ISSN No:-2456-2165

Exploring the Design and Development of Cloud-Native Applications

(Area of Focus: Cloud-Native Applications)

Richard Karegeya[1]; Dr. Wilson Musoni[2] (PhD)
Masters of Science with honors in Information Technology, at the University of Kigali, Rwanda

Abstract:- Cloud-native application development is a paradigm that embraces the principles of scalability, elasticity, resilience, and agility to harness the full potential of cloud platforms. Cloud-native applications are designed to take advantage of the cloud computing model, which offers a number of benefits, such as scalability, elasticity, and availability. This thesis investigates the design and development of cloud-native applications, with a focus on the following topics: The principles of cloud-native design, the use of microservices in cloud-native applications, the development of cloud-native applications using containerization, and The deployment. The thesis then presents a case study of the development of a cloud-native application. Cloud-native applications offer a number of benefits over traditional monolithic applications. They are more scalable, adaptable, and evolvable. They are also easier to deploy and manage. Data will be gathered using a physical survey which will target different store software developers, cloud architects, and IT managers of the Ministry of Justice through questionnaires. Overall, this thesis aims to contribute to the growing of knowledge on cloud-native application development. Here are some of the key points from the abstract: Cloud-native applications are designed to be scalable, elastic, resilient, and agile. They are made up of small, independent services, which makes them easier to deploy and manage. Cloud-native applications can be developed using containerization, which makes them portable and easy to deploy across different cloud platforms. The thesis will investigate the design and development of cloud-native applications, as well as the benefits they offer over traditional monolithic applications. The thesis will also present a case study of the development of a cloud-native application.

I. INTRODUCTION

Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider (or CSP).

The analysis has shown that cloud computing is replacing traditional outsourcing and premise-based data centers for software applications and service delivery.

With the rise of cloud computing, there has been a paradigm shift in how applications are designed and developed. Cloud-native applications are applications that are designed specifically for cloud environments, taking advantage of cloud technologies to improve scalability, resilience, and agility.

Cloud native applications are software applications that are designed and built specifically to run on cloud infrastructure, such as public, private, or hybrid cloud environments. These applications are developed using a set of principles and practices that prioritize scalability, resilience, and flexibility.

- *This Study will Focus on Containers, Micro-Services, and Serverless Architectures as a Cloud Native Applications.*

- *Containers:* Containers are a lightweight way to package and deploy applications, along with their dependencies and configurations, in an isolated environment. Containers use a shared operating system kernel, but each container has its own file system, network, and resources. Containers enable faster deployment, portability, scalability, and consistency across different environments.
- *Microservices* are an architectural style that structures an application as a collection of small, loosely coupled, and independently deployable services. Each service represents a specific business capability and can be developed, deployed, and scaled independently of other services. Micro services communicate with each other through well-defined APIs, typically over lightweight protocols such as HTTP or messaging queues. This approach enables teams to work on different services concurrently, promotes scalability, and improves fault tolerance. micro services architecture facilitates agile development, continuous deployment, and allows organizations to evolve their applications more rapidly.
- *Serverless Architecture*, also known as Function-as-a-Service (FaaS), is a cloud computing model where developers focus solely on writing and deploying individual functions or code snippets, without managing the underlying infrastructure. In a serverless setup, developers write functions that are triggered by specific events or requests, such as HTTP requests, database changes, or timers. The cloud provider takes care of automatically scaling the infrastructure to execute the functions, and developers are billed based on the actual

usage of resources. Serverless architectures promote a pay-as-you-go model, improve resource efficiency, and abstract away infrastructure management tasks, allowing developers to focus on writing business logic rather than managing servers.

➢ *Background of Cloud-Native Applications (CNA)*

Cloud native is a term that is invoked often but seldom defined beyond saying ``we built it in the cloud'' as opposed to "on-prem". Although there is no single set of rules for designing cloud applications, there is a growing consensus on some key ideas and informal design patterns that have been used successfully in many applications.

Cloud native applications have emerged as a result of the growing trend towards cloud computing, which began in the early 2000s. This trend was fueled by the need for businesses to leverage the advantages of cloud computing, such as scalability, flexibility, and cost efficiency. Cloud computing allowed businesses to move away from traditional on-premises IT infrastructure, which was costly and inflexible.

The term "cloud native" was first coined in 2010 by Adrian Cockcroft, then a cloud architect at Netflix. It was used to describe applications that were designed specifically to run in the cloud, taking advantage of the cloud's unique features and capabilities. These applications were built using cloud-native technologies and architectures, which were optimized for the cloud environment.

There are various cloud native technologies which are Containers, microservices, and serverless The emergence of cloud-native technologies, such as containers and microservices, has played a significant role in the development of cloud-native applications. Containers allow developers to package their applications into self-contained units, which can be easily deployed and scaled in the cloud. Microservices, on the other hand, break down applications into smaller, modular components, which can be developed and deployed independently.

In this introduction, we will describe these cloud native concepts and illustrate them with examples. We will explore the technical trends that are shaping the future of cloud applications. We will start by discussing the basic properties that many cloud native applications have in common. Once we have identified the basic properties of cloud native applications, we can then describe how these properties are achieved through the use of technical design patterns.

II. METHODOLOGY

A. *Data Collection Methods and Instruments/ Tools*

In this study, the researcher engaged in data collection through specified procedures, employing a mixed-method approach encompassing both qualitative and quantitative methodologies, while examining secondary data sources.

B. *Data Analysis*

The process of discovering solutions through investigation and interpretation is known as data analysis. Understanding survey and administrative source results and presenting data information require data analysis. Data analysis is anticipated to provide light on the subject of the research and respondent's perceptions as well as to increase reader's understanding of the subject and pique their interest in this position of the research

C. *Research Design*

This research design will explore the creation and design of cloud-native applications. The research will focus on the key principles of cloud-native design, microservices and containerization, benefits of cloud-native applications, challenges of developing and deploying cloud-native applications, as well as the best practices for cloud-native application development.

➢ *Process of the Proposed CNA Flowchart Algorithm*

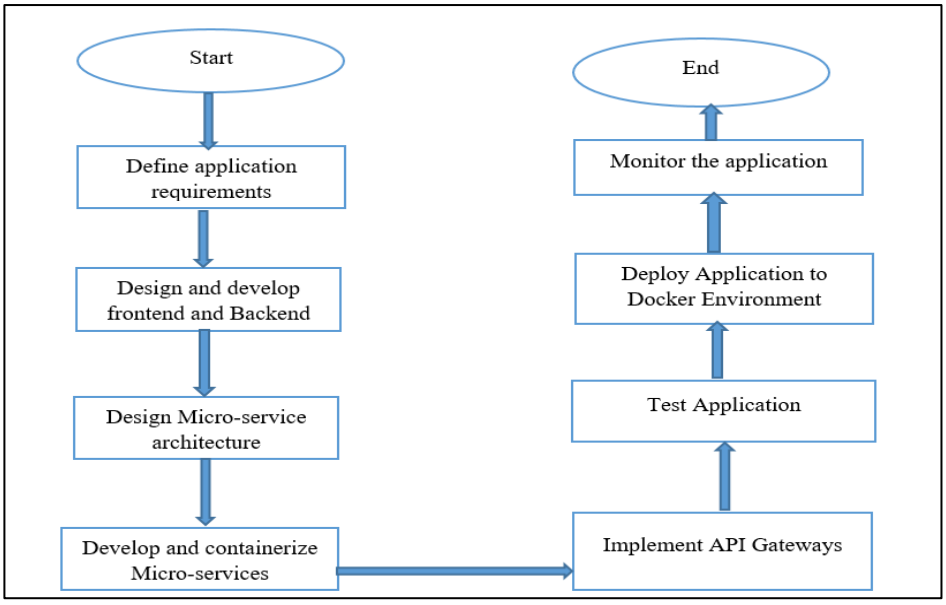

Fig 1 Cloud-Native Application Flowchart Algorithm

➢ *Conceptual Framework*

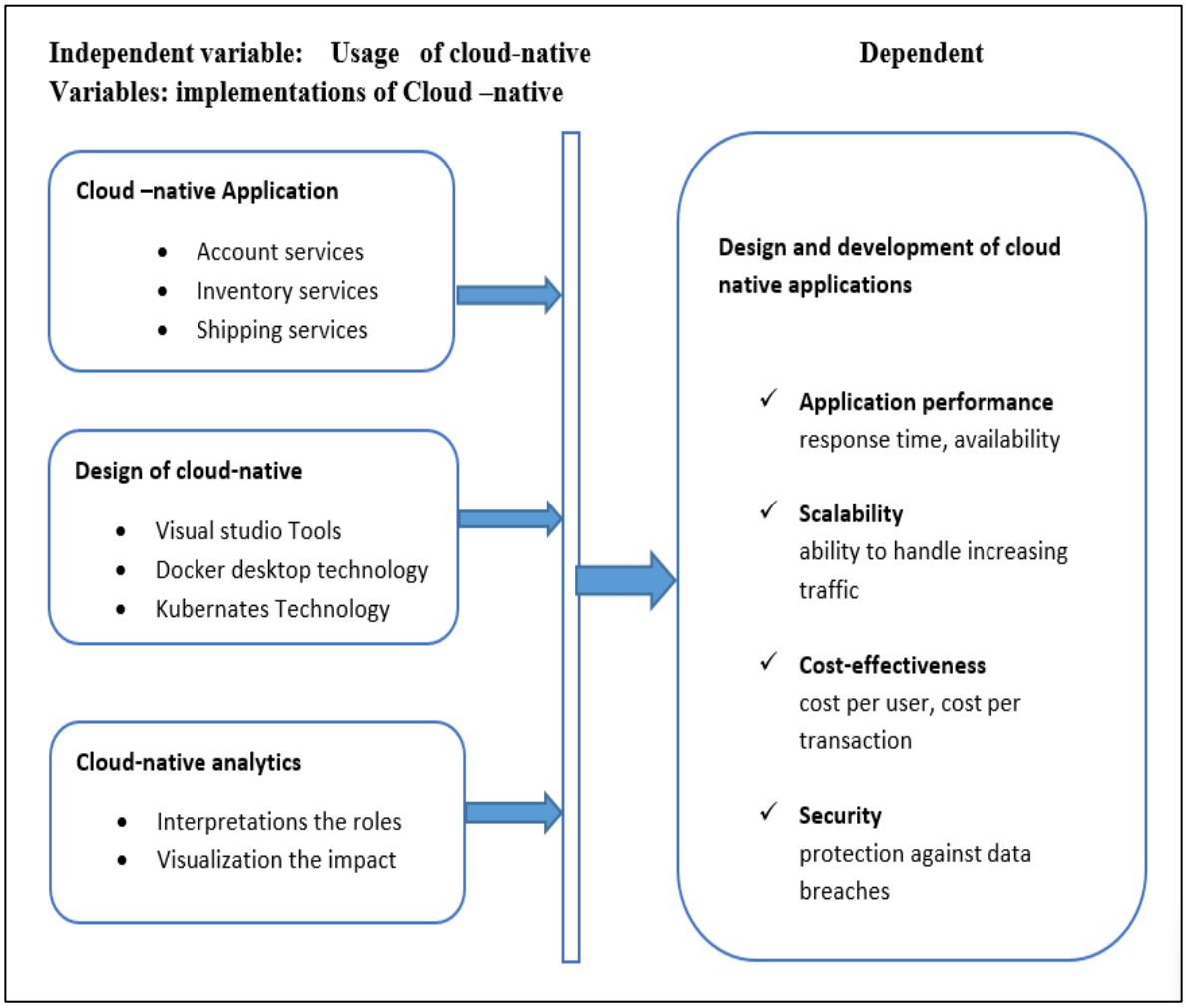

Fig 2 Conceptual Framework

➢ *Cloud-Native Design*

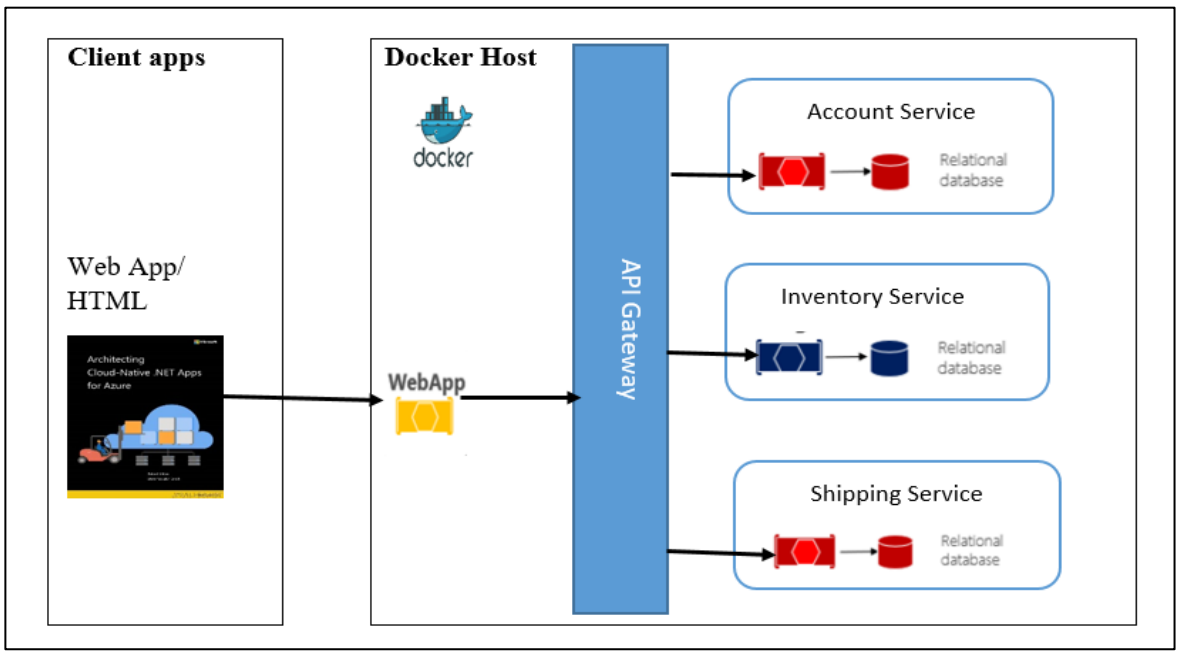

Fig 3 Cloud-Native Design

➢ *Traditional Monolithic Design*

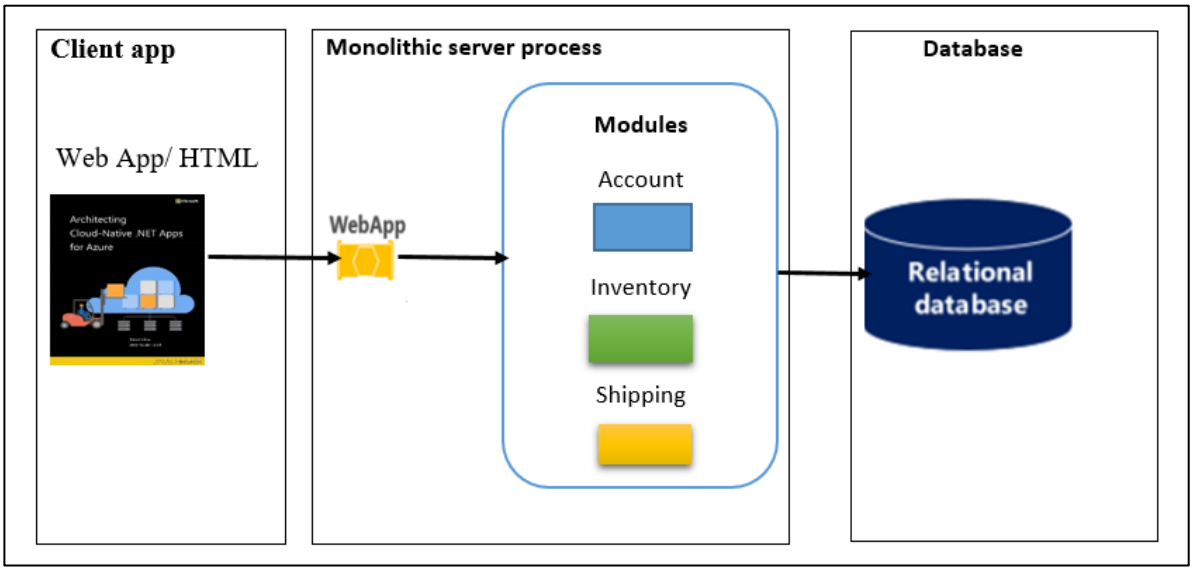

Fig 4 Traditional Monolithic Design

- *Objective:*

The study aims to investigate the current state of cloud-native application development in Rwanda, including the level of adoption, the challenges faced, and the growth opportunities. The study will also examine the advantages of utilizing cloud-native technologies in application development, such as scalability, agility, and cost-effectiveness.

- *Data Collection:*

Is the procedure of acquiring information using predetermined methodologies to respond to the study's predetermined research topic, in this study the researcher will use interviews and questionnaires as the research instruments and examine secondary data. it has been stated that approaching people with questions is an obvious way to gather quantitative as well as qualitative data from them. The survey method is used in this study to gather data. (Walliman, 2011).

- *Data Preprocessing:*

The University of Kigali officers will offer a recommendation letter that will authorize the researcher to go for data gathering. Self-administered questionnaires will be delivered to the sampled elements for Ministry of Justice.

- *Ethical Considerations:*

Address ethical considerations such as data privacy, security and trust, algorithmic bias and fairness, and intellectual property and open source.

✓ *Data Privacy:*

Cloud-native applications often handle large amounts of user data. Ethical considerations arise regarding the collection, storage, processing, and sharing of personal and sensitive information. It is essential to ensure appropriate data privacy practices, consent mechanisms, and compliance with relevant data protection regulations.

✓ *Security and Trust:*

Cloud-native applications must prioritize robust security measures to protect user data and systems from unauthorized access, breaches, and cyberattacks. Ethical considerations involve implementing strong encryption, secure access controls, regular security audits, and keeping up with evolving security threats to maintain user trust.

✓ *Algorithmic Bias and Fairness:*

Cloud-native applications that involve machine learning or data-driven algorithms should address concerns related to algorithmic bias and fairness. Ethical considerations involve ensuring fairness in decision-making, avoiding discrimination, and mitigating biases in data collection, training, and model development processes.

✓ *Intellectual Property and Open Source:*

Ethical considerations arise regarding intellectual property rights and the use of open-source software in cloud-native development. Organizations should respect licensing agreements, give proper attribution, and contribute back to the open-source community when appropriate.

III. DATA ANALYSIS AND PRESENTATION OF FINDINGS

➢ *Cloud-Native Application*

A cloud-native application is a computer program is designed to be deployed and operated in the cloud. It is built using cloud-native technologies, such as microservices, containers, and DevOps practices.

➢ *Key Characteristics of Cloud-Native Applications*

Cloud-native applications have features such as;

- *Microservices:* Cloud native applications are developed using microservices, which are small, independent services that communicate with each other using APIs. This makes the applications more modular and scalable.

- *Containers:* Containers are lightweight virtualization technology that packages an application and all its interdependencies into a single unit. This makes it easy to deploy and manage cloud computing applications.
- *DevOps:* DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). This ensures that applications are developed and deployed in a way that is efficient, reliable, and secure.
- *Auto-scaling:* Cloud native apps can be scaled up or down automatically based on demand. This helps to ensure that the applications are always running at peak efficiency.
- *Continuous delivery:* Cloud native applications are typically deployed using continuous delivery, which is a process that automates the delivery of new code to production. This helps to ensure that applications are always up-to-date and secure.
- This is the demonstration presentation of development cloud native application; researcher used various tools which includes;
- *Visual studio Code:* VS Code, short for Visual Studio Code, is a popular source-code editor developed by Microsoft. It is highly extensible, lightweight, and supports a wide range of programming languages and frameworks
- *XAMPP* is a popular open-source software package that allows you to establish a local web server environment on your Windows 10 machine. It includes Apache, MySQL, PHP, and Perl, which are essential components for running a web server and developing web applications.
- *PHP:* a shorthand for "PHP: Hypertext Preprocessor" a popular open-source scripting language used for web development. For building dynamic web pages and applications, it is frequently utilized.
- *Postman:* A well-liked platform for teamwork in API development is Postman. It makes it easier for developers to efficiently design, test, and document APIs. With Postman, you can make API requests, view responses, and analyze API performance all in one place.
- *Docker Desktop:* is a tool It enables you to run Docker containers on your local machine. It provides an easy-to-use interface for managing Docker images, containers, and networks.
- *Git Bash:* is a command-line interface (CLI) tool that provides a Unix-like environment on Windows systems. It enables the use of Git commands and other Unix utilities using the command line

➢ *Different Containers and Images Created*

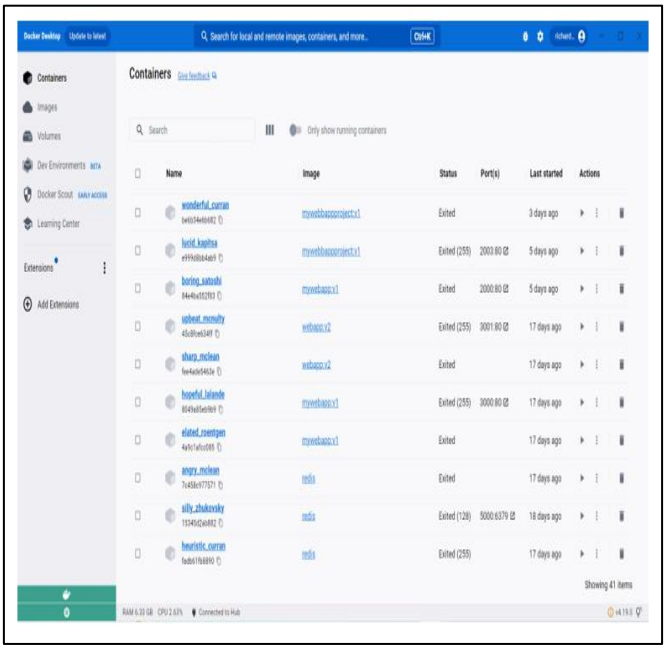

Fig 5 Docker Desktop

```
# Use nginx base image
FROM php:7.4-apache
# Copy webpages from local directory to container's default nginx html directory
WORKDIR /var/www/html
RUN docker-php-ext-install mysqli
# Copy the application files to the container
COPY . /var/www/html
# Expose port 80
EXPOSE 80
# Start nginx service
CMD ["apache2-foreground"]
FROM mysql:5.7
ENV MYSQL_ROOT_PASSWORD=password
ENV MYSQL_DATABASE=dbproject
EXPOSE 3306
version: '3'
services:
  php-app:
    build:.
    ports:
      - "2001:80"
  mysql-db:
    build:
      context:.
      dockerfile: Dockerfile.mysql
    ports:
      - "3306:3306"
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
```

IV. CONCLUSIONS AND DISCUSSIONS ON THE FINDINGS

➢ *Summary*

The exploration of creating and creating cloud-native software has revealed a transformative contemporary software architecture strategy. With the help of cloud-native apps, businesses can increase the productivity, adaptability, and resilience of their software systems while taking full advantage of the dynamic and scalable nature of cloud environments. This exploration delved into key principles such as microservices architecture, containerization, and DevOps practices, highlighting their role in facilitating rapid development, Continuous integration and delivery are both practices. The adoption of cloud-native strategies requires a fundamental shift in mindset, focusing on modularity, automation, and seamless scaling. As organizations continue to embrace cloud-native paradigms, they are better positioned to meet the evolving demands of digital business landscapes and deliver innovative, reliable, and efficient applications.

➢ *Conclusion:*

Evolution of the design and cloud-native applications represents a pivotal advancement in software engineering. This exploration underscores the significance of decoupling monolithic applications into microservices, encapsulated within containers. This architecture not only enables independent development and scaling of components but also enhances fault tolerance through isolation. Embracing DevOps practices further accelerates the development lifecycle, enabling frequent releases and continuous improvement. While transitioning to cloud-native approaches demands careful planning, including consideration for security, monitoring, and resource management, the benefits in terms of agility and scalability are substantial. As organizations worldwide continue to migrate their applications to the cloud and adopt cloud-native strategies, the software landscape is poised for a paradigm shift.

REFERENCES

[1]. Andrikopoulos, V., Strauch, S., Fehling, C., and Leymann, F. (2012). CAP-Oriented Design for Cloud-Native Applications. In Proceedings of the 2nd International Conference on Cloud Computing and Service Science, CLOSER 2012, 18-21 April 2012, Porto, Portugal, pages 365–374. SciTePress

[2]. Barr, J. (2010). The history of cloud computing. O'Reilly Radar. Retrieved from https://www.oreilly.com/radar/the-history-of-cloud-computing/

[3]. Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). Mastering cloud computing. McGraw-Hill Education.

[4]. Blog, Elastisys Tech (2022-05-16). "Cloud Native: why bother, its benefits, and its greatest pitfall". elastisys. Retrieved 2022-11-08.

[5]. Brunner, S., Blochlinger, M., Toffetti, G., Spillner, J., and Bohnert, T. M. (2016). Experimental Evaluation of the Cloud-Native Application Design. In Proceedings – 2015 IEEE/ACM 8.

[6]. Pahl, "Containerization and the PaaS Cloud," in IEEE Cloud Computing, vol. 2, no. 3, pp. 24-31, May-June 2015, doi: 10.1109/MCC.2015.51.

[7]. Codallo, Ana. "Council Post: Building A Tech Stack For Wartime Economy: Six Things I Learned". Forbes. Retrieved 2022-11-08.

[8]. Cockcroft, A. (2014, December). DockerCon Europe: Adrian Cockcroft in regards to Microservices. Retrievedfromthenewstack.io: https://thenewstack.io/dockerconeurope-adrian-cockcroft-on-the-state-of-microservices/

[9]. Cloud Native Foundation, Frequently Asked Questions, https://www.cncf.io/about/faq/

[10]. Gannon, R. Barga and N. Sundaresan, "Cloud-Native Applications," in IEEE Cloud Computing, vol. 4, no. 5, pp. 16-21, September/October 2017, doi: 10.1109/MCC.2017.4250939.

[11]. CNCF Cloud Native Definition v1.0". GitHub(CNCF). 2018-06-11. Retrieved 2020-05-15.

[12]. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2016, June 13). Microservices: yesterday, today, and tomorrow. Retrieved from arxiv.org: https://arxiv.org/pdf/1606.04036v1.pdf

[13]. Gannon, Dennis & Barga, Roger & Sundaresan, Neel. (2017). CloudNative Applications. IEEE Cloud Computing. 4. 16-21. 10.1109/MCC.2017.4250939.

[14]. https://mars.gmu.edu/bitstream/handle/1920/11608/hassan_cloud.pdf?sequence=1

[15]. https://www.geeksforgeeks.org/architecture-of-cloud-computing/amp/

[16]. https://www.devteam.space/blog/microservice-architecture-examples-and-diagram/

[17]. https://www.ridge.co/blog/docker-vs-kubernetes-whats-the-difference/

[18]. https://www.atlassian.com/microservices/microservices-architecture/microservices-vs monolith#:~:text=A%20monolithic%20architecture%20is%20a,monolith%20architecture%20for%20software%20design.

[19]. https://www.sciencedirect.com/science/article/abs/pii/S0164121217300018

[20]. https://www.ijert.org/research/impact-of-implementing-cloud-native-applications-in-replacement-to-on-premise-applications-IJERTV9IS061021.pdf

[21]. https://www.commvault.com/blogs/cloud-native-applications-and-containerization

[22]. https://www.commvault.com/blogs/cloud-native-applications-and-containerization#:~:text=With%20containers%2C%20all%20the%20dependencies,container%20making%20them%20more%20predictable.

[23]. https://www.techtarget.com/searchcloudcomputing/definition/cloud-native-application

[24]. https://openliberty.io/docs/latest/cloud-native-microservices.html#:~:text=Cloud%2Dnative%20applications%20adapt%20microservice,scale%2C%20and%20upgrade%20the%20microservices.

[25]. International Utility and Cloud Computing Conference, UCC 2015, pages 488–493

[26]. Kratzke, N. and Peinl, R. (2016). ClouNS-a Cloud-Native Application Reference Model for Enterprise Architects. In Proceedings – IEEE International Enterprise Distributed Object Computing Workshop, EDOCW, volume 2016-Septe, pages 198–207. IEEE
[27]. Kratzke, N. and Quint, P.-C. (2017). Understanding cloud-native applications after 10 years of cloud computing – A systematic mapping study. Journal of Systems and Software, 126:1–16.