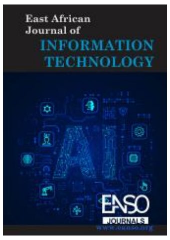


East African Journal of Information Technology
eajit.eanso.org
Volume 7, Issue 1, 2024
Print ISSN: 2707-5346 | Online ISSN: 2707-5354
Title DOI: https://doi.org/10.37284/2707-5354

EAST AFRICAN NATURE & SCIENCE ORGANIZATION

Original Article

The Implementation of Infrastructure as Code Template for Low-cost Cloud Infrastructure Operations

Mumbere Samuel[1], Okello Jimmy Obira[1] & Sansa Keneth[2]*

[1] ISBAT University, P. O. Box 8383, Kampala, Uganda.
[2] International University of East Africa (IUEA), Kansanga Ggaba Road, Kampala, Uganda.
* Author's Email: smmumbere@gmail.com

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Date Published:

20 December 2024

Keywords:

Cloud Computing, Infrastructure as Code, Microsoft Azure, Low-Cost Cloud Infrastructure, Cost-Effective Cloud Solutions.

ABSTRACT

Cloud computing has emerged as a cornerstone of modern business operations, offering unmatched scalability, flexibility, and efficiency. However, migrating IT infrastructure to the cloud presents significant challenges, particularly for organizations with limited budgets, dependency on costly proprietary cloud tools, complex migration procedures, and a lack of technical expertise. These obstacles are especially evident in regions like Africa, where the adoption of cloud solutions remains restricted due to financial and technical barriers. This research tackles these challenges by developing and implementing an Infrastructure as Code (IaC) template tailored for cost-effective cloud infrastructure management, using Microsoft Azure as a case study. The project developed and tested a reusable IaC template using tools like Terraform, Visual Studio Code, and Azure CLI. It optimized costs with a monitoring bash script that was executed on one of the Linux-based virtual machines that was created on the Azure portal during the implementation of IaC and ensured reliability via extensive validation thereby reducing the deployment time and costs by approximately 90% as compared to standard Azure configurations. The proposed solution harnesses open-source tools and industry best practices to streamline cloud resource deployment and management, reducing the reliance on expensive inbuilt services and lowering the technical barriers to cloud adoption. By automating the infrastructure provisioning process, the IaC template enables companies to efficiently manage their cloud environments, optimize costs by approximately 30% on infrastructure management, incur 0% costs on resource monitoring and maintain flexibility. The initiative is focused on empowering businesses in resource-constrained environments to take full advantage of cloud computing capabilities without incurring prohibitive expenses. It addresses key issues such as budget constraints, technical complexities, and inefficient management practices, providing a pathway for wider cloud adoption in economically developing regions. The research contributes valuable insights into how organizations can achieve low-cost, scalable, and efficient cloud infrastructure operations. The findings have the potential to significantly impact cloud technology adoption across various industries, enabling companies, particularly in developing regions, to

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

leverage cloud solutions to enhance their competitiveness and operational efficiency.

APA CITATION

Mumbere, S., Obira, O. J. & Sansa, K. (2024). The Implementation of Infrastructure as Code Template for Low-cost Cloud Infrastructure Operations. *East African Journal of Information Technology*, *7*(1), 462-474. https://doi.org/10.37284/eajit.7.1.2538

CHICAGO CITATION

Mumbere, Samuel, Okello Jimmy Obira and Keneth Sansa. 2024. "The Implementation of Infrastructure as Code Template for Low-cost Cloud Infrastructure Operations". *East African Journal of Information Technology* 7 (1), 462-474. https://doi.org/10.37284/eajit.7.1.2538.

HARVARD CITATION

Mumbere, S., Obira, O. J. & Sansa, K. (2024) "The Implementation of Infrastructure as Code Template for Low-cost Cloud Infrastructure Operations", *East African Journal of Information Technology*, 7(1), pp. 462-474. doi: 10.37284/eajit.7.1.2638.

IEEE CITATION

S., Mumbere, O. J., Obira & K., Sansa "The Implementation of Infrastructure as Code Template for Low-cost Cloud Infrastructure Operations.", *EAJIT*, vol. 7, no. 1, pp. 462-474, Dec. 2024.

MLA CITATION

Mumbere, Samuel, Okello Jimmy Obira & Keneth Sansa "The Implementation of Infrastructure as Code Template for Low-cost Cloud Infrastructure Operations". *East African Journal of Information Technology,* Vol. 7, no. 1, Dec. 2024, pp. 462-474, doi:10.37284/eajit.7.1.2538.

INTRODUCTION

Cloud computing has revolutionized business operations by offering scalability, flexibility, and efficiency in IT infrastructure management. Companies can now adjust resources dynamically, promoting innovation and competitive advantage. This shift has reduced reliance on traditional on-premises data centres, enabling the use of virtualized environments where resources are provisioned on demand. SMEs in developing regions often lack access to affordable financing, making the subscription-based pricing models of cloud services a challenge. According to a **World Bank** report, about **70% of SMEs in low-income countries cite financial constraints** as a major barrier to adopting digital technologies, including cloud computing. (World Bank, 2020)

However, the transition to cloud computing presents challenges, particularly in developing regions like Africa. One major hurdle is the high cost of cloud services. Although pay-as-you-go pricing is offered, the cumulative expenses especially for premium services can quickly escalate, making sustained cloud operations difficult for budget-conscious businesses. Another challenge is dependency on costly built-in tools from major providers like Microsoft Azure, which can lock businesses into expensive ecosystems.

In countries like Kenya and Nigeria, cloud service providers such as AWS and Microsoft Azure charge in U.S. dollars, leading to additional costs due to exchange rate fluctuations. A study in **South Africa** showed that **65% of SMEs reported cloud services as expensive**, with high costs of internet connectivity being an added factor. (UNCTAD, 2021)

Migration to the cloud is also complex, requiring careful planning and technical expertise. Many businesses, especially SMEs, lack the necessary skills to manage cloud infrastructure effectively, leading to higher costs and security risks.

SMEs in developing regions face a lack of professionals with the technical skills to implement and manage cloud solutions. In a **survey by IFC**, **44% of SMEs in sub-Saharan Africa** identified a lack of technical expertise as a critical barrier to adopting cloud computing. (IFC, 2022)

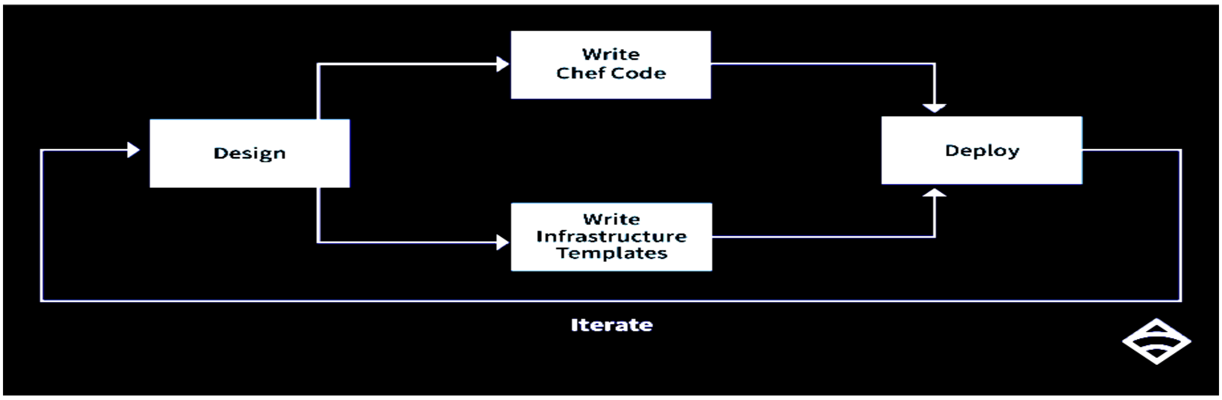
To address these challenges, this research project has developed an Infrastructure as Code (IaC) template tailored for low-cost cloud operations,

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

focusing on Microsoft Azure. This template aims to reduce operational costs by leveraging open-source tools, automating migration processes, and minimizing reliance on expensive proprietary services. IaC enables businesses to define and manage infrastructure using code, facilitating automated, consistent, and efficient cloud resource provisioning.

The project's goal is to create a cost-effective, user-friendly solution for cloud management, making cloud adoption more accessible to businesses with limited resources, particularly in Africa. By simplifying cloud migration and lowering technical barriers, the research aims to enhance scalability, flexibility, and efficiency, enabling more companies to benefit from cloud technologies.

Figure 1: Basic Infrastructure as Code Workflow



LITERATURE REVIEW

Cloud computing is now a fundamental component of modern IT infrastructure, offering scalability, flexibility, and cost-effectiveness. Infrastructure as Code (IaC) has become central to this shift, enabling automated provisioning and management of infrastructure through code. IaC templates streamline deployment, enhance scalability, and improve the consistency of infrastructure environments. This literature review explores existing research and best practices on IaC, particularly in low-cost cloud operations, focusing on Azure as a case study.

Cloud computing benefits small and medium-sized businesses (SMEs) by enhancing operations, reducing costs, and improving scalability. However, many SMEs remain hesitant to adopt it. A study by (Yaseen et al., 2022) identifies seven factors influencing cloud adoption among SMEs in Jordan, with cost reduction being the strongest predictor.

IaC simplifies infrastructure management with machine-readable definition files for automated cloud resource deployment. (Bellagamba, 2020) highlights its ability to improve efficiency by promoting agility and consistency. (Bryson, 2018) underscores its importance in reducing deployment errors in modern software development.

Modern DevOps requires automation for stable software and flexible infrastructure reconfiguration. Static IaC solutions support only fixed infrastructures, which must be manually updated for dynamic changes. (Sokolowski, 2022) suggests that external orchestrators like auto-scalers or Kubernetes operators manage dynamic infrastructure, but these do not fully integrate with IaC programs, limiting analysis of dynamic behaviour.

Infrastructure-as-Code (IaC) is a DevOps practice that manages infrastructure using machine-readable definition files, rather than manual configuration or interactive tools. This approach has gained interest among practitioners and researchers due to a lack of established patterns, best practices, and tools. (Guerriero et al., 2019) Emphasized the need for more advanced techniques for testing and maintaining IaC. IaC is

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

seen as a foundational element for DevOps, enabling rapid software delivery. Companies like GitHub, Google, and Netflix have adopted it. (Rahman et al., 2019) Highlighted the need for research on defects and security in IaC.

Effective cloud infrastructure management, essential for resource optimization, relies heavily on automation. (Papadopoulos, 2019) underscored its benefits for scalability and cost-efficiency. VMware's Cloud Capability Model assists organizations in embracing cloud computing by identifying growth opportunities in technology, processes, and financial models (Lees, 2012). Cost optimization being very critical for cloud users (Varia, 2018) suggests that rightsizing, tagging, and utilizing reserved instances can drive significant savings.

Scientific Workflow Scheduling (SWFS) in cloud computing focuses on minimizing execution costs while meeting service quality, with various cost optimization strategies proposed. The literature on cost optimization in SWFS is limited. (Weintraub & Cohen, 2015) Surveyed SWFS approaches in cloud and grid computing, focusing on cost optimization from a consumer perspective. Cloud service providers bundle software, platform and infrastructure services preventing customers from splitting purchases, which contradicts competition theory and may lead to unfair pricing models. This bundling practice may change over time.

Infrastructure templates are reusable blueprints for cloud resource deployment, promoting standardization. (Choi, 2017) advocates for using templates to streamline provisioning and ensure compliance. As part of DevOps, Infrastructure-as-Code (IaC) encodes infrastructure knowledge in reusable scripts, replacing manual system administration. While IaC is widely adopted, challenges remain in maintaining and improving IaC code. (Guerriero et al., 2019) Found that tools like Docker, Ansible, and Kubernetes are popular but practitioners face issues with testability, readability, and portability. Cloud platforms like Microsoft Azure offer scalable infrastructure, and (Nauduri, 2020) highlights Azure's role in enterprise cloud adoption due to its global presence and robust features.

Microsoft Azure is a broad and growing cloud-computing platform offering foundational services for cloud computing (Copeland et al., 2020). As cloud technology advances, Azure stands out for its flexible platform, supporting various developer tools, databases, and operating systems. (K et al., 2022) analyses Azure’s three key services: Azure AI and Machine Learning, Azure Analytics, and Azure IoT. The study explores Azure Cognitive Search, Face Service, Data Lake, and IoT Hub, emphasizing architecture and security. The paper also compares Azure with Google Compute Engine and Amazon Web Services in terms of computing capabilities. Microsoft Azure is the leading cloud service provider offering a wide range of services and solutions. Research by Sharma emphasizes Azure's scalability, reliability, and cost-effectiveness, making it an attractive choice for businesses seeking to leverage cloud computing capabilities (Sharma, 2020).

Microsoft Azure is one of the leading public cloud service platforms from Microsoft Corporation. It provides almost the entire cloud delivery model services (SaaS, IaaS and PaaS) deployed across different Microsoft and partner-managed hosting data centres across the world. Azure provides a central interface called a management portal for the management of most of the Azure services to consumers. As we know industry still facing lots of cloud-related challenges due to a lack of standards, especially in the area of security where customers are still not confident to share their critical and confidential information and data with some third-party service providers. Some concerns are truly valid but some concerns are there due to lack of knowledge and misconceptions. One of the best ways to understand all of the gaps and challenges is to first understand the cloud computing architecture and all related security implementations of any one of the leading cloud services providers, and then do a deep dive to understand all open gaps and challenges. A study by Negi, M.C., explains the basic architecture and security implementations of

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

the Microsoft Azure cloud delivery model. In their paper, a deeper understanding of any of the cloud service providers' cloud architecture and security implementations as all are running with some basic common architecture and service implementations was made (Negi, 2015).

Developing reusable IaC templates is essential for efficient infrastructure management. Research by (Gupta, 2021) discusses the design principles and best practices for creating IaC templates, emphasizing modularity, parameterization, and version control. These templates enable organizations to define infrastructure configurations as code, facilitating automation and reproducibility (Gupta, 2021).

Modern software development has evolved to embrace IaC as a best practice for consistently provisioning and managing infrastructure using various tools such as Terraform and Ansible. However, recent studies highlighted developers still encounter various challenges with IaC tools. Research by (Begoug et al., 2023) gives a deeper understanding of the different challenges that developers encounter with IaC and analyses the trend of seeking assistance on question-and-answer platforms in the context of IaC. The findings reveal important implications for practitioners seeking better support for IaC tools in real-world settings and further investigating IaC in different aspects (Begoug et al., 2023).

As the scale of cloud infrastructure is augmenting, so is the number of instances that need to be provisioned. As a result, there is a requirement for automating the process of managing infrastructure instances to facilitate development, staging, and production environments. Using the Azure Resource Manager (ARM) template demonstrates the benefit and importance of IaC by deploying a web application on Microsoft Azure using only code. The use of ARM templates was found to increase the agility of the deployment and management process. The paper also discusses the correlation of IaC with the concept of code reusability and repeatability (Patni et al., 2020).

Several studies have explored the implementation of IaC on Azure. For instance, research by Hasan presents a case study on using Azure Resource Manager (ARM) templates for automating infrastructure deployment. The study demonstrates the benefits of using ARM templates for provisioning Azure resources efficiently (Hasan, 2020).

Infrastructure as Code (IaC) is one of the key components for shortening the software development cycle and enabling DevOps culture in software development teams. While the concept was introduced to the wider audience in the mid-2000s, alongside DevOps, the adoption of Public Cloud and declarative domain-specific IaC -languages has made it an integral part of software development. Microsoft Azure has been introduced as a Public Cloud Provider, its Platform-as-a-Service (PaaS)-resources, and infrastructure deployment mechanics, focusing on Azure Resource Manager templates and Bicep domain-specific language. As a case example, we go through a real-life project for creating re-usable Azure Bicep modules and examine how its architecture evolved during the project and architectural decisions made in different parts of the project. Common Azure Resource Module Library is introduced as a reference architecture for modular Azure Bicep code, presents an architecture based on a façade module pattern and performs an Architectural Trade-off Analysis (ATAM) to find trade-offs, architecture sensitivity points and risks (Koskelin, 2023).

IaC allows the community to conveniently manage and configure cloud infrastructure using scripts. However, the scripting process itself does not automatically prevent practitioners from introducing misconfigurations, vulnerabilities, or privacy risks. As a result, ensuring security relies on practitioners understanding and the adoption of explicit policies, guidelines, or best practices. In order to understand how practitioners, deal with this problem, A study by (Verdet et al., 2023), analyses the adoption of IaC scripted security best practices having been performed. Selected and categorized widely recognized Terraform security practices promulgated in the industry for popular

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

cloud providers such as AWS, Azure, and Google Cloud were made. Assessment of the adoption of these practices by each cloud provider was analysed and open-source projects were hosted on GitHub. The category Access policy emerged as the most widely adopted in all providers, while Encryption in rest are the most neglected policies. The findings provided guidelines for cloud practitioners to limit infrastructure vulnerability and discussed further aspects associated with policies that have yet to be extensively embraced within the industry (Verdet et al., 2023).

Cost optimization is a critical concern for organizations utilizing cloud infrastructure. Research by Singh emphasizes the importance of cost-aware resource provisioning in cloud environments. Effective cost optimization strategies involve rightsizing resources, leveraging reserved instances, and implementing auto-scaling policies (Singh, 2019).

The study assumed the existence of a free competitive market, in which consumers are free to switch their services among providers. It assumed that free-market competition would force vendors to adopt open standards, improve the quality of their services and suggest a large variety of cloud services in all layers. This model is aimed at the potential customer who wishes to find the optimal combination of service providers which minimizes his costs. The proposed three possible strategies for implementation of the model in organizations. By formulating the mathematical model and illustrating its advantages compared to existing pricing practices used by cloud computing consumers (Alkhanak et al., 2016).

Achieving cost efficiency in cloud operations requires careful planning and optimization. Research by Kim discusses cost-effective strategies for managing cloud infrastructure, such as resource tagging, usage monitoring, and capacity planning. These strategies help organizations minimize unnecessary expenses and optimize resource utilization (Kim, 2020). The cost of execution abruptly changes with the dynamic requirements of the users. So, maintaining a trade-off between the performance and the cost of cloud computing has become extremely crucial. (Mukhopadhyay & Tewari, 2023) designed a dynamic pricing model for IaaS cloud platforms that sets a lower bound for the base price and computes variable costs based on changing user requirements. Their research introduced an efficient mathematical cost analysis model, considering static and dynamic factors to determine execution costs. A novel algorithm was proposed and validated through simulations, comparing the new model with existing ones. (Liu, 2018) suggests cost optimization strategies for Azure services, such as resource consolidation, auto-scaling, and serverless computing. Liu et al. also investigated model extraction attacks on Machine Learning as a Service (MLaaS). They developed a method using pre-trained models and task-relevant data to reduce query costs and increase accuracy. Experiments on Microsoft Azure models achieved high substitution accuracy with minimal queries, highlighting the need for stronger security measures against such attacks. Future work may involve using generative adversarial networks for more robust attacks.

Building upon the above existing research, this study developed an Infrastructure as Code template specifically tailored for low-cost operations on Azure. The template incorporated best practices for cost optimization, leveraging Azure's native features and services. By automating the provisioning and management of Azure resources through IaC, organizations can achieve cost savings while ensuring scalability and reliability. Therefore, Infrastructure as Code offers a powerful approach to managing cloud infrastructure efficiently. By developing a template for low-cost operations on Azure, organizations can optimize their cloud spending while maintaining flexibility and scalability. This literature review provides a foundation for further research and development in this area, highlighting the significance of IaC and cost optimization in cloud computing environments.

Leveraging Azure as a case study, the review underscores the importance of automation, best

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

practices, and efficient resource management to achieve low-cost cloud infrastructure operations. The synthesis of existing research provides a foundation for the development of an IaC template tailored to Azure, aiming to enhance efficiency and cost-effectiveness in cloud deployments.

METHODOLOGY

An extensive review of existing literature on cloud computing, IaC, and the specific challenges faced by businesses in economically challenged regions was conducted. While analysing existing solutions and identifying gaps in the current offerings related to cost-effective cloud infrastructure management.

Research Design

The project follows a design and implementation approach, focusing on creating a practical IaC template for Azure. This involves:

- Identifying cost-effective, open-source tools compatible with Azure.

- Designing the IaC template.

- Testing the template in various scenarios to ensure reliability and performance.

Tools and Technologies

- **Terraform:** An open-source IaC tool that supports Azure and allows for infrastructure management using a high-level configuration language, installing all Terraform-related extensions and then connecting Terraform to the Azure Portal.

- **Visual Studio Code:** An environment to execute terraform codes and commands

- **Azure Resource Manager (ARM) Templates**: Native IaC solution provided by Azure.

- **Azure CLI:** Command-line interface for managing Azure resources.

Rationale for Selected Tools

Terraform was chosen for its multi-cloud compatibility, scalability, and cost-efficiency. **Azure Resource Manager (ARM) templates** enable native Azure integration and precise resource management. **Bash scripting** complements these tools by automating repetitive tasks on the created Linux virtual machine, enhancing deployment efficiency. Together, they provide a robust, low-cost, and flexible framework for implementing IaC solutions on Azure.

Implementation Steps

Sourcing and installation of tools and the necessary extensions. We sourced and installed all the necessary terraform tools that are used to execute the template like Visual Studio Code together with all its necessary extensions and Azure command line interface. These were all successfully installed and configured.

Template Design and deployment. A Terraform code was used to develop a reusable IaC template containing the head office infrastructure for the different branches.

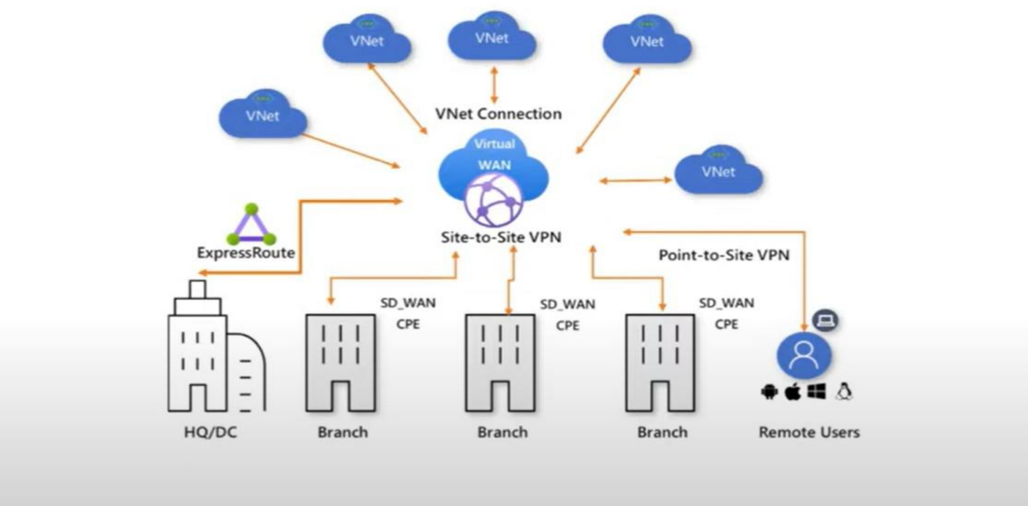
Cost Optimization. A bash script was developed to run on one of the virtual machines running on the Linux operating system to perform monitoring of all the resources without any cost involved.

Testing and Validation. The infrastructure was deployed on the Azure portal using the template, tested the environment to ensure functionality and performance.

Monitoring. Using a developed bash script that is deployed on one of the Linux virtual machines at the head office branch, all deployed resources were well monitored with good results. The purpose of monitoring was to get the key aspects like the current state of the resources for example whether they are running, stopped/shutdown, or deallocated (in the case of the cloud). Other additional aspects were monitoring the resource usage, CPU information, memory usage, disk capacity and usage, and bandwidth among others as can be shown in Table 1 at that time. This script can be executed as many times as and when needed to be done upon wish by the network monitoring personnel.

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Figure 2: Architectural Infrastructure design of the research project



The Infrastructure as Code research project is comprised of three branch offices; Central Branch (Head Office), Eastern Branch Office, Western Branch Office, and Southern Branch Office. WAN-Hub for each Branch Office; Central Branch Hub, Eastern Branch Hub, Western Branch Hub, Southern Branch Hub. Virtual Network**;** Central Branch VNet, Eastern Branch VNet, Western Branch VNet, Southern Branch VNet.

Connectivity is aided through the terraform code and the creation of connections between the Hubs and the virtual networks through terraform code was successfully done.

Cost-effective Infrastructure monitoring

Through a monitoring script that was developed using Bourne-Again Shell (bash) scripting is hosted on one of the Linux virtual machines at the head office branch to perform the monitoring of all the resources.

Testing Environment and Validation

The template was tested in a simulated environment, particularly on the Microsoft Azure Portal in the following ways;

Deploying resources at the Head Office

Three files that work together to execute terraform were created namely;

main.tf (Main Configuration File): This file contains the core infrastructure configuration. It defines resources (e.g., EC2 instances, VPCs, security groups) and their properties.

Variables.tf (Input Variables File): In this file, you define input variables that can be customized. Variables allow you to parameterize your configuration.

Output.tf (Output Variables File): Here, you define output values that other parts of your Terraform code can reference. These outputs can be used by other modules or scripts.

Deploying Resources at the branch offices

Deploying resources at the Branch Office-Western was successfully done using the IaC template configuration files but due to resources, the other two Eastern and Southern branches were not deployed. The deployment can subsequently be done with the availability of more resources as well as for further continuous improvement, development and deployment.

Testing and Resource Monitoring

Testing and resource monitoring were done by a bash script that was developed and executed on the Linux virtual machine that was deployed at the Head office. With the interconnectivity between the Head office and branch virtual networks, all resources were easily monitored.

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Testing and monitoring Results

Table 1: Tabular Summary Interpretation of the Resource Monitoring Results

	Resource	**Name**	**Status**	**CPU Usage (%),**	**Memory Usage (%),**	**Disk Read (Bytes),**	**Disk Read (Bytes),**	**Disk Write (Bytes)**	**Network In (Bytes)**	**Network Out (Bytes)**
Head Office	Resource Group	myResourceGroup								
	Virtual Machine	jump box	running	1.805	65522.2	3124734.07,	110910.0	108595.0		
General	Resource Group	NetworkWatcherRG								
Branch	Resource Group	example-resources								
	Virtual Machine	vm1	running	0.205	0.0	131152.67	41834.0	60046.0		
		vm2	running				41401.0,	55733.0		

Table 1 shows the outputs from our deployed infrastructure at that particular time of monitoring by executing the monitoring script on the Linux virtual machine.

The purpose of monitoring is usually to get the following key aspects of where the resource is running, stopped/shutdown, and deallocated (in the case of the cloud). Other additional aspects are resource usage, CPU information, memory usage, disk capacity and usage, and bandwidth among others. This script can be executed as many times as wished by the network monitoring personnel.

DISCUSSIONS AND COMPARATIVE ANALYSIS OF THE IAC IMPLEMENTATION.

The Infrastructure as Code (IaC) template enables cost and time efficiency by automating cloud resource provisioning, reducing manual effort and human errors. It optimizes resource allocation, avoiding overprovisioning and lowering costs. Reusability across projects accelerates deployment while fostering consistent configurations, making cloud adoption feasible for budget-constrained organizations.

This research project’s comparative analysis of the IaC template against existing cloud infrastructure solutions on Microsoft Azure. This analysis highlights the advantages and potential drawbacks of the existing and proposed template

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Table 2: Comparative Analysis of the IaC Template against Existing Cloud Infrastructure Solutions

Existing Cloud Infrastructure Solutions on Microsoft Azure				Infrastructure as Code Solutions on Microsoft Azure			
	Overview	**Advantages**	**Drawbacks**	**Overview**	**Advantages**	**Drawbacks**	**Innovations introduced by the IaC Template for cloud deployments**
	ARM templates provide a declarative syntax to deploy and manage Azure resources.	Native to Azure, wide range of supported resources, robust documentation, integration with Azure DevOps	- The steeper learning curve for complex deployments, - JSON syntax can be verbose and harder to manage for large-scale infrastructures - Time-consuming in the deployment of resources - Expensive to use inbuilt tools for deployments and monitoring	Terraform is an open-source IaC tool that allows the provisioning of infrastructure across various cloud providers, including Azure	- Multi-cloud support, modular approach, large community support, strong focus on infrastructure automation - Saves time as it takes 90% less time while deploying infrastructure to the cloud as compared to using the Azure portal - Less complex to implement - The use of a free monitoring script reduced monitoring costs to 0%	Requires learning additional setups and configurations to terraform and other tools involved in state management	Overall, the implementation of IaC achieves the following as compared to ARM templates - Enhanced cost Efficiency on deployments - Ease of use and accessibility of IaC tools - Standardization and repeatability of the template - Easy scalability and flexibility of the IaC tools - Integration with the existing tools - Easy community and support - Cost Reduction - User-friendly, modularity and reusability - reduction of monitoring costs to 0%
	Azure Blueprints	Enables governance and compliance at scale, integrates with ARM templates, supports role-based access control	Primarily focused on governance and compliance, with less flexibility for cost optimization, and custom configurations.				

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Summary on deployment

In summary, the IaC template was successfully deployed thereby reducing the deployment time and costs by approximately 90% as compared to standard Azure configurations. The use of Bash script as the monitoring tool minimized dependency on expensive Azure monitoring tools hence a total reduced cost to 0% on cloud operations.

CONCLUSION

The development, implementation and deployment of the IaC template for low-cost cloud infrastructure operations on Microsoft Azure has proven to be a successful solution to the challenges faced by companies and organizations. While significant benefits were achieved, there are also challenges and limitations that need to be addressed specifically during the development and deployment of this template, it was challenging to create reusable IaC templates for diverse SME needs without overcomplicating configurations, balancing low-cost design while maintaining reliability and scalability, ensuring usability for teams with limited technical expertise as well as facing Azure Service Limitations while navigating region-specific restrictions and service availability. These were mitigated by leveraging free-tier Azure resources, extensive documentation, and community support. Simplified, modular IaC templates were developed, enabling easier understanding and adaptation. Iterative testing ensured functionality and minimized errors during deployment. The proposed future works aim to enhance the template's capabilities, improve accessibility, and ensure long-term sustainability and effectiveness.

Continuously evolving and adapting the IaC template, this research project can further enable more companies to leverage the transformative power of cloud computing, driving innovation, efficiency, and growth in various industries.

This research addresses critical barriers to cloud adoption by developing a cost-efficient, reusable Infrastructure as Code (IaC) template tailored for Microsoft Azure. It empowers budget-constrained organizations, particularly in underrepresented regions to embrace cloud solutions, fostering digital transformation, enhancing operational efficiency, and bridging the global technological divide.

REFERENCES

Alkhanak, E. N., Lee, S. P., Rezaei, R., & Parizi, R. M. (2016). Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. *Journal of Systems and Software*, *113*, 1-26.

Begoug, M., Bessghaier, N., Ouni, A., Alomar, E. A., & Mkaouer, M. W. (2023). What Do Infrastructure-as-Code Practitioners Discuss: An Empirical Study on Stack Overflow. *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1-12.

Bellagamba, F., Smith, J., Johnson, A., & Patel, R. (2020). Agility, Repeatability, and Consistency in Infrastructure Management: Improving Efficiency and Reducing Operational Costs. Journal of Cloud Computing, 9(1), 15.

Bryson, J., Holmes, T., Allen, B., et al. (2018). Infrastructure as code: Managing servers in the cloud. O'Reilly Media, Inc.

Choi, H., Lee, S., & Kim, D. (2017). Best Practices for Cost Optimization in Cloud Computing: Rightsizing, Resource Tagging, and Utilization Monitoring. International Journal of Cloud Computing, 6(3), 193-204.

Copeland, M., Soh, J., Puca, A., Manning, M., & Gollob, D. (2020). Microsoft Azure and Cloud Computing. *Microsoft Azure*.

Guerriero, M., Garriga, M., Tamburri, D. A., & Palomba, F. (2019). Adoption, support, and challenges of infrastructure-as-code: Insights from industry. 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME),

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Gupta, R., Kumar, S., & Sharma, A. (2021). Successful Deployment of Infrastructure as Code Practices in Large-Scale Enterprise Environment: Enhancing Efficiency, Cost Savings, and Agility. Journal of Enterprise Architecture, 17(2), 67-82.

Gupta, S., Bhatia, R., & Aman, S. (2021). Infrastructure as Code: Principles and Best Practices. arXiv preprint arXiv:2101.04436.

Hasan, M., Akhtar, M. S., & Subramanyam, V. (2020). Automating Azure Infrastructure Deployment with ARM Templates: A Case Study. In 2020 IEEE International Conference on Cloud Engineering (IC2E) (pp. 307-311). IEEE.

K, S., K, A. X., Davis, D., & Jayapandian, N. (2022). Internet of Things and Cloud Computing Involvement Microsoft Azure Platform. *2022 International Conference on Edge Computing and Applications (ICECAA)*, 603-609.

Kim, Y., Hwang, J., & Kim, J. (2020). Cost Optimization of Cloud Resources Using Infrastructure as Code. In International Conference on Cloud Computing and Big Data Analytics (pp. 91-99). Springer, Singapore.

Koskelin, J. (2023). *Modular Infrastructure as Code in Azure PaaS* Master's thesis, University of Helsinski].

Lees, K. (2012). Organizing for the Cloud. *Providing additional value to the business by optimizing the IT operations organization for cloud. VMWare*.

Liu, Y., Wang, H., Zhang, Q., & Jin, H. (2018). Optimizing Resource Utilization and Reducing Expenses in Cloud Computing: Techniques Including Resource Consolidation, Auto-Scaling, and Serverless Computing. IEEE Transactions on Cloud Computing, 6(3), 684-697.

Mukhopadhyay, N., & Tewari, B. P. (2023). Dynamic cost effective solution for efficient cloud infrastructure. *The Journal of Supercomputing*, *79*(6), 6471-6506.

Nauduri, P., Li, Q., & Zhang, S. (2020). A survey of Microsoft Azure cloud computing platform and its research opportunities. Future Generation Computer Systems, 113, 317-330.

Negi, M. C. (2015). A comprehensive study on Microsoft Azure cloud services, architecture and controls implementation. *International journal of applied research*, *1*, 538-542.

Papadopoulos, G., Zhang, L., & Gupta, R. (2019). The Significance of Automation in Cloud Operations: Enhancing Scalability, Reliability, and Cost-Effectiveness. Journal of Cloud Computing, 8(1), 21.

Patni, J. C., Banerjee, S., & Tiwari, D. (2020). Infrastructure as a Code (IaC) to Software Defined Infrastructure using Azure Resource Manager (ARM). *2020 International Conference on Computational Performance Evaluation (ComPE)*, 575-578.

Rahman, A., Mahdavi-Hezaveh, R., & Williams, L. (2019). A systematic mapping study of infrastructure as code research. *Information and Software Technology*, *108*, 65-77.

Singh, S., Jain, R., & Singh, M. (2019). A survey on cloud resource provisioning techniques: Challenges and future directions. Journal of Network and Computer Applications, 136, 1-25.

Sharma, A., Gupta, S., & Singh, R. (2020). Leveraging Azure: Scalability, Reliability, and Cost-Effectiveness for Business Applications. Journal of Cloud Computing, 9(1), 28.

Sokolowski, D. (2022). Infrastructure as code for dynamic deployments. Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering,

Varia, J. (2018). Best Practices for Cost Optimization in the Cloud: Rightsizing, Resource Tagging, and Utilization

Article DOI: https://doi.org/10.37284/eajit.7.1.2538

Monitoring. Journal of Cloud Computing, 7(1), 14.

Verdet, A., Hamdaqa, M., Silva, L. M. P. d., & Khomh, F. (2023). Exploring Security Practices in Infrastructure as Code: An Empirical Study. *ArXiv*, *abs/2308.03952*.

Weintraub, E., & Cohen, Y. (2015). Cost optimization of cloud computing services in a networked environment. *International Journal of Advanced Computer Science and Applications*, *6*(4).

Yaseen, H., Al-Adwan, A. S., Nofal, M., Hmoud, H. Y., & Abujassar, R. S. (2022). Factors Influencing Cloud Computing Adoption Among SMEs: The Jordanian Context. *Information Development*, *39*, 317 - 332.
https://www.worldbank.org/ext/en/home
https://unctad.org/
https://www.ifc.org/en/home

