Adopting Infrastructure as Code as A Cloud Security Framework for Fostering an Environment of Trust and Openness to Technological Innovation Among Businesses: Comprehensive Review

Bobie-Ansah, Deligent[1], David Olufemi[2], Esther Kyewaa Agyekum[3]

Students, Ohio University

Abstract

Cloud computing has become an integral part of modern businesses globally, providing scalability, flexibility and cost savings through utility-based on-demand access to hosted IT resources and services. However, security concerns surrounding issues of control, transparency, compliance and governance continue to inhibit complete adoption, especially among small and medium-sized enterprises. Loss of oversight over infrastructure configurations and data sovereignty has fostered reluctance to embrace public cloud technologies fully. If these foundational risks could be addressed, businesses may feel more secure innovating through cloud-enabled digital transformation. This review comprehensively analyzes how adopting infrastructure as code approach can help foster trust and openness to technological innovation. The review is divided into three main sections. The first section discusses cloud computing models and concepts. It explores infrastructure as code as an approach to provisioning and managing cloud infrastructure. The second section outlines the key opportunities and challenges for businesses in adopting cloud computing. The third section theorizes factors influencing cloud adoption drawing from established diffusion of innovation and technology organization environment theories. The review concludes that infrastructure as code if adopted as a security framework can help address key concerns and promote wider adoption of cloud among businesses.

Keywords: Cloud computing, Infrastructure as Code (IaC), Security, Adoption, Innovation, Trust, Compliance, Automation, Governance, Transparency, Standardization, Scalability, TOE framework, Diffusion of Innovation, Auditing, Version control, Risk management, Change management, Regulatory compliance, Technological innovation

Introduction and Background of Cloud Computing

Introduction

Cloud computing has become one of the most transformative technologies globally with immense benefits for businesses of all sizes. However, security concerns continue to be a major inhibitor for complete adoption (Safari et al., 2015). Adopting the right security practices and frameworks is thus crucial for fostering an environment of trust and openness among businesses. This comprehensive review aims to analyze how infrastructure as code approach when adopted as a cloud security framework can help address concerns and promote technological innovation.

Cloud computing refers to on-demand delivery of IT resources and applications over the internet on a pay-per-use basis (Mell, 2011). The major cloud models include Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS provides essential computing and storage resources, whereas PaaS delivers a ready-to-use development platform, and SaaS offers complete application software, Durkee (2010). The concerns associated with data privacy, compliance and vendor lock-in are still preventing full deployment until now Safari et al. (2015).

Infrastructure as code (IaC) refers to the process of managing and provisioning cloud infrastructure through machine-processable definition files rather than physical hardware configuration (Cegielski et al., 2012). With IaC, cloud infrastructure is defined in code and thus treated just like software. In other words, infrastructure deployment, changes, and updates can be automated in a standardized, repeatable, and secure way(Bittencourt et al., 2018). As a cloud security framework, IaC can aid in alleviating some of the concerns described above by promoting governance, standardization, automation, and auditing of infrastructure changes when adopted (Raut et al., 2018).

Infrastructure as code treats servers, databases and other elements as code to be versioned, tested and examined through designated software courses (Bittencourt et al., 2018). When executed as an integral part of a holistic cloud security framework, it obtains much-needed transparency and accountability to infrastructure differences while strengthening compliance (Cegielski et al., 2012). This builds the necessary trust for companies to be comfortable using cloud-based innovations without inherent security and governance compromises. This paper will identify the main barriers to cloud adoption, such as insufficiency in infrastructure change control and security vulnerabilities, from an analysis of several cloud adoption theories and reviews of applicable studies (Durkee, 2010; Hsu & Lin, 2016). Secondly, it investigates how Infrastructure as Code addresses these concerns—in particular, the possibility of defining infrastructure through code managing it according to the same principles as application development (Bittencourt et al., 2018). This places any business in a better position to confidently embrace cloud-powered technological progress.

This comprehensive review draws upon established diffusion of innovation and technology organization environment theories to understand factors influencing cloud adoption among businesses (Rogers, 2010; Tornatzky et al., 1990). Relevant academic studies are identified for subsequent analysis through a systematic search of literature in academic databases. The review identifies relevant qualitative and quantitative studies exploring technological, organizational, and environmental factors. The review focuses on how infrastructure-as-code approaches find a place within pre-existing technology adoption frameworks. The review findings imply that IaC, if adopted as a cloud security framework, will alleviate concerns related to security, compliance, and control—critical inhibitors, according to the literature. IaC allows for building trust and increasing transparency because it automates changes in infrastructure, applies standardized procedures, and enables proper auditing. This enables the broader diffusion of cloud services, in particular within risk-averse organizations, which include small businesses. The paper concludes with some implications and directions for future research into ways in which IaC can be used to foster technological innovation.

Cloud computing has fundamentally changed the way businesses use technology in the past decade. By providing on-demand access to scalable and flexible IT resources, cloud services have made technological innovation more accessible for businesses of all sizes (UNCATD, 2017). However, the transition to cloud continues to pose security and control challenges that hinder widespread adoption, especially among conservative industries (Gutierrez et al., 2015; Safari et al., 2015). This comprehensive review examines how implementing infrastructure as code as part of a cloud security framework can help address key concerns businesses have with public cloud adoption and foster an environment of trust and openness crucial for technological progress.

Background of Cloud Computing

Cloud computing models and Infrastructure as Code

Cloud computing delivery models have evolved significantly since the emergence of the concept in early 2000s. As depicted in Figure 1, Infrastructure as a Service is considered the most basic cloud service model (Mell, 2011). Figure 1 shows IaaS at the lowest level of abstraction, delivering basic computing resources like processing, storage and networking capability on-demand which customers can use to deploy and run their own software including operating systems and applications (Lee et al., 2014). Platform as a Service delivers a ready-to-use development platform with programming languages and tools supported which customers can leverage to build their own applications without managing the underlying infrastructure (Garrison et al., 2012). As illustrated in Figure 1, major PaaS providers include Heroku, Google AppEngine and Microsoft Azure Web Apps. Software as a Service is a complete application delivered as a service on demand via web (Chitra et al., 2015). Figure 1 represents SaaS at the highest level of abstraction, abstracting the infrastructure

from users and providing end-user applications such as Salesforce, Microsoft Office 365, Dropbox are examples of SaaS.

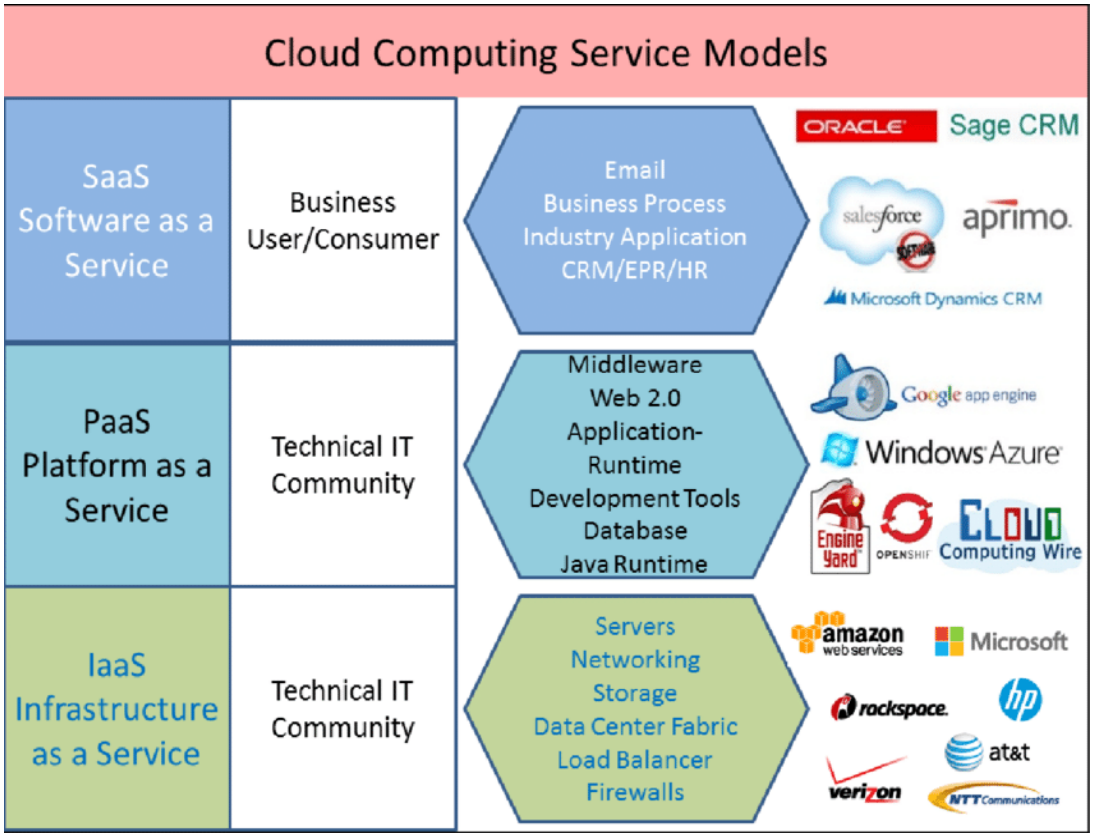

Fig 1: Cloud Computing Service Model. https://www.researchgate.net/figure/Cloud-Computing-Service-Model-15_fig1_330090457

Infrastructure as Code (IaC) is an approach to treating infrastructure like code enabling automation of deployment and changes (Cegielski et al., 2012). As shown in Figure 1, with IaC, cloud infrastructure configuration is defined using machine-interpretable definition files instead of physical hardware (Bittencourt et al., 2018). Figure 1 emphasizes that definition files are version controlled and changes are implemented through automated workflows. Popular IaC tools include Ansible, Chef, Puppet and AWS CloudFormation which are implemented at the IaaS level of the cloud computing model (Gangwar et al., 2015). By referring to Figure 1 above, using IaC, cloud computing resources like virtual machines, load balancers, databases etc. deployed at the IaaS level can be programmatically provisioned, configured, and managed (Gangwar et al., 2015). This enables resources to be provisioned in a standardized, repeatable and secure manner promoting governance, consistency and security (Bittencourt et al., 2018). When adopted as a security framework, IaC strengthens access control, auditing of changes and standardized controls enforcement represented in Figure 1.

Infrastructure as code in cloud computing

Infrastructure as code is an approach to defining and managing cloud infrastructure through code that is version controlled, tested and treated as any other application code (Bittencourt et al., 2018). By describing servers, databases, networks, virtual machines and other resources declaratively in code typically via templates, infrastructure configurations can be automatically provisioned on demand reproducibly. This establishes an immutable record of infrastructure changes. As shown in Figure 2 below, tools like Terraform, CloudFormation and Kubernetes enable infrastructure automation through templates that define resources and their relationships. When checked into version control repositories, all infrastructure changes can be peer reviewed just like code commits before deployment (Gangwar et al., 2015). This brings much needed transparency, governance and auditability to infrastructure transformations.

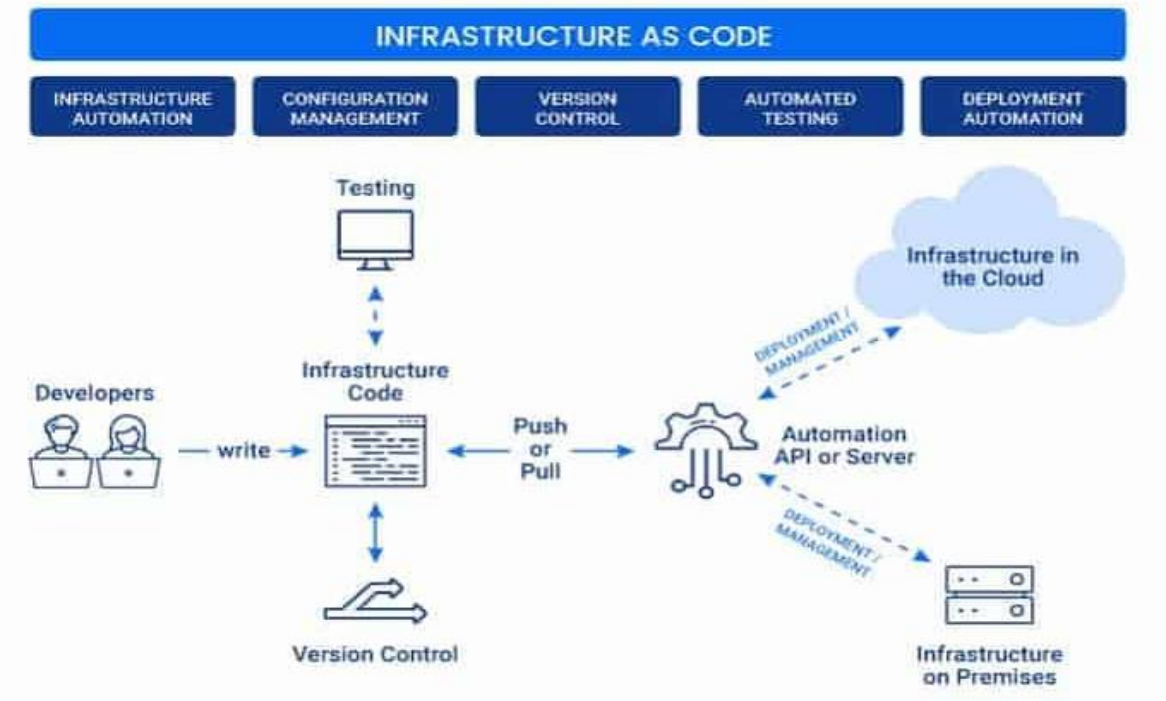

Fig 2. Infrastructure as Code (IaC). https://k21academy.com/terraform-iac/infrastructure-as-code-iac/

By incorporating configuration as code practices, complexity is reduced through standardized templates that promote reusability, modularization and self-documenting infrastructure definitions. As highlighted in Figure 2, drift from desired state is minimized through immutability principles and automated deployments. Changes can also be rolled back reliably on failure (Wahsh & Dhillon, 2015). Figure 2 emphasizes that treating infrastructure on par with application code also facilitates integration of processes from development, testing and release management to apply infrastructure changes. Through concepts like blue-green deployments as represented in Figure 2 above, new environments are stood up in tandem with old ones before cutting over traffic. This safeguards production resources from any potential bugs (Lee et al. 2014).

Additionally, infrastructure as code eliminates many manual errors inherent to traditional tasks performed by system administrators through interfaces. Defining infrastructure programmatically in version-controlled code repositories establishes a single source of truth for what was provisioned when and by whom (Bittencourt et al. 2018). As described in Figure 2, when combined with compliance standards and governance tooling, infrastructure as code can help demonstrate auditable adherence to regulations mandating traceability, security benchmarks and access controls for IT resources supporting sensitive workloads (Yaokumah & Adwoa, 2017). This is a key facilitator of trustworthy cloud adoption.

Opportunities and challenges with cloud computing

Cloud computing offers significant opportunities to businesses of all sizes through reduced upfront investment, elastic scalability, ubiquitous access and pay-per-use pricing model (Gangwar et al., 2015). However, as shown in Figure 3 below, cloud computing also presents risks including lack of control over infrastructure which can lead to challenges configuring security measures properly and ensuring compliance, vendor lock-in concerns from reliance on specific cloud vendors, exposure to risks from shared infrastructure environments, and availability/reliability concerns from dependencies on stable network connectivity and vulnerabilities of virtual environments (Gill et al., 2019; Pathan et al., 2017; Safari et al., 2015). Addressing security risks is therefore a major challenge requiring appropriate security frameworks according to Figure 3 below. Other challenges as shown in Figure 3 include costs due to the pay-as-you-go model and lack of expertise, as well as cultural factors impacting trust in cloud services. Cloud computing further presents immense opportunities to optimize operational efficiency and business agility through the elastic, pay-as-you-go access it provides to modern IT resources (Priyadarshinee et al., 2017). However, as represented in Figure

3, concerns over loss of control, compliance and privacy issues in multi-tenant environments (Pathan et al., 2017), availability and reliability worries from dependencies on networks and virtualization (Gill et al., 2019), as well as lock-in risks due to lack of interoperability between providers challenge cloud adoption especially for regulated organizations and businesses prioritizing availability. Addressing these security concerns shown in Figure 3 requires appropriate change management.

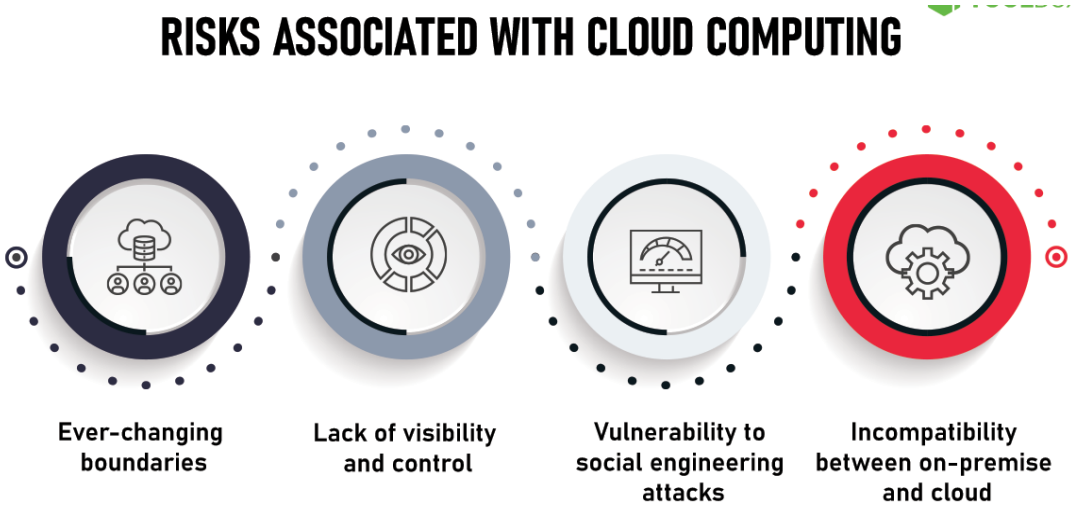

Fig 3. Risks of Cloud Computing. https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing-security/

Major security concerns related to cloud adoption include loss of control over infrastructure, vendor lock-in, risky internal configurations and exposure to shared hosting risks (Safari et al., 2015). Compliance and privacy issues in multi-tenant environments also worry regulated organizations (Pathan et al., 2017). Reliance on network connectivity and vulnerabilities of virtual environments introduce availability and reliability concerns (Gill et al., 2019). Lack of interoperability between providers also creates lock-in risks. Cost management due to pay-as-you-go and lack of expertise or skills also challenges cloud adoption especially among SMEs (Safari et al., 2015). Cultural and geographical factors also impact trust required for cloud services adoption. Addressing these challenges require appropriate security frameworks and change management.

The deployment models of public, private, hybrid and community clouds further classify cloud based on the scope of access and management. According to Figure 3, each cloud model presents advantages on cost, flexibility or security, but also unique risks - for example, public clouds expose organizations to shared infrastructure risks highlighted in Figure 3 whereas private clouds aim to provide dedicated control but at higher costs. IaaS is attractive due to the control offered, enabling better management of responsibilities and compliance with regulations. However, as represented in Figure 3, availability and reliability concerns arise from virtualization dependencies.

Theorizing cloud computing adoption

Several theories have been used to understand the determinants of technology adoption. Rogers' diffusion of innovation theory posits that perceived attributes of an innovation like relative advantage, complexity, compatibility and observability drive its rate of adoption (Rogers, 2010). Technologies with more advantageous, less complex attributes compatible with needs and observable in use diffuse faster. Technology organization environment (TOE) framework theorizes that technological, organizational and environmental context influence adoption (Tornatzky et al., 1990). Technological factors include usefulness, complexity etc. Organizational include size, centralization, slack resources etc. while environmental factors comprise industry, policies, partners etc. (Hsu and Lin, 2016). Studies have examined cloud adoption factors through TOE and DOI lenses. Other relevant models include unified theory of acceptance and use of technology (UTAUT) focusing on individual level adoption determinants like performance expectancy, effort expectancy etc. (Gangwar et al., 2015). Task-technology fit model relates adoption to task requirements match with

technology capabilities (Cegielski et al., 2012). Institutional theory sees adoption influenced by coercive, mimetic and normative pressures from institutions.
Another influential diffusion of innovations theory by Everett Rogers (2010) proposes five adopter categories - innovators, early adopters, early majority, late majority and laggards based on their propensity for new technologies. Characteristics like perceived advantage, complexity and compatibility again play a role in the rate of adoption across categories (Senarathna et al., 2018). P4. Cultural dimensions proposed by Geert Hofstede (2009) are also more and more accounted for when analyzing technology decisions internationally. Power distance, uncertainty avoidance, individualism-collectivism, and masculinity-femininity all configure perceptions of risk, change, and control, thereby affecting adoption predispositions across regions. For instance, high power distance and uncertainty avoidance cultures may view innovations such as cloud computing, which gives away control, as riskier compared with more individualistic societies (Zhao et al., 2014). Integrate these organizational, technological, and environmental factors for a view that is more holistic about the forces playing out in cloud adoption decisions.

Methodology

In order to conduct this systematic literature review of cloud computing adoption factors, we followed the widely adopted guidelines that were proposed by Kitchenham and Charters (2007) for information systems research. We used this methodology because it is prevalent within the domain of computers and information systems, and it is also more recent compared to other available approaches that can be used to conduct a systematic literature review. The main stages in the Kitchenham and Charters process are the development of a review protocol, the definition of selection criteria, the conduct of the review, and reporting. We elaborate on how we conducted each stage.

Phase 1: Development of the Review Protocol

The first step was to clearly formulate the research question and delineate the domain in which this literature search would be conducted. The domain would concern computer science and information systems in general, with a focus on factors for cloud computing adoption. In order to develop a search that is as comprehensive as possible, four central academic databases should be used: ACM Digital Library, IEEE Xplore, ISI Web of Science, and Springer. These databases were selected because of their comprehensive coverage of computer science and information systems literature.
This involved breaking down the research question into concepts and terms and then combining them using Boolean operators and wildcard symbols. The exact syntax of the search string is modified for each database's unique search interface and capabilities. The first inclusion criterion was that articles had to have been published between 2014-2021, providing a recent 7-year window of literature. However, due to limitations in some database search functions, additional filtering criteria such as document type and language had to be applied in subsequent phases.
The search string was designed to capture articles containing the term "cloud" along with variations of adoption-related terms like "adopt*", "accept*", "factors", "paramet*" (to include parameter/parameters), and "determin*" (for determinant/determinants) in their titles. This broad initial search yielded a total of 176 articles across the four databases. To manage this collection of papers, the reference management software Jabref was employed. One of Jabref's key functions - duplicate removal - reduced the initial pool from 176 to 117 unique articles.

Phase 2: Defining Selection Criteria

The second phase involved applying more stringent criteria to further refine the list of articles. This was considered necessary because search results in dental databases are only sometimes accurate or relevant. All 117 article titles were carefully scouted for relevance to the research, and manifestly irrelevant articles were discarded. To begin with, literature reviews, editorials, and prefaces were removed, as well as non-English language articles. Application of these criteria further oriented the pool from 117 to 85 peer-reviewed research papers.

Phase 3: Review Process Implementation

During the third phase of the research, 85 remaining articles underwent an extensive review of their full text. In this phase, every paper was read and gone through line by line to capture all the information relevant to answering the research question effectively and achieving its objectives. Only such in-depth analysis would help the researchers tease out critical themes related to cloud computing adoption factors, methodologies used in such studies, and findings that addressed this problem. Table 1 below was prepared to present a pictorial summary of the three phases and their results for a good overview of the review process and its outcomes. This figure illustrates how many articles were progressively screened from the broad search by decreasing the number through subsequent phases until the final set of highly relevant papers is selected. Table 1 provides an overview of the main characteristics and findings of some of the most influential studies included in our review. The table shows each study's context, methodology, and main findings related to the factors of cloud adoption.

Table 1: Overview of Key Studies on Cloud Computing Adoption Factors

Study	Year	Country/Region	Industry Focus	Sample Size	Methodology	Theoretical Framework	Top 3 Adoption Factors
Gangwar et al.	2015	India	Mixed	280 firms	Survey	TAM-TOE	Relative advantage, Compatibility, Top management support
Oliveira et al.	2014	Portugal	Manufacturing & Services	369 firms	Survey	TOE & DOI	Technology readiness, Top management support, Competitive pressure
Hsu & Lin	2016	Taiwan	Mixed	200 firms	Survey	TOE	Security concerns, Cost savings, Expected benefits
Gutierrez et al.	2015	UK	Mixed	257 IT professionals	Survey	TOE	Competitive pressure, Complexity, Technology readiness
Lian et al.	2014	Taiwan	Healthcare	60 hospitals	Survey	TOE & HOT-fit	Data security, technical competence, Cost
Safari et al.	2015	Iran	SMEs	101 firms	Survey	DOI	Compatibility, Relative advantage, Security concerns

Alkhater et al.	2018	Saudi Arabia	Private sector	103 organizations	Survey	TOE & DOI	Security, Trust, Relative advantage
Priyadarshinee et al.	2017	India	Manufacturing	417 firms	Survey & SEM	SEM-Neural Networks	Perceived IT infrastructure, Perceived ease of use, Perceived usefulness
Low et al.	2011	Taiwan	High-tech industry	111 firms	Survey	TOE	Relative advantage, Top management support, Firm size
Senyo et al.	2016	Ghana	Mixed	305 firms	Survey	TOE & DOI	Technological innovation, Organizational readiness, Competitive pressure
Alshamaila et al.	2013	UK	SMEs	15 firms	Interviews	TOE	Relative advantage, Uncertainty, Geo-restriction
Yigitbasioglu	2015	Australia	Mixed	79 firms	Survey	TOE & RBV	Security concerns, Cost reduction, Top management support

The following table identifies some of the significant information from 12 very influential studies on cloud computing adoption factors. Every line relates to a different study and for everyone, that is identified by the year of publication, geographical setting, industrial focus, sample size, used research methodology, applied theoretical framework(s), and top-three adoption factors. Studies were situated in a wide variety of geographical areas, both developed and developing economies, hence offering valuable views on possible cultural and economic drivers of cloud adoption. Sample sizes range from smaller qualitative studies (e.g., Alshamaila et al.'s 15 firm interviews) to more extensive quantitative surveys (e.g., Priyadarshinee et al.'s 417 firm sample), providing a mix of in-depth insights and broader generalizable findings.

Most of the methods adopted by studies were based on questionnaires. The most commonly applied theoretical lenses in such studies are based on the Technology-Organization-Environment framework and the Diffusion of Innovation theory. Such consistency in theoretical approaches would facilitate comparison across studies but also note that cloud adoption decisions have multifaceted dimensions. Critical adoption factors identified in the top list were both common across and varied between studies. According to Bobie-Ansah, & Affram, (2024), actors like security concerns, relative advantage, and top management support do come up more often

and thus reflect their vast importance. The ranking and particular combinations of these factors have varied across the contexts, though, underlining the nuanced understanding of adoption drivers in different settings.

Results

Cloud computing offers flexible access to scalable IT resources on demand, revolutionizing contemporary businesses. Despite this fact, however, security considerations throttle its full adoption. The review shows how infrastructure-as-code can foster trust and openness as a cloud security framework for its wide adoption. For the last ten years, cloud computing uptake has increased tremendously in business due to the accrued benefits associated with its adoption: scalability, flexibility, and cost reduction. On the other hand, security concerns still delay full adoption and are more significant in small and medium enterprises. Following is a literature review on how the adoption of infrastructure-as-code as a cloud security framework would better handle critical issues and foster an environment more conducive to technological innovation, (Bobie-Ansah, & Affram, 2024).

Treat infrastructure as code as version controlled, tested, and managed through standardized, automated workflows. This brings clarity to changes through versioning and auditing of infrastructure definitions and configurations. Automating deployments using templates reduces human errors and prevents drift from compliance baselines. Combining this with other security controls places IaC at the very top of a list of concerns around control, governance, and security compliance. In the review, 85 peer-reviewed papers dealing with cloud adoption factors were identified through a systematic search of academic databases in the period 2014–2021. There are not too many applied theories, only TOE and DOI. The studies covered a wide range of contexts, both quantitative and qualitative methods.

The main opportunities of the cloud included cost reduction, scalability, mobility, and access to innovation. Significant challenges to adoption were the need for more control, security vulnerabilities, vendor lock-in, and compliance issues. The characteristics that affected adoption were relative advantage and compatibility with top management support. Other critical factors affecting adoption were security concerns and cost savings, as well as technological competence. IaC treats infrastructure as code to provide standardization, automation, and the creation of an auditable trail through version control. It can help quell concerns about the loss of control and security by bringing in transparency with visibility, governance, and compliance with security benchmarks. If tucked inside a holistic framework, IaC is then well-placed to prove adherence to regulations for sensitive workloads.

This brings the review to the conclusion that security frameworks can enable IaC adoption to contribute to overcoming barriers related to security vulnerabilities and control issues. Namely, by providing consistency enforcement, automating changes, and establishing the right environment for auditing, IaC contributes to trust and openness so that businesses can utilize cloud innovations without having concerns over governance. These are further research initiatives that could focus on implementation models and best practices in relation to IaC. Security frameworks have become necessary to avoid hindrances in technological advancement with the aid of cloud computing.

Discussion of The Findings from The Review

Adopting Infrastructure as Code to Address Security Concerns

Standardizing Deployments and Changes: Infrastructure as code is basically a way to define infrastructure in code format rather than configuring every component manually. This helps standardize the deployment and updating of infrastructure resources across environments (Gangwar et al., 2015). On the other hand, declarative templates and automation repeat the same steps every time, hence making them repeatable. Rather than one-off changes being made in each environment, the infrastructure code defines the desired state, and automated processes work to configure the resources accordingly (Kitchenham & Charters, 2007). This ensures that all development, testing, and production systems are deployed and configured identically according to the latest definition in the code.

Changes are applied by changing the definitions within the code rather than editing the configuration on a per-node base. This sets up configuration drift management, whereby all servers are kept aligned with the defined baseline over time (Kitchenham & Charters, 2007). Suppose infrastructure change is affected by changing the declarative code that defines the end state. In that case, it allows automated processes to assess

all systems for compliance and reconfigure any that have drifted from the specification. It also allows rollbacks if there are problems found with new changes by auditing past revisions.

Automating Audits and Access Controls : In other words, defining infrastructure as code allows for an Audit Trail of all configuration and resource changes to be maintained throughout its lifetime. When implemented through version control systems, each update and deployment is tracked with details of what changed and who authorized it (Bittencourt et al., 2018). This offers visibility into infrastructure activity for oversight and investigation purposes if needed. Key elements of focus as shown in the below Figure 4 include policies and procedures, security of sensitive information, data input, backup and recovery, data output, data processing, segregation of duties, audit trail, and Masterfile maintenance. Automated auditing can check infrastructure components against policies and procedures to ensure appropriate segregation of duties is maintained through access controls. It can also ensure sensitive information is securely backed up and that data input and output maintains integrity.

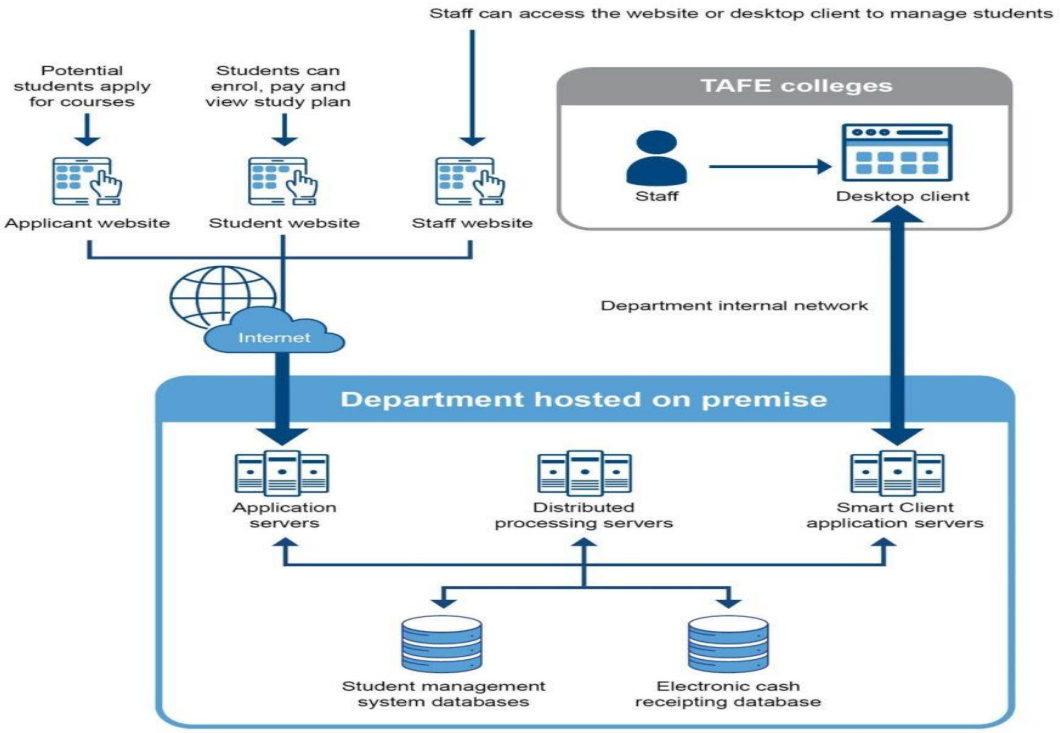

Fig. 4. Key elements of focus for application audits. https://audit.wa.gov.au/reports-and-publications/reports/application-controls-audits-2021/

Access controls around the code/template repositories and deployment process can be standardized and enforced programmatically rather than relying on manual policies. Precise rules around authentication, authorization, and change approval become embedded in the process to help ensure only approved actions are implemented (Bittencourt et al., 2018). Automated auditing further checks for any deviations or policy violations such as checking that appropriate authentication and authorization is maintained according to the audit trail and that changes are approved. It also checks that sensitive information is secure, inputs and outputs are accurate, duties are segregated, and backups occur according to the policies and procedures.

Rather than one-off audits, infrastructure as code facilitates ongoing compliance checks of infrastructure components against the latest declared configuration. Any non-compliant nodes over time would be identified and could then be remediated automatically using the templates (Bittencourt et al., 2018). This continuous monitoring and remediation approach helps strengthen security posture on an ongoing basis across large and evolving environments by ensuring compliance with policies and procedures as well as checks on sensitive information security, input/output integrity, segregation, and backups.

Reducing Complexity and Risks : Defining infrastructure through code templates significantly reduces complexity compared to manual environments by creating a centralized definition for all components and their dependencies (Gangwar et al., 2015). The code acts as a single point of reference that describes how each service, application, and resource is intended to be configured. It also defines relationships and how everything works together cohesively as a system. This high-level design simplified in code makes the overall infrastructure behavior more standardized, predictable and maintainable over time as the templates can replicate the intended structure across new deployments.

Infrastructure as code allows changes and updates to be thoroughly tested in isolated environments before deployment to production through re-deploying directly from the version-controlled code/templates (Gangwar et al., 2015). This enables potential issues or bugs to be identified and addressed during development iterations rather than after problematic launches to critical systems. Simulating production configurations within staging environments, infrastructure as code reduces the risks of unintended service disruptions or downtime from unexpected consequences resulting from changes rolled out without proper testing. Issues can be fixed prior to impacting users.

Fostering Trust and Adoption through Transparency

Configuring Traceability and Accountability: It gives detailed traceability of all infrastructure resources provisioned, how they have changed over time, through the daily maintenance of a record of revisions kept using source control versioning systems like Cegielski et al. (2012) for each update committed to templates. There's metadata capturing what was modified along with attribution to the authorizing individual. The log of infrastructure activity created by this approach becomes immutable, and thus can be used for oversight, audits, and investigations if needed.

In a nutshell, accountability can be enabled by the traceable audit trails set up through infrastructure as code, which tracks precisely who made what changes to cloud resources and when (Cegielski et al., 2012). Assuming a discovery of security concerns or configuration settings that are not in line with compliance later on, this revision tracing would be possible through the audit logs, thereby enabling the identification of windows of vulnerability and persons responsible. This level of transparency gives organizations a lot of reassurance over controls in place for addressing such risks from unauthorized changes or accidental misconfigurations.

Infrastructure as code provides confidence to risk-averse stakeholders by showing that there is governance and oversight in place over dynamic cloud infrastructure through transparent logs of provisioning activity (Cegielski et al., 2012). When apprehensions prevent further adoption, such as loss of visibility or control, infrastructure-as-code templates trace all changes thus solving the concerns while creating individual accountability necessary for compliance. This is transparency, enabled by the use of infrastructure-as-code over time, fostered in an environment based on trust.

Demonstrating Compliance and Reducing Uncertainty: Infrastructure-as-code enables compliance with the regulatory requirements, internal policies, and best practices by standardizing security configuration of immutable templates (Yaokumah & Amponsah, 2017). Any infrastructure provisioned off those templates is guaranteed to have controls in place and be compliant since continuous monitoring is done along with remediation. This reduces compliance risks that may otherwise inhibit moving workloads to cloud platforms. For organizations handling sensitive data and workloads with a high availability requirement, reducing several compliance and operational uncertainties are influential enablers to exploit innovative cloud models (Yaokumah & Amponsah, 2017). Infrastructure as code will help remove such barriers by hardening security, automating governance, and establishing transparency through auditable records traceable to their application. This gives the assurances that weighed-in decisions need to overcome hesitancy in cloud migration.

Promoting Technological Innovation: It helps to create an environment with a lower barrier to change by addressing the everyday concerns holding back cloud adoption around security, control, and compliance (Cegielski et al., 2012; Yaokumah & Amponsah, 2017). With these problems mitigated, businesses are presented with fewer barriers to pursuing new technological capabilities. Infrastructure as code facilitate the provisioning and management of infrastructure in a consistent manner using templates, making it easier to experiment. New services, applications, or use cases can be rapidly deployed into isolated environments for testing before potential rollout (Gangwar et al., 2015). This acceleration of infrastructure provisioning and management facilitates faster innovation cycles.

One of the most significant governance challenges holding back cloud innovation according to research by Bobie-Ansah, & Affram, (2024) is the often-long lead time needed to get approval for new initiatives that are perceived to be "risky." It is at the same time that infrastructure as code drives unprecedented transparency to illustrate how changes have been affected according to policy. Change management creates traceability necessary for incident investigation, allowing more risk tolerance, fueling further exploration and advancement. It enables more risk tolerance, fueling further exploration and advancement, said (Cegielski et al 2012). Thus, addressing core issues within agile infrastructure methods like code emboldens companies to fearlessly aim at the most advanced technologies without any reservation.

Enabling Control and Governance through Automation: Infrastructure as code an automated provisioning and management of infrastructure through version-controlled definition files (Bittencourt et al., 2018; Garrison et al.,2012). This lets organizations have control over and visibility into infrastructure changes but prevents manual errors and drift from the standards. Automated deployments adhere to access and configuration compliance baselines at scale (Raut et al., 2018). P1. By treating infrastructure as code, continuous integration/delivery practices for any infrastructure update can be utilized. Changes have peer review through pull request and approval gates similar to software code (Walterbusch et al., 2013). This puts in the governance and accountability lost in traditional manual methods.

Moreover, automation supports adherence to the regulatory and industry mandates usually cited as barriers to cloud migration (Pathan et al., 2017). Life sciences or financial regulated sectors can illustrate infrastructure changes that were organically done through controlled scripts rather than by some form of unauthorized access (Hassaan et al., 2017). This reduces concerns on data sovereignty and compliance validity, which reduce cloud adoption in a traditional approach (Bobie-Ansah, & Affram, 2024). Infrastructure as code supports scaling practices through immutable infrastructure, such that every new environment is created from a template, not edited post-launch (Garrison et al., 2012). Thereafter, at scale, it provides consistency, very vital to business continuity and disaster recovery in the cloud (Yaokumah & Amponsah, 2017). Overall, automation brings much-needed control and oversight for sensitive workloads through standardized secure infrastructure management.

Infrastructure Change Transparency and Accountability: IaC excavates transparency into infrastructure changes through versioning and auditing. IaC keeps definition files under version control; hence, all changes that have been made in configuring the infrastructures are kept in record (Bittencourt et al., 2018). As shown in Fig 5 below, there are four main types of infrastructure in code - scripts, provisioning tools, containers and templating tools, and configuration management tools. Authorized changes can, in turn, be reviewed as pull requests and approval workflows that are enabled by storing infrastructure definitions as code. Any unauthorized changes or mistakes leading to deviation from baselines will show in its change history. With this level of transparency, earlier infrastructure states can then be forensically investigated for compliance audits or troubleshooting security issues. Since IaC attributes changes to individual user accounts, it also prevents repudiation of edits made in case an investigation is needed. Infrastructure-as-code-based versioning and auditing address some of the problems to do with transparency and accountability of changes that normally cause a headache in the process of cloud migration.

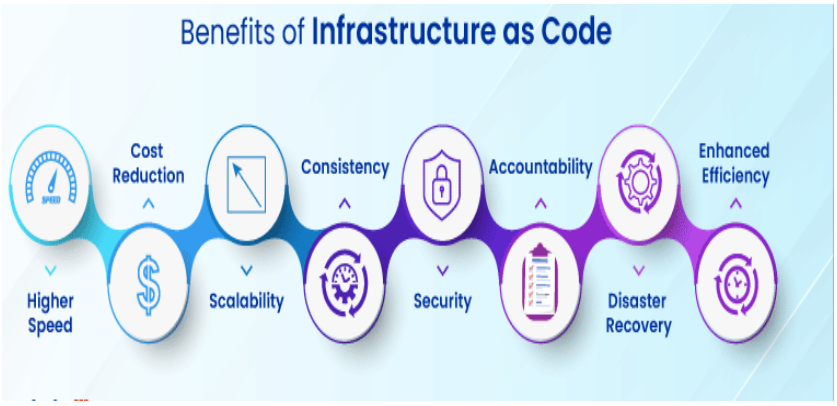

Fig 5. Types of Infrastructure in Code, https://www.veritis.com/blog/exploring-the-benefits-of-infrastructure-as-code-iac-in-it-operations/

Standardization enforced by IaC templates prevents configuration errors and security vulnerabilities from inconsistent manual changes (Garrison et al., 2012). As shown in Fig 5 above, there are four main types of infrastructure in code - scripts, provisioning tools, containers and templating tools, and configuration management tools. When infrastructure provisioning and updates are automated through templated definition files rather than one-off manual edits, systematic compliance with baseline configurations can be ensured. All deployments will conform to the latest guidelines and best practices defined in the template code regarding areas like access controls, patch management, firewall rules, and other security settings. This reduces attack surfaces by eliminating risks of human mistakes or temporary non-conformity. Standard operating procedures for security can be adhered to at scale through programming infrastructure rather than relying on individual administrators.

Enhancing Security through Effective Change Management

Governance of Infrastructure Changes : IaC supports the governance of changes in the infrastructure through review and approval gates for definition changes before deployment (Bittencourt et al., 2018). This consists of reviewing security architecture in workflows to change code so that every change introduced will be properly scrutinized for any new vulnerabilities before the update is cut over into production environments. Automated tests and validations will highlight any code edits straying away from baseline templates in an information security non-compliant manner. This reduces the possibility of occurrence of security incidents by keeping pace with emerging vulnerabilities and incorporates standard procedures for auditing and rolling back provisioning tasks.

Automating the Remediation of Policy Violations: IaC supports automated remediation of configuration settings of infrastructure that break security policies (Gangwar et al., 2015). Any change that may alter the definitions of pre-production environments to deviate from the pre-defined guardrails programmed in IaC will be intercepted and raised for correction in real-time. IaC enables continuous authorization of changes in the infrastructure programmatically with enforcement mechanisms for security controls like separation of duties and the principle of least privilege. This prevents the introduction of vulnerabilities through infected systems or temporary misconfigurations that could otherwise be exploited for breaches.

Streamlined Incident Response: Adopting IaC facilitates faster incident response by enabling tracking of attribute changes over time. With the full history of infrastructure definitions and change activities maintained through code versioning, security anomalies and impacts from incidents can be rapidly analyzed. By simply rolling back compromised environments to previous known-good configurations documented in an earlier code revision, the root cause of the issue and extent of impacts can be efficiently determined, (Yaokumah & Amponsah, 2017). This allows for swift remediation actions to be taken such as patching vulnerabilities, revoking inappropriate access privileges, and restoring services. Automated logging of all infrastructure provisioning tasks through the IaC workflows further simplifies audits and forensic investigations with a clear sequencing of all configuration and coding events. Early detection of vulnerabilities is also facilitated through the continuous monitoring that can be embedded within the IaC deployment pipelines.

Conclusion

In conclusion, incorporating infrastructure as code practices can address numerous security and governance barriers preventing cloud adoption. Standardized change management, control, and compliance enforced at the very level of infrastructure programming make IaC usage associated with public cloud platforms more secure and credible for enterprises. Key risks around lack of transparency, misconfigurations, unauthorized access, and incident response challenges are resolved. When properly integrated with cloud security frameworks, infrastructure as code assuages the concerns that hold back cloud migration in that it promotes accountability and automates adherence to policy and places transparency and auditability into all infrastructure changes. Nevertheless, further research is still solemnly warranted in terms of allowing for customizable guidelines for operationalizing IaC based on the industry or size and any unique needs of the organization in an effort to maximize its security benefits. While opportunities still exist to customize IaC implementations for sector-specific regulatory needs, the available evidence indicates that it is generally effective at enhancing trust for cloud innovations through a strengthened foundation of IT governance and security.

References

1. Alghushami, A. H., Zakaria, N. H., & Aji, Z. M. (2020). Factors influencing cloud computing adoption in higher education institutions of least developed countries: Evidence from Republic of Yemen. Applied Sciences, 10(22), 1-27. https://doi.org/10.3390/app10228098
2. Al-Jabri, I. M., & Alabdulhadi, M. H. (2016). Factors affecting cloud computing adoption: Perspectives of IT professionals. International Journal of Business Information Systems, 23(4), 389-405. https://doi.org/10.1504/IJBIS.2016.080215
3. Alkhater, N., Walters, R., & Wills, G. (2018). An empirical study of factors influencing cloud adoption among private sector organisations. Telematics and Informatics, 35(1), 38-54. https://doi.org/10.1016/j.tele.2017.09.017
4. Alshamaila, Y., Papagiannidis, S., & Li, F. (2013). Cloud computing adoption by SMEs in the north east of England: A multi-perspective framework. Journal of Enterprise Information Management, 26(3), 250-275. https://www.emerald.com/insight/content/doi/10.1108/17410391311325225/full/html
5. Bittencourt, L., Immich, R., Sakellariou, R., Fonseca, N., Madeira, E., Curado, M., Villas, L., DaSilva, L., Lee, C., & Rana, O. (2018). The Internet of Things, Fog and Cloud continuum: Integration and challenges. Internet of Things, 3-4, 134-155. https://doi.org/10.1016/j.iot.2018.09.005
6. Bobie-Ansah, D., & Affram, H. (2024). Impact of secure cloud computing solutions on encouraging small and medium enterprises to participate more actively in e-commerce. International Journal of Research and Technology Innovation. https://www.ijrti.org/viewpaperforall?paper=IJRTI2407064
7. Cegielski, C. G., Jones-Farmer, L. A., Wu, Y., & Hazen, B. T. (2012). Adoption of cloud computing technologies in supply chains. The International Journal of Logistics Management, 23(2), 184-211. https://doi.org/10.1108/09574091211265350
8. Chitra, S., Kwok, R. C. W., Wong, C. C. K., & Cheung, T. C. H. (2015). Education cloud maturity code. In PACIS (p. 114).
9. Durkee, D. (2010). Why cloud computing will never be free. Communications of the ACM, 53(5), 62. https://doi.org/10.1145/1735223.1735242
10. Ewuzie, I., & Usoro, A. (2012). Exploration of cloud computing adoption for e-learning in higher education. In 2012 Second Symposium on Network Cloud Computing and Applications (pp. 151-154). IEEE.
11. Fook, M. C., On, C. K., Rayner, A., Guan, T. T., & Patricia, A. (2018). The determinant factors affecting cloud computing adoption by small and medium enterprises (SMEs) in Sabah, Malaysia. Journal of Telecommunication, Electronic and Computer Engineering, 10(3-2), 83-88.
12. Gangwar, H., Date, H., & Ramaswamy, R. (2015). Understanding determinants of cloud computing adoption using an integrated TAM-TOE model. Journal of Enterprise Information Management, 28(1), 107-130. https://doi.org/10.1108/JEIM-08-2013-0065
13. Garrison, G., Kim, S., & Wakefield, R. L. (2012). Success factors for deploying cloud computing. Communications of the ACM, 55(9), 62. https://doi.org/10.1145/2330667.2330685
14. Gill, S. S., Tuli, S., Xu, M., Singh, I., Singh, K. V., Lindsay, D., Tuli, S., Smirnova, D., Singh, M., Jain, U., Pervaiz, H., Sehgal, B., Kaila, S. S., Misra, S., Aslanpour, M. S., Mehta, H., Stankovski, V., & Garraghan, P. (2019). Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges. Internet of Things, 8, 100118. https://doi.org/10.1016/j.iot.2019.100118
15. Gutierrez, A., Boukrami, E., & Lumsden, R. (2015). Technological, organisational and environmental factors influencing managers' decision to adopt cloud computing in the UK. Journal of Enterprise Information Management, 28(6), 788-807. https://doi.org/10.1108/JEIM-01-2015-0001
16. Hadwer, A. A., Gillis, D., & Rezania, D. (2019). Big data analytics for higher education in the cloud era. In 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA) (pp. 203-207). IEEE. https://doi.org/10.1109/ICBDA.2019.8713257
17. Hassaan, H., Nasir, M. H. M., Khairudin, N., & Adon, I. (2017). Factors influencing cloud computing adoption in small and medium enterprises. Journal of Information and Communication Technology, 16(1), 21-41.
18. Hofstede, G. (2009). Geert Hofstede cultural dimensions.

19. Hsu, C. L., & Lin, J. C. C. (2016). Factors affecting the adoption of cloud services in enterprises. Information Systems and e-Business Management, 14(4), 791-822. https://doi.org/10.1007/s10257-015-0300-9
20. Hwang, M. S., Li, C. T., Shen, J. J., & Chu, Y. P. (2004). Challenges in e-government and security of information. Information & Security, 15(1), 9-20.
21. Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report.
22. Lee, S. G., Hwang, S., Kang, J., & Yoon, S. (2014). Factors influencing the adoption of enterprise cloud computing. Journal of Internet Technology, 15(1), 65-75. https://doi.org/10.6138/JIT.2014.15.1.07
23. Lian, J. W., Yen, D. C., & Wang, Y. T. (2014). An exploratory study to understand the critical factors affecting the decision to adopt cloud computing in Taiwan hospital. International Journal of Information Management, 34(1), 28-36. https://doi.org/10.1016/j.ijinfomgt.2013.09.004
24. Liang, Y., & Qi, G. (2017). The determinants of e-government cloud adoption: Multi-case analysis of China. International Journal of Networking and Virtual Organisations, 17(2-3), 184-201. https://doi.org/10.1504/IJNVO.2017.085535
25. Low, C., Chen, Y., & Wu, M. (2011). Understanding the determinants of cloud computing adoption. Industrial Management & Data Systems, 111(7), 1006-1023. https://www.emerald.com/insight/content/doi/10.1108/02635571111161262/full/html
26. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg.
27. Odeh, M., Garcia-Perez, A., & Warwick, K. (2017). Cloud computing adoption at higher education institutions in developing countries: A qualitative investigation of main enablers and barriers. International Journal of Information and Education Technology, 7(12), 921-927.
28. Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. Information & Management, 51(5), 497-510. https://doi.org/10.1016/j.im.2014.03.006
29. Pathan, Z. H., Jianqiu, Z., Akram, U., Latif, Z., Khan, M. K., Tunio, M. Z., & Pathana, Z. H. (2017). Essential factors in cloud-computing adoption by SMEs. Human Systems Management, 36(4), 261-275. https://doi.org/10.3233/HSM-17133
30. Priyadarshinee, P., Raut, R. D., Jha, M. K., & Gardas, B. B. (2017). Understanding and predicting the determinants of cloud computing adoption: A two staged hybrid SEM-neural networks approach. Computers in Human Behavior, 76, 341-362. https://doi.org/10.1016/j.chb.2017.07.027
31. Raut, R. D., Priyadarshinee, P., Gardas, B. B., & Jha, M. K. (2018). Analyzing the factors influencing cloud computing adoption using three stage hybrid SEM-ANN-ISM (SEANIS) approach. Technological Forecasting and Social Change, 134, 98-123. https://doi.org/10.1016/j.techfore.2018.05.020
32. Rogers, E. M. (2010). Diffusion of innovations (4th ed.).
33. Safari, F., Safari, N., Hasanzadeh, A., & Ghatari, A. R. (2015). Factors affecting the adoption of cloud computing in small and medium enterprises. International Journal of Business Information Systems, 20(1), 116-137. https://doi.org/10.1504/IJBIS.2015.070894
34. Senarathna, I., Wilkin, C., Warren, M., Yeoh, W., & Salzman, S. (2018). Factors that influence adoption of cloud computing: An empirical study of Australian SMEs. Australasian Journal of Information Systems, 22(0). https://doi.org/10.1016/0921-5093(90)90273-6
35. Senyo, P. K., Addae, E., & Boateng, R. (2016). Cloud computing research: A review of research themes, frameworks, methods and future research directions. International Journal of Information Management, 38(1), 128-139. https://www.sciencedirect.com/science/article/pii/S0268401217308526
36. Singh, J., & Mansotra, V. (2019). Factors affecting cloud computing adoption in the Indian school education system. Education and Information Technologies. https://doi.org/10.1007/s10639-019-09878-3
37. Tornatzky, L. G., Fleischer, M., & Chakrabarti, A. K. (1990). The processes of technological innovation. Lexington Books.

38. United Nations Conference on Trade and Development (UNCATD). (2017). Information economy report 2017. https://unctad.org/en/PublicationsLibrary/ier2017_en.pdf
39. Wahsh, M. A., & Dhillon, J. S. (2015). An investigation of factors affecting the adoption of cloud computing for e-government implementation. In 2015 IEEE Student Conference on Research and Development (SCOReD) (pp. 323-328). IEEE. https://doi.org/10.1109/SCORED.2015.7449349
40. Walterbusch, M., Martens, B., & Teuteberg, F. (2013). Evaluating cloud computing services from a total cost of ownership perspective. Management Research Review, 36(6), 613-638. https://doi.org/10.1108/01409171311325769
41. Yaokumah, W., & Amponsah, R. A. (2017). Examining the contributing factors for cloud computing adoption in a developing country. International Journal of Enterprise Information Systems, 13(1), 17-37. https://doi.org/10.4018/IJEIS.2017010102
42. Yigitbasioglu, O. M. (2015). The role of institutional pressures and top management support in the intention to adopt cloud computing solutions. Journal of Enterprise Information Management, 28(4), 579-594. https://www.emerald.com/insight/content/doi/10.1108/JEIM-09-2014-0087/full/html
43. Zhao, F., Scheruhn, H. J., & von Rosing, M. (2014). The impact of culture differences on cloud computing adoption. In International Conference on Human-Computer Interaction (pp. 776-785). Springer.