International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
http://ijcjournal.org/

Cloud Infrastructure Automation Through IaC (Infrastructure as Code)

Muhammad Raquibul Hasan[a*], Md. Sifuddin Ansary[b]

[a]*Cloud CoE Expert Consultant, Project Head, BRAC, Dhaka, Bangladesh*
[b]*Head, Network Biz, Samsung Electronics, Dhaka, Bangladesh*
[a]*Email: mr_hasan@yahoo.com*
[b]*Email: sifuddin@gmail.com*

Abstract

Cloud computing has emerged as a key force in the IT sector, allowing businesses to take use of the cloud's scalability and cost-effectiveness to satisfy their IT requirements. When compared to the traditional on-premise method, companies that use cloud services can benefit from higher efficiency, dependability, and agility, as well as lower operational expenses and improved security. In spite of all the benefits, configuring cloud services may be a difficult and time-consuming procedure, especially when done manually. As a solution to the mentioned challenges, the Infrastructure as Code (IaC) was introduced. IaC has revolutionised the way organisations approach IT. Organisations may gain improved efficiency, dependability, and agility by exploiting the cloud's scalability and cost-effectiveness, as well as automating the provisioning, configuration, and administration of infrastructure with different IaC tools. However, using cloud and IaC can also present challenges, including complexity, collaboration, versioning, testing, security, integration, and automation. This paper discusses the overview of cloud computing and IaC. Explores the benefits and challenges of using these technologies. Concludes by emphasising the importance of careful planning and execution when using cloud and IaC, in order to maximise the benefits while minimising the difficulties.

Keywords: IaC; Cloud Computing; cloud-automated.

1. Introduction

The cloud refers to an environment for sharing resources in the form of bottom-up frameworks, middleware, and application development platforms and business applications. Cloud operating models capture valuable free infrastructure services, other platform services, and subscription-based infrastructure services with additional application services. The service is free to the seller but shares the revenue generated by the buyer [1]. The cloud computing architecture is composed of two primary components: the frontend and the backend. In such architecture, the frontend acts as a client and interacts with the backend over a network or the internet. The client-side or frontend is visible to the end user in the cloud computing architecture.

---

* Corresponding author.

The middleware routes queries from the frontend to the backend. On the backend, there are several servers, information storage, and other cloud computing services operating. Cloud computing provides users with their necessities [2]. At the lower portion of the cloud architecture is the hardware, where the physical assets of clouds are situated. Then lies the infrastructure layer, platform layer. The application layer from where the users interact or access the cloud. Depending on the service type, it can be taken as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)[4, 5, 6]. Some benefits of the cloud can be mentioned as fewer skills are needed to work on it, as most of the work is done by the cloud service providers. To access cloud services, it can be easily accessed from anywhere. No hardware dependency, pay-as-you-run service, cost-efficient, and easily scalable [7]. But there are some issues in the cloud such as an internet connection is a must with moderate speed. Sometimes the cost gets exponentially high when the storage or devices are modified. But when a huge number of servers are there, it is costly and time-consuming to maintain those. To remedy this problem, Infrastructure as code (IaC) is being introduced and also being popular at the same time. While implementing a delivery practice IaC scripts are used. IaC technologies such as Chef and Puppet provide cloud-based instances that manage databases and accounts [9]. IaC escalates the productivity of the IT operation teams. Before the introduction of IaC tools, infrastructure scaling had to be handled and managed with complex manual work. It costs a great amount of time, causes unwanted errors, and slows the growth of the organisation. IaC also reduces human error rates. It standardises the processes and helps the new employees to progress the work without any setbacks. It gives us an opportunity to optimise the financial cost by efficient usage of the environment. IaC enables our business cloud infrastructure to comply with applicable regulations. So, it's very much essential to integrate the IaC into the business model to move forward in this smart era.

2. Literature Review

The paper "Terraform: Automating Infrastructure as a Service" discusses the importance of using a rigorous software development process for both the software service and its underlying infrastructure. The author argues that manual provisioning of infrastructure can become tedious and error-prone as the number of resources grows, especially if spread across multiple regions. To overcome these challenges, the author suggests using Terraform, a platform that allows for the automation of infrastructure provisioning through the definition of resources in code. This not only streamlines the provisioning process but also enables a rigorous development and review life cycle for the infrastructure, similar to that of the application software. Deployment of the infrastructure using the programming language is supported in Cloud Development Kit for terraform [10]. The paper "DevOps: Introducing Infrastructure-as-Code" discusses the use of infrastructure-as-code as a tactic in DevOps, a set of software engineering practices aimed at shortening the time between design changes and their implementation. The paper outlines the elements and abstractions required for writing and maintaining a blueprint for deployment in the cloud, focusing on the use of the TOSCA (Topology and Orchestration Specification for Cloud Applications) industrial standard, which has been adopted by 60 or more major industrial players worldwide. The paper argues that TOSCA provides a key standard for expressing the blueprint and is necessary for successful infrastructure-as-code in a DevOps context [11]. Infrastructure as Code (IaC) is an approach for infrastructure automation that is based on software development practices. In previous work, it has been presented as an infrastructure modelling approach and tool (Argon) for cloud provisioning that

leverages model-driven engineering and supports the IaC approach. Argon is a tool for cloud provisioning that leverages model-driven engineering and supports the IaC approach. Argon accelerates the provisioning process by modelling the cloud infrastructure and automating the generation of scripts. The IaC approach supports code-centric tools that use scripts to specify the creation, updating, and execution of cloud infrastructure resources [12]. The method of automatically configuring system requirements and provisioning local and remote instances is known as infrastructure as code (IaC). IaC is seen by practitioners as a vital pillar for the implementation of DevOps principles, which enables them to provide software and services to end customers quickly. Organisations in the information technology (IT) sector, including GitHub, Mozilla, Facebook, Google, and Netflix, have embraced IaC. Researchers can find promising IaC research topics, such as flaws, by conducting a thorough mapping analysis of existing IaC research. and potential security holes in IaC scripts. By undertaking a methodical mapping analysis of IaC-related research, this work seeks to assist academics in identifying research areas connected to infrastructure as code (IaC) [13].

3. Traditional Approaches

Administrators need to manually configure cloud servers typically involving logging into the server through a command line interface, such as SSH, and then using various commands to install and configure the necessary software and settings. This can include installing the operating system, configuring network settings, setting up user accounts, and installing any necessary applications or services. Some cloud providers also provide web-based management consoles that allow them to perform these tasks through a graphical interface. Picking up different components for the infrastructure such as CPU, ram, storage drive, network groups, machine images, and also configuring all these different tools/services altogether for different applications (Web, Mobile) can be challenging. Whenever the number of virtual machines increases in numbers, manual work will increase too. Large organisations such as e-commerce, NGOs, educational institutions, and government services will require a lot of human expertise to maintain the infrastructure.

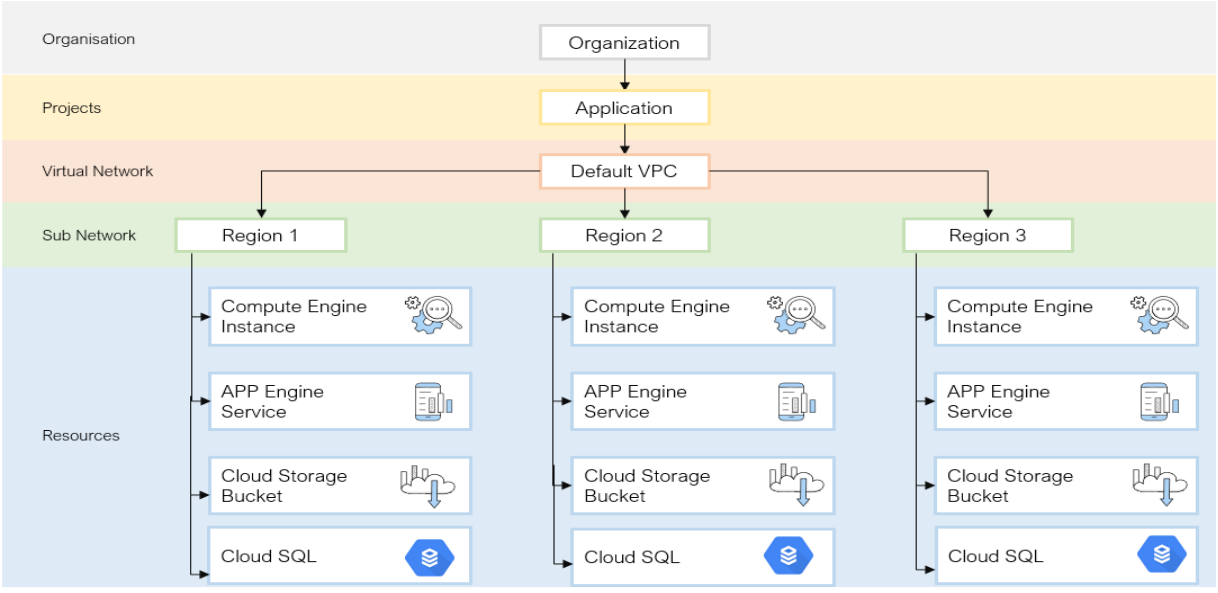

Figure 1: Traditional Approach.

Manually managing cloud servers can have several drawbacks, including

- Time-consumption: Manually configuring and maintaining servers can be a time-consuming and labour-intensive process, especially if multiple servers need to be managed.
- Prone to human error: Manually working requires lots of repetitive tasks, which can lead to misconfigurations and security vulnerabilities.
- Difficulty in scaling: Manually managing servers can make it difficult to quickly scale up or down as needed, such as during periods of high traffic.
- Privation of standardisation: Without automation, it can be difficult to ensure that all servers are configured in the same way, which can lead to inconsistencies and compatibility issues.
- Absence of Visibility: Without proper monitoring, it can be difficult to identify and troubleshoot issues with servers that are manually managed.
- Difficulty in disaster recovery: It is difficult to quickly and easily recover from a disaster or outage if the work is done manually.

Because of the above-mentioned drawbacks, large organisations opt to use automated tools such as orchestration and configuration management tools like Ansible, Puppet, Chef and Terraform to manage their cloud servers.

4. What is IaC

Infrastructure as Code (IaC) is a modern software engineering practice that enables the provisioning and management of IT infrastructure through code, rather than manual configuration. This approach helps to reduce human error, increase consistency and repeatability, and streamline the overall process of setting up, configuring, and deploying infrastructure. IaC also enables teams to manage their infrastructure in a more agile and flexible way, by allowing them to version, test, and automate changes just as they do with software. This, in turn, supports the principles of DevOps and helps organisations to improve the speed and quality of their application delivery. The use of IaC can greatly benefit organisations by reducing the time and effort required to provision and manage infrastructure, while also improving the reliability, consistency, and scalability of their IT systems

5. Benefits of IaC

By proposing this solution, some benefits which are very much effective and efficient in terms of the organisation's perspective can be observed. One of the main benefits of this cloud-automated infrastructure is that it allows for the automatic scaling of servers. This can be particularly useful for organisations that experience fluctuations in traffic, as it allows them to quickly and easily scale up or down as needed. This can help to save costs and ensure that the infrastructure is able to handle the load.

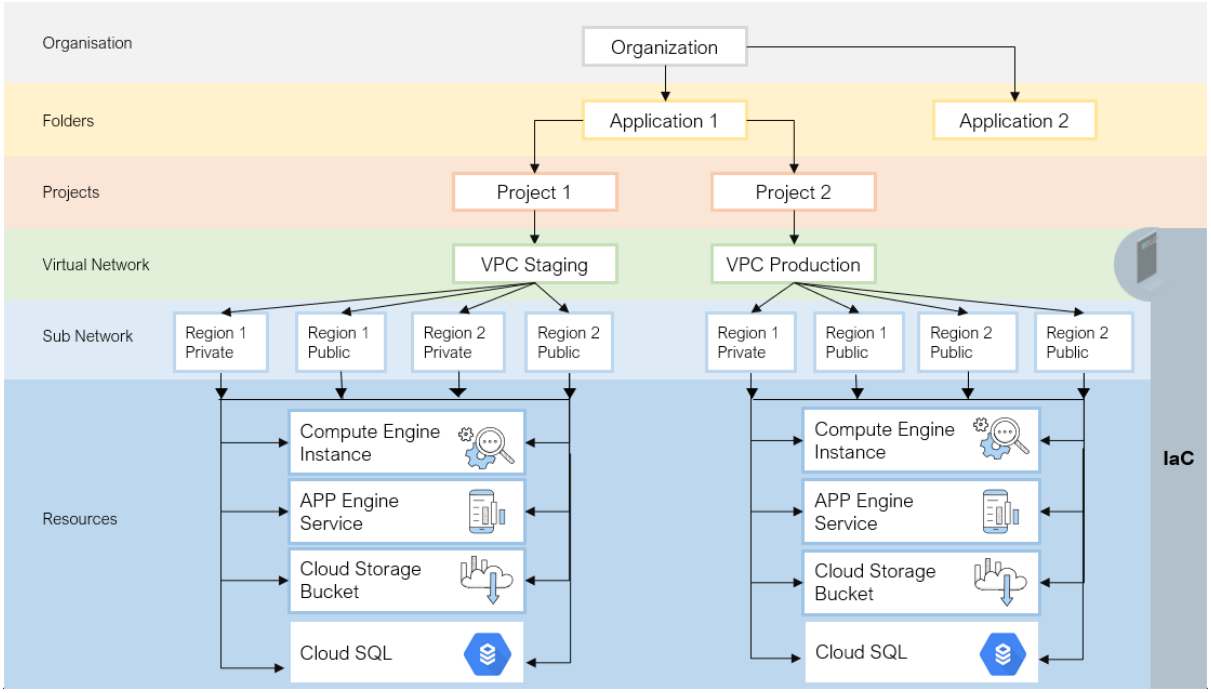

Figure 2: Proposed Approach.

Another benefit is that it reduces the need for manual intervention. With cloud-automated infrastructure, servers can be provisioned and managed automatically, which reduces the need for manual tasks and can help to improve efficiency. Additionally, cloud-automated infrastructure allows for better monitoring and troubleshooting. With automated tools, it is easier to identify and troubleshoot issues with servers, which can help to improve the overall performance and availability of the infrastructure. It also allows for version control and auditing capabilities. With the ability to track changes to the infrastructure over time, it can be useful for compliance and security purposes. Finally, it can help in disaster recovery. With the automation of infrastructure, it is easier to quickly and easily recover from a disaster or outage.

Overall, cloud-automated infrastructure provides many benefits for organisations, including improved scalability, reduced human labour, and improved monitoring and troubleshooting capabilities. However, it also requires a certain level of technical expertise and may be complex for large organisations with a large and complex infrastructure. It is important to weigh the benefits and challenges when considering implementing cloud-automated infrastructure.

6. Challenges and Considerations

Infrastructure as Code (IaC) workflow entails writing code to define infrastructure, storing it in a version control system, testing, deploying, monitoring, and maintaining it over time. This code is used to automate infrastructure provisioning, configuration, and maintenance.

It allows organisations to manage their infrastructure in a uniform, repeatable, and versioned manner. While IaC has many benefits, it also comes with some challenges and considerations that organisations should keep in

mind. One of the main challenges is that IaC requires a certain level of technical expertise. Organisations will need to have the necessary skills and resources to write and maintain the code that defines their infrastructure.

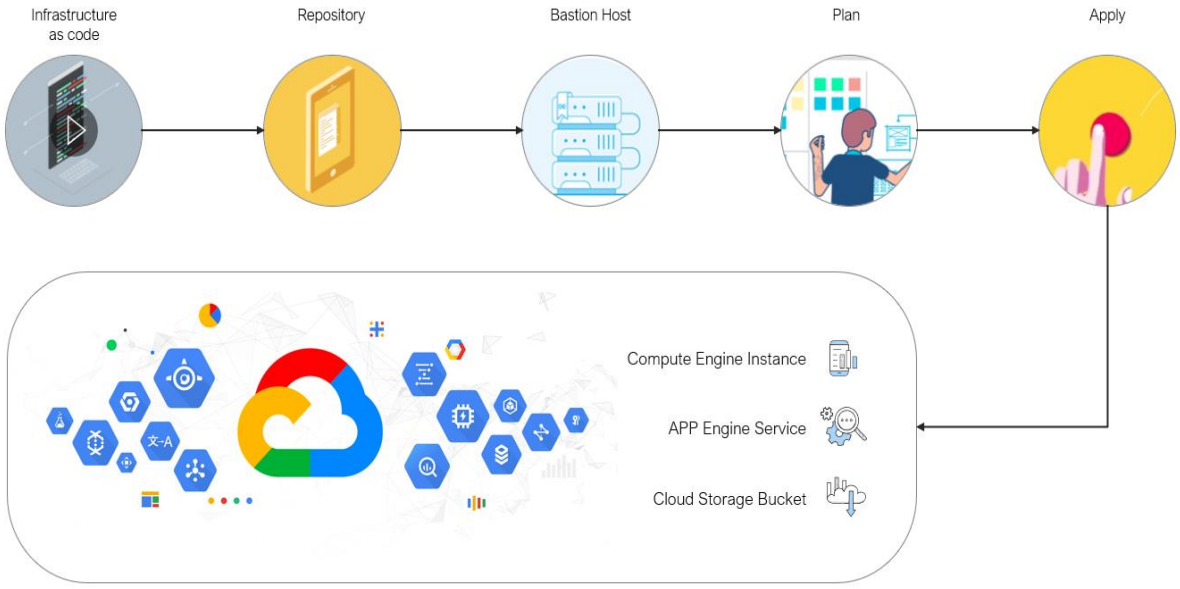

Figure 3: IaC Workflow.

Another challenge is that IaC can be complex, particularly for organisations with a large and complex infrastructure. Complexity, collaboration, versioning, testing, security, integration, and automation may all be issues when implementing Infrastructure as Code (IaC). These obstacles can make it difficult to adopt and operate IaC successfully, but the benefits can outweigh the difficulties, including better efficiency, dependability, and scalability, as well as lower operational costs and improved security.

7. Conclusion

Infrastructure as code (IaC) is a powerful method of managing and provisioning IT infrastructure through the use of code, rather than manual processes. This approach has been gaining popularity in recent years as organisations look to improve their agility and scalability while also reducing errors and increasing efficiency. The literature review showed that there are different tools and technologies such as Terraform and TOSCA that are being used to implement IaC, which have their own advantages and disadvantages. The benefits of IaC include increased productivity, reduced human error rates, and cost optimization. However, organisations should be aware of the potential security vulnerabilities that can arise when using IaC and take necessary precautions. Overall, IaC is an essential aspect of modern IT operations and organisations should consider integrating it into their business models in order to stay competitive in the smart era.

References

[1] Liang-Jie Zhang; Qun Zhou, "CCOA: Cloud Computing Open Architecture," in proceeding of IEEE International Conference on Web Services (ICWS), 2009, pp. 607-616, 6-10 July 2009

[2] Yashpalsinh Jadeja; Kirit Modi, “Cloud Computing - Concepts, Architecture and Challenges” in Proceedings of International Conference on Computing, Electronics and Electrical Technologies [ICCEET], 2012.

[3] Qi Zhang, Lu Cheng and Raouf Boutaba, “Cloud computing: state-of-the-art and research challenges”.

[4] Adem TEPE, Güray YILMAZ, “A Survey on Cloud Computing Technology and Its Application to Satellite Ground Systems”.

[5] Yashpalsinh Jadeja; Kirit Modi, “Cloud Computing - Concepts, Architecture and Challenges” in Proceedings of International Conference on Computing, Electronics and Electrical Technologies [ICCEET], 2012.

[6] Tharam Dillon, Chen Wu and Elizabeth Chang, “Cloud Computing: Issues and Challenges,” in Proceeding of 2010 24th IEEE InternationalConference on Advanced Information Networking and Applications, pp. 27-33, 20-23 April 2010

[7] Xu Wang; Beizhan Wang; Jing Huang, “Cloud computing and its key techniques” in Proceeding of IEEE Conference, 2011

[8] C. Parnin, E. Helms, C. Atlee, H. Boughton, M. Ghattas, A. Glover, J. Holman, J. Micco, B. Murphy, T. Savor, M. Stumm, S. Whitaker, L. Williams. The top 10 adages in continuous deployment IEEE Softw., 34 (3) (2017), pp. 86-95, 10.1109/MS.2017.86

[9] P. Labs, Puppet Documentation, 2017, (https://docs.puppet.com/). [Online; accessed 10-October-2017].

[10] Automating Infrastructure As A Service Michael Howard Computer Science Department Portland State University Portland, USA mihoward@pdx.

[11] Tadej Borovsak ˇ 2 , Elisabetta Di Nitto1 , Michele Guerriero1 , Damian Andrew Tamburri1 1DEEPSE group - DEIB - Politecnico di Milano,” in DevOps: Introducing Infrastructure-as-Code Matej Artacˇ, vol. 2, Milano, Italy 2XLAB - Ljubljana, Slovenia matej.

[12] J. Sandobalin, E. Insfran, and S. Abrahao, “On the effectiveness of tools to support infrastructure as code: Model-driven versus code-centric,” IEEE Access, vol. 8, pp. 17734–17761, 2020

[13] A systematic mapping study of infrastructure as code research Akond Rahman∗ , Rezvan Mahdavi-Hezaveh. Raleigh, NC, US