technology can aid tax compliance in the US Economy. Journal of Knowledge Learning and Science Technology, 2(2), 491-499.

[9] Daramola, G. O., Jacks, B. S., Ajala, O. A., & Akinoso, A. E. (2024). AI applications in reservoir management: optimizing production and recovery in oil and gas fields. Computer Science & IT Research Journal, 5(4), 972-984.

[10] Edu, Y., Eimunjeze, J., Onah, P., Adedoyin, D., David, P.O., & Ikegwu, C. (2022). Fintech update: SEC new rules on the issuance, offering platforms and custody of digital assets- what you need to know. Mondaq (July 6, 2022).

[11] Ngwuli, O.D., PC, Moshood, F.J., Uwaga, A.M., & Chukwuemeka (2022). Comparative Evaluation of Nutritive values of four fodder plant species in Umudike, Abia State, South-eastern Nigeria. Conference Proceeding of the 8th Biennial Conference of the Forest and Forest products Society on Forestry and the Challenges of Insecurity, Climate Change and COVID -19 Pandemic in Nigeria Volume 8 Issue 2022 Pages 188 – 193.

[12] Osuagwu, E. C., Uwaga, A. M., & Inemeawaji, H. P. (2023). Effects of Leachate from Osisioma Open Dumpsite in Aba, Abia State, Nigeria on Surrounding Borehole Water Quality. In Water Resources Management and Sustainability: Solutions for Arid Regions (pp. 319- 333). Cham: Springer Nature Switzerland.

[13] Martos, V.; Ahmad, A.; Cartujo, P.; Ordoñez, J. Ensuring Agricultural Sustainability through Remote Sensing in the Era of Agriculture 5.0. Appl. Sci. 2021, 11, 5911.

[14] Köksal, Ö.; Tekinerdogan, B. Architecture design approach for IoT-based farm management information systems. Precision Agric. 2019, 20, 926–958.

[15] Gaire, R.; Lefort, L.; Compton, M.; Falzon, G.; Lamb, D.; Taylor, K. Demonstration: Semantic Web Enabled Smart Farm with GSN. In Proceedings of the International Semantic Web Conference (Posters & Demos), Sydney, Australia, 21–25 October 2013.

[16] Sokolov, Strahil A., Stefan M. Vlaev, Mihail Vukadinoff, Asen P. Zahariev, Teodor B. Iliev, and Ivaylo S. Stoyanov. "Performance Estimation of Scalable e-Learning Systems in the Cloud." In 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging(SIITME), pp. 148-151. IEEE, 2018.

[17] Marschall, Matthias. Chef infrastructure automation cookbook. Packt Publishing Ltd, 2015.

[18] Varia, J., 2010. Architecting for the cloud: Best practices. 5. Loope, James. Managing infrastructure with puppet: configuration management at scale. “O’Reilly Media, Inc.", 2011.
[19] Taylor, M. and Vargo, S., 2014. “Learning Chef: A Guide to Configuration Management and Automation." O'Reilly Media, Inc...
[20] Geerling, Jeff. "Ansible for DevOps." (2014). 8. Morris, Kief. “Infrastructure as code: managing servers in the cloud." O'Reilly Media, Inc., 2016.
[21] Akhigbe, B.I.; Munir, K.; Akinade, O.; Akanbi, L.; Oyedele, L.O. IoT Technologies for Livestock Management: A Review of Present Status, Opportunities, and Future Trends. Big Data Cogn. Comput. 2021, 5, 10.
[22] Goddard, M.E. Uses of genomics in livestock agriculture. In Animal Production Science; CSIRO Publishing: Clayton, Australia, 2012; pp. 73–77.
[23] Awiti, J.; Vaisman, A.; Zimányi, E. Design and implementation of ETL processes using BPMN and relational algebra. Data Knowl. Eng. 2020, 129, 101837.
[24] Zhang, B. AWS Identity-based Policies with “Read”, “Write” and “Execute” Actions. Master’s Thesis, University of Waterloo, Waterloo, ON, Canada, 2020; pp. 15–19.
[25] Ahlam, A.; Nazmeen, K.; Zoya, R.; Pranali, T. Reinforcing Security of DNS Using AWS Cloud. In Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST), Sion, Mumbai, 8–9 April 2020.
[26] Tawalbeh, M.; Quwaider, M.; Tawalbeh, L.A. IoT Cloud Enabeled Model for Safe and Smart Agriculture Environment. In Proceedings of the 12th International Conference on Information and Communication Systems (ICICS), Valencia, Spain, 24–26 May 2021; pp. 279–284.
[27] Ilyas, Q.M.; Ahmad, M. Smart Farming: An Enhanced Pursuit of Sustainable Remote Livestock Tracking and Geofencing Using IoT and GPRS. Wirel. Commun. Mobile Comput. 2020, 2020, 6660733.
[28] Tsuchiya, T.; Mochizuki, R.; Hirose, H.; Yamada, T.; Koyanagi, K.; Tran, M.Q. Distributed Data Platform for Machine Learning Using the Fog Computing Model. SN Comput. Sci. 2020, 1, 164.