2024 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
MODELING, SIMULATION, PROTOTYPING & VALIDATION (MSPV) TECHNICAL SESSION
AUGUST 13-15, 2024 - NOVI, MICHIGAN

How the Cloud is a Mission Enabler for Embedded System Development

John Brabbs[1], B. Colby Jones, PhD[2]

[1]US Army DEVCOM GVSC, Warren, MI
[2]SAIC, Huntsville, AL

ABSTRACT

The Army can increase its software modernization effort for Embedded System software development by leveraging the Cloud to expand the capability of the DevSecOps environment to include automated testing at scale. The Cloud will support the integration of current and new off-the-shelf technologies; and merging next generation technologies from industry partners into a coherent DevSecOps Cloud ecosystem. The following areas are critical to meeting mission requirements and applications: virtual simulation, trade study analytics, technology adoption, DevSecOps capabilities, artificial intelligence applications and infrastructure, and collaborative single vehicle Systems Integration Laboratory (SIL). These areas are all essential to shortening the vehicle product lifecycle and time to deliver mission essential capabilities to the field to support warfighter needs.

Citation: J. Brabbs, B. Jones, "How the Cloud is a Mission Enabler for Embedded System Development," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 13-15, 2024.

1. INTRODUCTION

Ground Vehicle Systems Center's (GVSC) Software Engineering Center (SEC) has put together a vision for software modernization to support the Department of Defense (DoD) Software Modernization strategy and the National Defense Strategy where the DoD will "prioritize speed of delivery, continuous adaptation, and frequent modular upgrades" [5]. GVSC's SEC supports multiple embedded system projects and needs the ability to "enable the delivery of resilient software capability at the speed of relevance." [6]. SEC sees the Cloud as a critical enabler to our software modernization strategy around embedded systems. The Cloud allows SEC to scale the automated test vision (Figure 1) and supports the embedded system continuous integration (CI) pipeline (Figure 2). GVSC sees a Hybrid Cloud approach as the solution since it will allow data/applications to be shared seamlessly and allow SEC to selectively deploy/scale/move resources in the ways that make the most

---

DISTRIBUTION STATEMENT A. Approved for public release; distribution unlimited. OPSEC # 8660 sense for the software modernization mission at any given time.

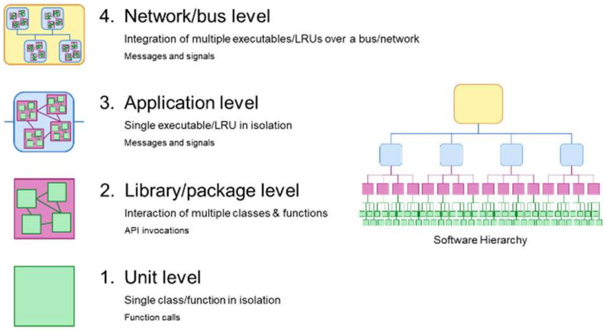

Figure 1: Automated Testing [10]

2. Embedded System Development Use Case

The Army's ground vehicle embedded systems path to production differs from a web or mobile app path since the target hardware is a vehicle platform. This unique path to production provides additional challenges with embracing development, security, and operations (DevSecOps) and using the Continuous Integration pipeline. Figure 2 shows an example of an embedded systems continuous integration pipeline.

The path to production for embedded systems includes a system integration lab where the software can be tested on hardware in a lab environment before testing on the vehicle. Embedded systems can have non-traditional hardware and do not always run on general-purpose operating systems (OS) like Microsoft Windows and Linux; instead, they may use Real-Time Operating Systems (RTOS). Even with the differences in embedded systems' path to production, implementing DevSecOps CI pipelines in the Cloud still has a lot of value. A key piece is to create simulations and emulations to virtualize the hardware for automated testing in the Cloud.

3. Current Software Development Environment

SEC uses the GVSC Development Environment (GDEV), which is hosted on-premises and includes Jira, Confluence, Bitbucket, SonarQube, Jenkins, and Artifactory. A few SEC projects also use the US Army's Enterprise Cloud Management Agency's DevSecOps (DSO) Services (formerly CReATE) environment hosted in cloud Army (cArmy). DSO-Services includes Gitlab, Jira, Confluence, Nexus Repository Manager, and SonarQube. GDEV comprises a few computing resources for pipeline runners that can be used to support continuous integration in both GDEV and DSO-Services.

3.1. Current Methods

The current method tends to spend more time manually testing after development. Unit testing has and is being implemented in pipelines for some projects. Automated testing at the user interface level has also been developed/implemented for some projects.

3.2. Challenges

SEC would like to implement CI pipelines, as shown in Figure 2, that can integrate and scale the different levels of automated testing using 100s to 1000s of tests with scalable computing. The current challenge is that on-premises compute resources are either dedicated to a system integration lab (SIL) or limited to a few virtualized servers, so there is no ability to scale the testing.

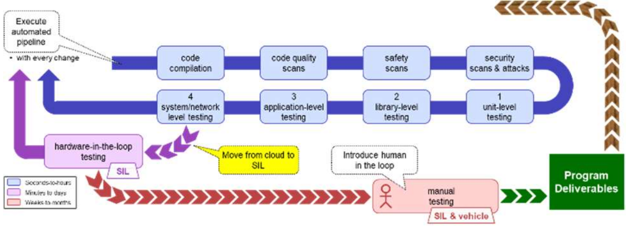

Figure 2: Embedded System CI pipeline [10]

4. Testing Environment

The current environment for testing software in the SIL uses virtual benches with simulations running on local computers; hardware benches with the networks, subsystems, line replaceable units (LRU) and other vehicle components; and the actual vehicle/system.

4.1. Current Methods

SEC is implementing automated testing across multiple projects, including the numerous testing levels shown in Figure 1. The current efforts support these initiatives using on-premises hardware in the SILs. Some initiatives have started implementing CI pipelines to integrate the testing process into the SILs.

4.2. Challenges

SEC would like to implement automated testing that can scale using 100s to 1000s of tests with scalable computing. The current challenge is that on-premises compute resources are either dedicated to a SIL or limited to a few virtualized servers, so there is no ability to scale the testing.

4.3. CASTLE Cloud Migration

The main objective of the Continuous Autonomy Simulation Test Laboratory Environment (CASTLE) migration to the cloud development project is to enhance the current setup in the SIL, shown in Figure 3, with a cloud-based approach to containerize all applications of the simulation and autonomy software. CASTLE supports autonomy software in the loop testing. The computer rack in the SIL consists of four computers where each computer represents a vehicle (the lead vehicle and three following vehicles). The initial approach was to perform a lift and shift migration by completing a Physical to Virtual (P2V) conversion of the physical configuration of CASTLE in the SIL to the virtual configuration.

Figure 3: CASTLE on-premises compute.

Migrating the CASTLE architecture shown in Figure 4 to the Cloud is to achieve compute scalability while allowing users to develop and run simulations remotely instead of physically going to the laboratory to access the hardware setup to create and run simulations. This approach will use the cArmy ecosystem. The next step is to obtain the Authority to Operate (ATO) in cArmy, upon which all other project milestones depend. This environment will be built in parallel with ATO development. However, once ATO is approved, all the pieces will be combined to prove the process within cArmy.

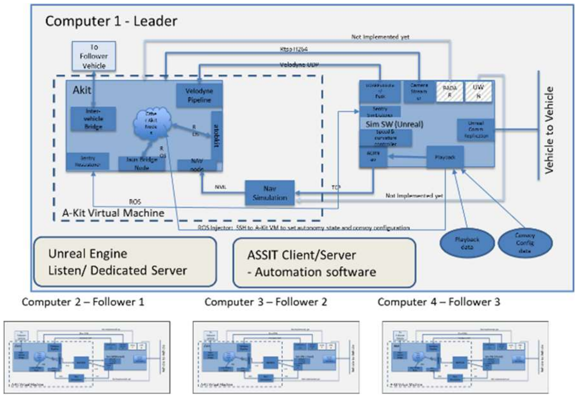

Figure 4: CASTLE Architecture

There is a list of lessons learned from the initial CASTLE development that have prompted a redesign of the architecture, now called GDEV-Cloud (GDEV-C).

GDEV-C will use Robotic Technical Kernel (RTK) as the autonomy software under test, with ProjectGL as the simulation software. ProjectGL from GVSC's Immersive Simulation team uses Unreal Engine 4 (UE4) as the simulation software for autonomous programs. The GDEV-C project will replace the on-premises hardware with cloud-based computing. In the future, GDEV-C will also port the SPDS Infrastructure as Code (IaC) and Configuration as Code (CaC) architecture, developed in Huntsville for Aviation Missile Center (AvMC), to deploy as the pipeline in this environment.

The original intent to "lift and shift" was not feasible due to being cumbersome and very expensive due to GPU utilization and the way the physical machines are configured; the new architecture will containerize all the application components using Project GL, which is the UE4, representing the physical vehicle and RTK is controlling it. The RTK application, the "brains" of the system being tested, interacts with UE4 to run a simulation based on the terrain defined as part of the simulation. The simulation will be run in the Cloud in a non-rendered, "headless" state where the simulation is not visualized. The simulation will collect the data of the messages communicated between RTK and UE4.

The minimum viable product (MVP) for the project will have a working simulation of the lead vehicle. This step is followed by adding one following vehicle as the next step. Each server is configured similarly, with one designated as a leader and the others as followers. GDEV-C will also adapt to use Azure for containerization to further the scalability.

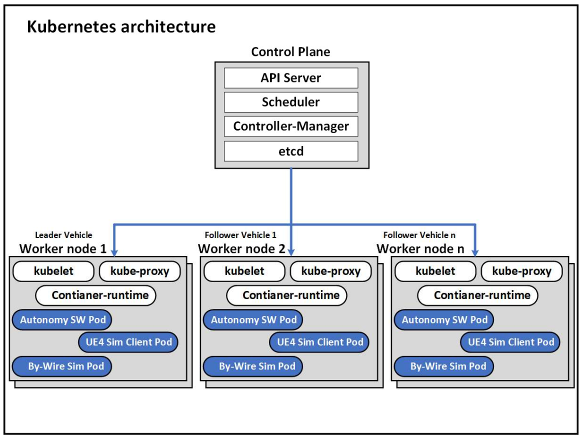

Figure 5: Notional Future CASTLE Architecture

The plan is to codify this environment and to spin out 1-N followers based on CaC and IaC. A control file will have the simulation parameters, which kick off the environment build, execute the simulation, collect the data, and tear everything back down; the only time it instantiates is when a simulation runs. All data is collected in a central repository for data analytics for reporting and creating dashboards.

5. Enter the Cloud

Cloud computing promises reduced capital expenditures, on-demand computing resources, and increased organizational agility. Entering the Cloud allows organizations to have automated, self-provisioning software development platforms and the requisite supporting infrastructure to enable rapid, incremental software development [11]. This paradigm shift in the availability and automated provisioning of computing resources requires effective governance, enhanced cybersecurity oversight, and cloud resource utilization management.

The reality is that introducing Cloud capabilities into an organization requires a disciplined, incremental approach to successfully modernize and migrate existing on-premises capabilities to take advantage of the Cloud. Many on-premises capabilities are siloed with unsuitable application architectures for fully leveraging Cloud capabilities. These on-premises capabilities can either be migrated to the Cloud using a lift-and-shift approach with slight modification, resulting in higher recurring Cloud costs, or rearchitected to optimize Cloud utilization, resulting in higher migration costs. Undertaking an incremental approach to Cloud migration reduces risk and divides effort into manageable segments. The incremental approach focuses on long-term return on investment (ROI) and the tangible and intangible factors of cloud migration supported by the latest technical capabilities.

5.1. Organizational Agility: Compute Resources with Flexibility

Cloud platforms allow organizations to realize highly customizable and flexible computing resources. The critical cloud agility characteristics are listed below:

1. The appearance of infinite on-demand computing resources, mitigating the need to develop extensive Cloud Compute resource procurement plans as required for on-premises computing capabilities [4].
2. Removal of the need for Cloud user up-front commitments, allowing funding-constrained organizations to begin small and grow hardware resources as needs arise [4].
3. The ability to cycle computing capabilities for short-term needs and deprecate them when no longer needed using infrastructure as code (IaC) and configuration as code (CaC). The ability to codify infrastructure and configuration allows organizations to create environment "templates" for rapid deployment into new Cloud enclaves as a replica of previous environments. The concept of templated on-demand test labs rapidly deployed in minutes, persisting throughout the test cycle, and then being torn down enables funding flexibility by reducing test lab investments and sustainment costs [4].

Other benefits are related to enabling human subject matter expert (SME) resources to be more fluid in supporting mission objectives. In this sense, fluidity is defined as the ability of SMEs to rapidly transition between key activities and launch templated resources in support of mission objectives [4]. Imagine an essential SME being moved between projects with no downtime even though the new tasking requires a radically different technology stack from the previous assignment. Introducing SME fluidity within the organization promotes organizational agility by ensuring seamless SME transition between assignments.

5.2. Automation, Automation, Automation

Building out computing infrastructure has traditionally been a complex process that evolves as enhanced capabilities encompassing scalability, fault tolerance, redundancy, and maintainability are integrated into the environment. Historically, this infrastructure build-out has been a manual, labor-intensive process that decreases organizational agility.

The advent of Infrastructure as Code (IaC) and Configuration as Code (CaC) shifted the infrastructure build paradigm to treat environmental components as software constructs for placement under configuration management and reuse across the Cloud enterprise [7]. The definition of IaC is an infrastructure automation approach leveraging development, security, and operations (DevSecOps) best practices for continuous integration and continuous delivery (CI/CD) of infrastructure software components. The following principles describe an IaC solution:

Configuration version control is leveraged to manage and maintain IaC releases, where each build is stored as a versioned artifact. Changes are tracked using source code repository version-control commits to provide change traceability and rollback capability [7].

Predictability is an IaC principle that ensures consistent deployment of the same environmental and associated attributes from each release point in the version control system.

Consistency guarantees that each time the IaC/CaC baseline is deployed, it generates a mirrored environment. This principle allows complex infrastructure entities to be rapidly deployed, avoiding configuration drift and environmental inconsistencies.

Repeatability ensures the outcome of IaC/CaC deployment is always the same based on the known inputs.

Composability is defined by abstraction and modular approach to building complex application systems. This capability allows end users to concentrate on their needs rather than provisioning infrastructure components [7].

Key IaC Concepts:

- Each resource or component in the environment, including servers, user accounts, configuration, and utilities, is declared as code.
- IaC tool source code is referenced using specific terminologies such as recipes, manifests, cookbooks, and templates.
- IaC presents repeatable capabilities, translating into increased reliability and significant operational efficiencies.
- IaC introduces environment-build consistency for managing promotion to production landscapes comprised of development, test, and production environments. By leveraging the same IaC codebase, configuration drift is mitigated due to the standard build process from a static IaC baseline.

5.3. Cloud Value Proposition

The Cloud enables scalable Continuous Integration/Continuous Deployment (CI/CD) capabilities, offering shorter software development cycles and improved software quality while optimizing resource utilization. Continuous integration uses a shared repository that facilitates continuous integration of software changes from any number of developers [3]. On the other hand, Continuous delivery is defined as the automation of integrated code deployment into the promote-to-production landscape (Dev, Test, Production) [3]. Shorter development cycles use automation to deliver software modifications to downstream systems in smaller releases using CI/CD pipelines. CI/CD pipelines are a grouping of linked tools to perform automated tasks (build, test, cybersecurity checks, etc.) required to validate the software. While these tool pipelines traditionally validate and deliver software releases to the promote-to-production for web applications, they can also support embedded software.

CI/CD for embedded systems shifts the traditional promote-to-production concept of automated software validation and deployment capabilities. This concept shift is due to the embedded system "production" environment residing on a vehicle platform rather than in a Cloud environment. The challenge is compounded by the fact that embedded systems can have unique hardware and run on real-time operating systems (RTOS). The concept of embedded system CI/CD must consider leveraging Cloud resources to validate software in an environment mirroring the vehicle production environment continuously. Mitigating these challenges requires using Virtual Software Integration Labs (VSILs) as part of the CI/CD pipeline and edge connections to the physical software integration labs. Understanding the differences between CI and CD for embedded systems is essential to determine automation, risk, and release strategy requirements.

Critical Cloud Value Principles

Automation: Scalable CI/CD is valuable because it automates software build, validation, and release [9]. CI/CD for embedded systems requires adding production platform emulation capabilities for release validation. The use of IaC in the Cloud allows for the launch of multiple platform emulators on demand for automated build, testing, and verification of each embedded software release. This capability reduces test cycle timelines and provides shorter feedback loops to accelerate software development activities [3].

Consistency: Using IaC to build CI/CD pipelines and VSILs automatically ensures that each validation environment is identical across development and test environments. Identical environments ensure that observed defects or anomalies are reproducible and consistent for troubleshooting while minimizing environment-related issues.

Fast feedback loops / Scalability: Leveraging automation and consistency in CI/CD Cloud pipelines accelerates developer feedback [3]. Imagine launching a CI/CD pipeline that scales to automatically build hundreds or thousands of emulated platforms and then execute hundreds or thousands of test cases simultaneously. A test cycle that traditionally took days or weeks can be completed in hours, significantly reducing the developer feedback loop while improving software quality.

Cost Optimization: Cloud resources are available on demand and torn down when not needed. This optimizing capability is possible using IaC/CaC to automatically spin up computing infrastructure and resources to support test cycles rather than purchasing on-premises computing, network, and storage capacity.

Organizational Agility: Skilled developers are a valuable and costly resource. Transitioning developers between projects with diverse development platforms and environments is challenging when an individual uses a GFE laptop for development activities. Leveraging preconfigured Cloud-based virtual desktops for development increases organizational agility by freeing valuable development resources from reconfiguring their GFE laptops to match each project's confirmation requirements [3].

6. Cloud Challenges

As discussed in the previous sections, cloud computing has definitive advantages. With the maturation of Cloud services leading to increased adoption across the DoD, organizations are grappling with the shift to new operational models. These new models leverage the flexibility of on-demand computational resources but must also consider new challenges.

6.1. Cost Management

The allure of on-demand computing resources and a "pay as you go" model is increasingly attractive to DoD organizations. Organizations must manage the benefit of lowering the costs associated with compute and store resources to avoid unexpected costs [8]. This cost management need has given rise to the concept of FinOps. FinOps contains three pillars: Cloud Financial Management (CFM) (budgeting, burn rates, dashboards), Capacity Operations (CapOps) (resource utilization monitoring, anomaly detection, resource optimization), and DevSecOps (guardrails, CI/CD, continuous ATO). CFM ensures the tracking of computing resource burn rates against a budget to safeguard against utilization exceeding available funding. CapOps assists in avoiding hidden fees associated with underutilized resources and issues alerts when there are unexpected usage spikes (anomalies) with integrated reporting dashboards to provide a holistic view of the Cloud environment. Finally, DevSecOps is a set of automation principles that serve as the operational mechanism for the cloud environment.

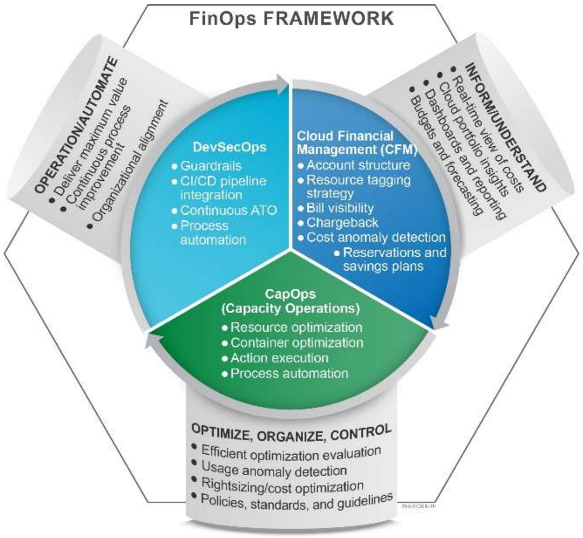

Figure 6: FinOps Framework

6.2. Performance Challenges

Performance is an additional dimension that needs consideration during cloud adoption. Organizations should understand Usage models well, emphasizing Cloud ingress and egress. The amount and frequency of data movement into and out of a Cloud environment can significantly impact costs. When considering modeling and simulation use cases, it is important to evaluate simulation rendering options (pixel streaming, headless, etc.) and validate that network bandwidth is sufficient to support organizational needs. Other performance-related areas include fault tolerance, disaster recovery, and load balancing [8].

6.3. Cybersecurity

Effective cybersecurity is a principal consideration when using a Cloud environment. Modern Cloud environments must have a continuous monitoring capability for intrusion detection, malware threats, user activity monitoring, and change management [8]. An effective continuous monitoring capability may include multiple tools for tracking specific cybersecurity threats to ensure environment availability, integrity, and confidentiality. Additionally, including a change management monitoring component assists in maintaining an ATO by logging architectural or configuration changes over time, which helps move towards a continuous ATO.

6.4. Interoperability and Flexibility

Each Cloud Service Provider (CSP) uses different terminology for Cloud-native services that require modifications to IaC/CaC when transitioning between Cloud platforms. There are methods for abstracting these unique platform attributes out of IaC/CaC that require forethought and planning to execute successfully. Organizations must ensure maximum interoperability and flexibility when investing significant resources into a singular CSP [8].

6.5. Lack of Knowledge and Expertise

The shift to a Cloud operational model requires subject matter experts to build, administer, and maintain the environment effectively. Support personnel must continually develop and upskill their abilities to keep up with the ever-changing Cloud technical landscape [1]. This shift in operational mindset requires a motivated, continual learning workforce that actively researches, recognizes, manages, and advances Cloud capabilities for the organization.

7. Hybrid Cloud Approach

Organizations with significant investments in non-cloud computing and store resources can approach Cloud adoption on a more incremental basis using a hybrid Cloud architecture. The hybrid Cloud approach seeks to leverage the investment in on-premises compute and store resources while also tapping into the scalable, on-demand capabilities of the Cloud [2]. The interconnection of traditional computing and storage resources with the Cloud provides organizational agility by allowing organizations to incrementally add or migrate capabilities to the Cloud [1]. This architectural approach will enable organizations to utilize existing technology investments while avoiding significant capital expenditures in developing new or expanding existing capabilities.

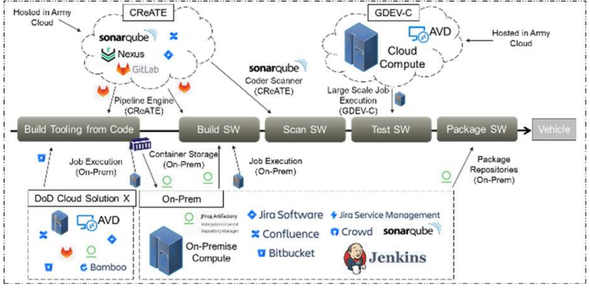

Figure 7: Notional Hybrid Approach

The hybrid Cloud approach enables organizations to securely access and manage compute resources seamlessly in a heterogeneous environment [2]. Organizations can incrementally migrate capabilities into the Cloud environment as on-premises infrastructure reaches the end of life. Additionally, when unexpected requirements arise, the hybrid Cloud approach provides the necessary scalability to add resources quickly without significant capital expenditure.

8. Software Development for Embedded Systems

Developing software for Embedded Systems can include non-traditional hardware and Real-Time Operating Systems (RTOS). A key piece in supporting embedded system software development is creating simulations and emulations to virtualize the hardware for software testing, which allows for automated testing instead of manual testing on real hardware.

8.1. Advantages over Current Methods

As mentioned above, the advantage of SEC using the Cloud would be the ability to scale the automated testing with the flexibility of running 100s to 1000s of tests in parallel if needed. The Cloud also allows multiple developers to have simulations not locked down to one piece of hardware in the lab and can be accessible at any time. This approach expands the ability to add additional software tools, simulations, emulations, and testing capabilities without investing in on-premises hardware that might sit idle for long periods until needed.

8.2. Challenges

Cloud technology is available today in the Department of Defense (DoD) that can support SEC's software modernization vision. Some challenges are DoD and Army policies, changing the culture, transitioning on-premises architectures to support a cloud architecture, implementing the solution, and resources/funding.

9. Conclusion

GVSC's SEC is moving forward to utilize the Cloud as a critical enabler for its software modernization vision. SEC sees the Cloud as a key piece of supporting embedded system projects as future ground systems become more Software-intensive systems. A Hybrid Cloud approach will also be key to achieving this goal. GVSC data/applications can be shared seamlessly between on-premises and the cloud, and SEC can selectively deploy/scale/move resources in the ways that make the most sense for the software modernization mission at any given time.

10. REFERENCES

[1] A. Al Hadwer, M. Tavana, D. Gillis, and D. Rezania, “A Systematic Review of Organizational Factors Impacting Cloud-based Technology Adoption Using Technology-Organization-Environment Framework,” Internet of Things, vol. 15, p. 100407, Sep. 2021, doi: https://doi.org/10.1016/j.iot.2021.100407.

[2] Atieh, A. T. (2021) ‘The Next Generation Cloud technologies: A Review OnDistributed Cloud, Fog And Edge Computing and Their Opportunities and Challenges’, ResearchBerg Reviewof Science and Technology, 1(1), pp. 1–15.

[3] B. Godavarthi, N. Narisetty, K. Gudikandhula, R. Muthukumaran, D. Kapila, and J. V. N. Ramesh, “Cloud computing enabled business model innovation,” The Journal of High Technology Management Research, vol. 34, no. 2, p. 100469, Nov. 2023, doi: https://doi.org/10.1016/j.hitech.2023.100469.

[4] E. Ramos, A. S. Patrucco, and M. Chavez, “Dynamic capabilities in the ‘new normal’: a study of organizational flexibility, integration and agility in the Peruvian coffee supply chain,” Supply Chain Management: An International Journal, vol. ahead-of-print, no. ahead-of-print, Aug. 2021, doi: https://doi.org/10.1108/scm-12-2020-0620.

[5] J. Mattis, “Summary of the 2018 National Defense Strategy of the United States of America,” Department of Defense, 2018. Available: https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf

[6] Kathleen Anne Holland Hicks, ‘‘Department of Defense Software Modernization Strategy,’’ Office Secretary Defense, Washington, DC, USA, 2022

[7] S. Achar, “Enterprise SaaS Workloads on New-Generation Infrastructure-as-Code (IaC) on Multi-Cloud Platforms”, Glob. discl. econ. bus., vol. 10, no. 2, pp. 55–74, Jul. 2021, doi: 10.18034/gdeb.v10i2.652.

[8] U. M. Kondi, P. Yaddanapudi, K. Burugupalli, P. Vamsi, D. M. Shameem, and D. M. Kavitha, “A Review of the Challenges and Opportunities in Cloud Computing,” in Proceedings of the International Conference on Innovative Computing & Communication (ICICC, Mar. 2022.

[9] Vamsi Krishna Thatikonda, “Beyond the Buzz: A Journey Through CI/CD Principles and Best Practices,” European Journal of Theoretical and Applied Sciences, vol. 1, no. 5, pp. 334–340, Sep. 2023, doi: https://doi.org/10.59324/ejtas.2023.1(5).2

[10] Wilkinson, Robert, “Software Modernization and DevSecOps”, Detroit Arsenal Acquisition Insight Days, AcqDays Topic 2-Cybersecuirty and DEVSECOPS, Nov. 2023, dau: https://www.dau.edu/events/acquisition-insight-days .

[11] Z. Mahmood and S. Saeed, Software Engineering Frameworks for the Cloud Computing Paradigm. London: Springer London, 2013.