Green Microservices: Optimizing Resource Utilization and Maximizing the Performance of Heterogeneous Cloud Servers

***Abstract*— Autoscaling microservices have emerged as a critical technique for modern web applications to efficiently manage resources, maintain performance, and optimize costs in dynamic environments. Despite being a common design architecture, it suffers from issues like the "cold start problem" and issues with handling large spikes in usage. This paper explores the implementation of a common resource broker that can be easily tweaked per service basis and scripted to allow easy autoscaling of microservices using Kubernetes orchestration on a cloud service. Along with the dynamic broker, Kubernetes' robust infrastructure and cloud providers' scalable resources, developers can easily manage, and scale microservices with ease. Implementation of autoscaling features such as Horizontal Pod Autoscaler (HPA) and Cluster Autoscaler is setup automatically. We highlight the impacts of autoscaling microservices including scalability, cost optimization, and increased reliability which in turn reduces the overall power consumption and resource usage.**

Keywords*— *Autoscaling microservices, Cold start problem, Green Cloud Computing, Energy Efficiency, Horizontal Pod Autoscaler, Kubernetes orchestration.

I. INTRODUCTION

Green Cloud Computing, an evolution of green computing principles applied to cloud environments, offers innovative solutions to the challenges of resource management, performance optimization, and energy efficiency in modern web applications. This approach integrates various strategies to enhance sustainability in IT infrastructure while maintaining high performance. Key advancements include optimized virtual machine allocation to reduce energy consumption [19], power-aware scheduling techniques [20], and multi-layered energy-efficient architectures [21]. Green Cloud Computing aims to create more resilient, efficient, and environmentally friendly cloud infrastructures. This paradigm shift in cloud technology not only optimizes resource utilization but also significantly contributes to reducing the carbon footprint of rapidly expanding digital ecosystems.

Our research explores the strategic implementation of a common resource broker, enhanced by the innovative use of Kubernetes, an orchestration platform that excels in managing containerized applications across a cloud infrastructure. Kubernetes facilitates not only straightforward deployment and management but also enables sophisticated autoscaling mechanisms such as the Horizontal Pod Autoscaler (HPA) and the Cluster Autoscaler. These tools dynamically adjust the number of active server instances, ensuring efficient resource usage and consistent application performance without human intervention.

Furthermore, the integration of Kubernetes Event-driven Autoscaling (KEDA) allows for advanced scaling based on real-time event data, offering a more granular control over microservices scaling. KEDA extends Kubernetes' native autoscaling capabilities by interfacing with various event sources, thus tailoring resource allocation more precisely to current workloads and system demands.

Our work delves into how these technologies collectively support the scaling of microservices, thereby enhancing scalability, reducing costs, and improving performance and reliability. The broader implications of these technologies on power consumption and resource utilization are also examined, underscoring the transformative potential of advanced autoscaling in cloud-based microservices environments. Through a detailed analysis of these systems, this research aims to illustrate their impact on the modern web applications landscape and provide insights into their operational benefits and challenges.

Green Cloud Computing, also called sustainable IT, is an environmentally friendly approach using information technology that maximizes energy efficiency and optimizes resource utilization. This has become a major research challenge in recent years. The environmental impact of the IT industry has become increasingly evident due to its substantial carbon footprint. The energy needs of data centers, servers, and personal computing devices are staggering, leading to large amounts of greenhouse gas emissions and thus contributing to global warming and climate change. The IT industry can make a significant positive impact on the environment by implementing green cloud computing practices, making operations greener at all levels.

This research conducted a survey aimed to provide a comprehensive understanding of green cloud computing awareness and adoption among different demographic groups in India. It collected data on participants' age, occupation, location, familiarity with green computing, and the practices they engage in. The survey also sought suggestions for improving the adoption of green computing practices and assessed the effectiveness of corporate initiatives in promoting green computing.

The data collected includes responses from individuals of India from various age groups, occupations, and demographic areas (urban and rural), offering insights into their familiarity with green computing and their familiarity in the terminology, adoption of green computing devices and their suggestions for improving adoption not only in aspect to the product but also how companies need focus themselves with green computing. The majority of respondents were in the 18-24 age group. This demographic is likely more exposed and educated about technology and environmental issues and are aware of green computing. This is also reflected in their responses. Occupation wise, the survey saw participation from students and IT professionals. The location distribution showed a higher representation from urban areas compared to rural areas, highlighting the bias of this demographic in accessing and answering this survey. Despite very high ownership of Mobile devices and the easy access to internet connection throughout India [18], the participants were mostly from urban

areas indicating another bias towards richer demographics engaging in this survey.

The remainder of the paper is structured as follows, Section II discusses the related work, Section III encompasses the proposed methodology, including Docker setup, container building, and the data flow diagram. Finally, Section IV presents the results and discussion, addressing accuracy and future avenues for research.

II. Related Work

This survey covers autoscaling systems designed for cloud environments and microservices. The first section highlights sophisticated solutions such as container-aware scheduling and model-driven controllers like ATOM, that maximises resource utilisation and save expenses. The second segment focuses on specialised solutions for certain contexts, such as bursty workloads and CloudIoT systems, with researchers offering customised approaches for managing workload variations while retaining performance. Finally, the third portion examines future-oriented techniques and autonomous autoscaling systems, emphasising the transition towards proactive resource management via predictive analytics and reinforcement learning. These efforts seek to dynamically adapt to changing resource demands, guaranteeing effective resource utilisation while meeting service quality restrictions in dynamic and heterogeneous situations. Microservices are contributing significantly to making data centres greener by enabling more efficient resource utilization and energy consumption. The modular nature of microservices allows for fine-grained scaling, where only the necessary components are scaled up or down based on demand, rather than scaling entire monolithic applications. This granular control leads to more precise resource allocation, reducing overprovisioning and idle resource waste. Furthermore, microservices facilitate better workload distribution across data centre resources, allowing for more effective load balancing and potentially reducing the number of active servers during low-demand periods. The containerization often associated with microservices also contributes to energy efficiency by enabling higher resource density and faster deployment, which can lead to reduced overall energy consumption. By enabling more efficient autoscaling and resource management, microservices are playing a crucial role in the development of greener, more sustainable data centres that align with the principles of Green Cloud Computing.

A. Advanced Autoscaling Techniques for Cloud and Microservices

Several recent studies have offered novel techniques to address the issues of autoscaling in cloud settings and microservices. Satish Narayana Srirama et al. [1] presented a container-aware application scheduling technique with auto-scaling strategies that prioritises resource optimisation and cost reduction via dynamic modification of container instances. Alim Ul Gias et al. [2] introduced ATOM, a model-driven autoscaling controller specifically developed for microservices, which uses a layered queueing network model to dynamically assign resources and assure peak performance. Meanwhile, Nathan Cruz Coulson et al. [3] worked on auto-scaling web apps and IoT scenarios, using a hybrid sequence and supervised learning model to offer scaling strategies based on workload trends. These studies illustrate the necessity of autoscaling systems that take into account the characteristics and constraints of cloud-based architectures and various demand patterns, opening up new paths for optimising resource management and improving system performance.

B. Specialized Autoscaling Solutions for Varied Environments

In the field of autoscaling, researchers have focused on tackling specific issues encountered in various contexts, including bursty workloads and CloudIoT systems. Muhammad Abdullah et al. [4] presents a burst-aware autoscaling solution designed to successfully manage unexpected demand spikes in cloud-hosted microservices. Their solution includes proactive detection of workload bursts using previous performance traces and periodic model retraining, which allows for appropriate resource allocation modifications to avoid reaction time violations. Meanwhile, Manuel Gotin et al. [5] investigates threshold-based auto-scaling algorithms optimised for CloudIoT systems. Their research intends to ensure performance and reliability in dynamic IoT environments by looking into the selection of relevant performance metrics and threshold values. These studies highlight the crucial need of tailoring autoscaling approaches to the specific needs and features of distinct application domains, allowing for optimal resource management and system stability in a variety of computing environments.

C. Future-Oriented Approaches and Autonomous Autoscaling Systems

In addition to tackling existing issues, researchers are actively exploring innovative techniques and autonomous systems to further the area of autoscaling. Autoscaling is a crucial feature in cloud computing that automatically adjusts the number of computational resources allocated to an application based on its current demand. This dynamic resource allocation is closely related to Green Cloud Computing, as it helps optimize energy consumption by ensuring resources are used efficiently. Autoscaling contributes to Green Cloud Computing by minimizing energy waste. When demand is low, it reduces the number of active servers, thereby decreasing power consumption. Conversely, when demand increases, it adds resources to maintain performance without over-provisioning. This balance between performance and energy efficiency is at the core of Green Cloud Computing principles. Shutian Luo et al. [6] offers Madu, a proactive auto-scaler specifically designed for microservice frameworks. Madu's functionality is based on predicted workload analysis and performance profiling to enable optimal resource utilisation. Madu uses complex workload prediction models and performance profiling techniques to proactively alter resource allocation in anticipation of changing needs, improving system efficiency and responsiveness. Similarly, Abeer Abdel Khaleq et al. [7] present an autonomous autoscaling system that can automatically adjust to changing application resource requirements. This system uses generic autoscaling algorithms and reinforcement learning agents to dynamically optimise resource allocation decisions. The autonomous autoscaling system can efficiently respond to changing workload patterns and assure adherence to quality of service limitations in dynamic situations by learning and adapting over time. These findings emphasise the crucial role of proactive and adaptive autoscaling systems in guaranteeing efficient resource management and attaining performance goals in dynamic computing environments, while also supporting the objectives of Green Cloud Computing.

Academic research in green computing underscores the importance of eco-friendly practices in efficiently managing computer resources. Dr. Beena B.M et al. [13] analyse the Interquartile Range (IQR) VM algorithm, proposing an optimized version to address its drawback of relying solely on

CPU utilization for host assessment. This adjustment aims to enhance energy efficiency and reduce SLA violations. Meanwhile, Shinu M. Rajgopal et al. [14] introduce a meta-heuristic-based resource provisioning model for IoT microservices in smart healthcare systems, achieving significant efficiency improvements through modified genetic and flower pollination algorithms. Additionally, Sanghamitra Nemmini et al. [15] explore the security implications of monolithic and microservice architectures, emphasizing the potential for heightened security in microservices with additional effort and expertise. Future research should focus on fortifying security measures in both models to advance secure software development practices. Shutian Luo et al. [6] offers Madu, a proactive auto-scaler specifically designed for microservice frameworks. Madu's functionality is based on predicted workload analysis and performance profiling to enable optimal resource utilisation. Madu uses complex workload prediction models and performance profiling techniques to proactively alter resource allocation in anticipation of changing needs, improving system efficiency and responsiveness. Similarly, Abeer Abdel Khaleq et al. [7] present an autonomous autoscaling system that can automatically adjust to changing application resource requirements. This system uses generic autoscaling algorithms and reinforcement learning agents to dynamically optimise resource allocation decisions. The autonomous autoscaling system can efficiently respond to changing workload patterns and assure adherence to quality of service limitations in dynamic situations by learning and adapting over time. These findings emphasise the crucial role of proactive and adaptive autoscaling systems in guaranteeing efficient resource management and attaining performance goals in dynamic computing environments. Additionally, this paper by K. Dinesh Kumar et al. [16] proposes an efficient proactive VM consolidation technique utilizing an improved LSTM network to manage cloud resources effectively, thereby reducing power consumption and avoiding SLA violations. By predicting non-linear workload patterns with high accuracy, the technique minimizes unnecessary VM migrations and enhances resource management performance, demonstrating superior performance compared to conventional VM consolidation techniques.

The research gap in these studies show a wide variety of concerns in the field of autoscaling for cloud environments, microservices, and emerging computing paradigms. Common challenges include dynamically allocating resources to meet changing workload needs while preserving scalability, resource optimisation, and cost-effectiveness. Integrating autoscaling approaches with container-based deployment models and complex microservice architectures presents compatibility and organisational problems that necessitate precise deployment procedures and stakeholder interaction. Furthermore, problems persist in effectively predicting workload changes, optimising resource allocation to decrease response time violations, and efficiently managing bursty workloads in cloud-hosted microservice environments. Furthermore, autoscaling systems scalability and flexibility across different demand patterns, microservice architectures, and computing environments remain a cause of concern. Furthermore, there is a considerable need to develop new autoscaling approaches in fog computing settings, such as accurately forecasting workloads and meeting specified service-level objectives while minimising resource overhead and operational costs. Another increasing challenge is the effective use of resources to support green computing projects. Ensuring that autoscaling not only satisfies performance and scalability goals, but also reduces energy consumption and environmental effect, is critical for modern computing system longevity. This initiative especially tackles these issues by emphasising resource utilisation and green computing. It investigates the use of Kubernetes to orchestrate containerised applications while assuring effective resource utilisation using technologies such as the Horizontal Pod Autoscaler (HPA) and Cluster Autoscaler. These tools dynamically modify the number of active server instances, optimising resource utilisation while lowering total power and resource consumption. This project uses Docker and Kubernetes to optimize the system and make it faster, adhering to green IT practices that, in the grand scheme of things, contribute to sustainable computing. This is extremely important in closing these research gaps and avoiding further challenges in autoscaling and resource management, portability, security, and improving scalability and performance in dynamic computing systems. This focus on green computing in the study would ensure that technological advancement moves towards sustainability while ensuring this is highly effective and environmentally responsible.

III. Proposed Methodology

The process begins with Docker setup. Docker containers enclose application components, ensuring consistency and portability across environments. They are precisely tuned to maximise resource utilisation. These containers are orchestrated within Kubernetes Pods, allowing for easy management and scaling. Furthermore, using a Resource Broker, specifically an Event-Driven Autoscaler, automates scaling based on dynamic workload demands while smartly allocating resources. Triggers are designed to start scaling actions depending on various criteria, ensuring optimal performance and resource utilisation. The deployment process is completed by configuring infrastructure on AWS EC2 instances and S3 storage. EC2 instances offer scalable computational capacity for hosting the Kubernetes cluster, and S3 storage ensures consistent access and durability for static assets.

A. Setting up E-commerce website

This project builds an e-commerce website using a stack of HTML, CSS, and Node.js to create a dynamic and visually appealing platform. HTML is the cornerstone of our website's structure, allowing us to organise material and define page layout. CSS is used to style and improve the visual display of our website, maintaining uniformity and aesthetic appeal across all devices and screen sizes. Meanwhile, Node.js provides a server-side runtime environment, allowing us to create scalable and efficient web apps. Using Node.js, helps in developing server-side logic, handle user authentication, and connect with databases to manage product listings, user accounts, and transactions. The website is engaging and feature-rich that provides the customers with a seamless purchasing experience by integrating various technologies.

B. Docker setup

This research uses Docker Engine as the primary runtime environment for the Docker setup, which allows it to efficiently construct and manage containers. Docker Engine seamlessly encapsulates application components into containers, ensuring consistency and simplicity of deployment across several environments. Furthermore, this project makes use of Docker Hub as the central registry for storing and managing container images. This enables us to simply share

and distribute Docker images, speeding up collaboration and deployment procedures. By combining Docker Engine and Docker Hub, there is a solid foundation for our containerised application deployment, enabling agility and scalability in this cloud computing project.

C. Building containers

This study takes a containerised approach, building each component of our e-commerce website as a Docker container. Containers are lightweight, portable, and self-contained environments that include all of the dependencies needed to run a single piece of software. Each container contains the required code, runtime, system tools, and libraries, ensuring consistency and predictability across environments. This containerisation technique allows us to isolate and control different components individually, allowing for more efficient deployment, scaling, and maintenance of our application. By utilising Docker containers, we can achieve better flexibility, efficiency, and agility in our cloud computing infrastructure, allowing us to provide a strong and scalable e-commerce platform.

D. Setting up pods

In this research, pods are defined as the core unit for deploying and managing Docker containers within the Kubernetes ecosystem as shown in Fig. . Pods act as the execution environment for one or more containers that are closely coupled, allowing them to share resources like network and storage. Each Pod is designed to house the containerised components of our e-commerce website, ensuring a smooth functioning and effective resource utilisation. By clustering containers within Pods, we simplify deployment and scaling operations, allowing for seamless coordination and communication between application components. This technique improves reliability and scalability by allowing pods to be dynamically scaled up or down based on workload demands, contributing to the robustness and performance of our cloud-based e-commerce platform.

E. Resource Broker

In this study, the Resource Broker, which employs Event-Driven Autoscaling, is a critical component that enables dynamic resource allocation based on real-time workload changes. It does not mandate certain scaling techniques, but rather provides a flexible framework for integrating with various event sources and making scaling decisions based on the data they provide. It leverages its capabilities within our Kubernetes cluster to provide tailored scaling techniques that are specific to our application's demands. For example, we can use the Horizontal Pod Autoscaler (HPA) algorithm to scale deployments depending on CPU or memory consumption data, resulting in efficient resource utilisation and responsiveness to changing demands.

Furthermore, it allows us to set custom metrics such as queue length or message processing rates, allowing us to base scaling decisions on application-specific criteria. This adaptability enables us to tailor our autoscaling behaviour to the peculiarities of our e-commerce business, resulting in optimal performance and resource efficiency. Furthermore, it opens the door to more advanced scaling approaches, such as machine learning-based algorithms or threshold-based scaling, which can improve our system's ability to anticipate and respond to workload patterns effectively. By incorporating it into our project, it provides a strong foundation for autoscaling, which improves the scalability, dependability, and efficiency of our cloud-based e-commerce platform.

F. Deploying on AWS EC2 and S3

In this research, the deployment on AWS EC2 instances and S3 storage are seamlessly coordinated to provide a solid architecture for hosting our e-commerce application. It uses AWS EC2 instances as the foundation for the cloud environment to provide scalable computing resources for the Kubernetes cluster, assuring excellent performance and availability. It dynamically scales the resources using EC2 to meet shifting workload demands while preserving responsiveness and efficiency. Additionally, it uses AWS S3 storage to securely store static assets like photos and scripts, assuring consistent access and durability throughout our system. By including EC2 and S3 into the deployment strategy, a resilient and scalable infrastructure is built that allows the e-commerce platform to provide users with a seamless and reliable experience.

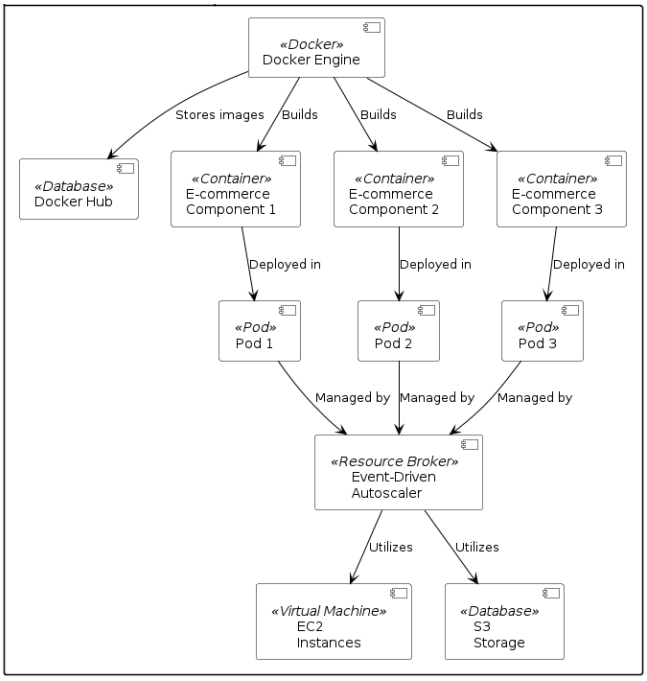

FIG. 1. COMPONENT DIAGRAM OF AN AUTOSCALING MICROSERVICES SYSTEM FOR E-COMMERCE PLATFORM DEPLOYMENT ON CLOUD INFRASTRUCTURE

IV. Results And Discussion

This section focuses on the outcomes of the proposed work. The unified resource broker demonstrated the capability to dynamically scale microservices on the cloud in response to application events, providing a more responsive and efficient scaling mechanism compared to traditional metric-based approaches. This event-driven scaling strategy ensures that resources are allocated based on actual application demands, optimizing performance and resource utilization. Compared to direct hosting on AWS, the implementation showcased a 10% lower CPU utilization, highlighting the efficiency and cost-effectiveness of utilizing the unified resource broker for autoscaling microservices. This reduction in CPU usage contributes to lower operational costs and improved resource management. The open-source nature of the resource broker allows for easy customization and extensibility based on specific use cases. Its vendor-agnostic approach enables seamless integration with various cloud providers and supports multiple workloads, enhancing flexibility and adaptability in diverse environments. By achieving lower CPU utilization and reduced downtime, the unified resource broker directly contributes to minimizing the environmental impact of large web applications. The carbon footprint can be quantified by considering the total power usage of cloud-based machines, with reduced downtime leading to decreased power spikes during redeployment and startup processes. This environmental consideration aligns with sustainable practices and energy efficiency in cloud computing.

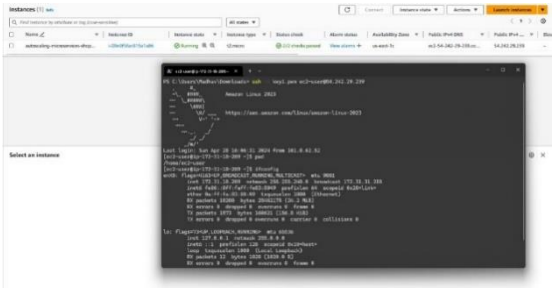

FIG.3. Connected remote EC2 instance

Fig. 1. Shows AWS EC2 instance being used to provide scalable computing resources in the cloud. Leveraging Docker containers on EC2 offers flexibility and efficiency in deploying applications. It also makes it easy to scale with the resource broker. To begin an SSH connection is established to their EC2 instance from a local terminal, granting remote access to the server. Once connected, navigate to their project directory containing Dockerfiles Fig. 2. and related configurations. Docker images are built locally, specifying dependencies and configurations within the Dockerfile. These images can then be pushed to a Docker registry if needed. Subsequently, using Docker commands or orchestration tools like Docker Compose, containers are deployed onto their EC2 instance, configuring networking, storage, and other parameters as required. These are directly parsed into the resource broker. This approach streamlines application deployment, enabling easy scalability and management within the AWS cloud environment with the power to handle the configurations of both docker and AWS from a single platform.

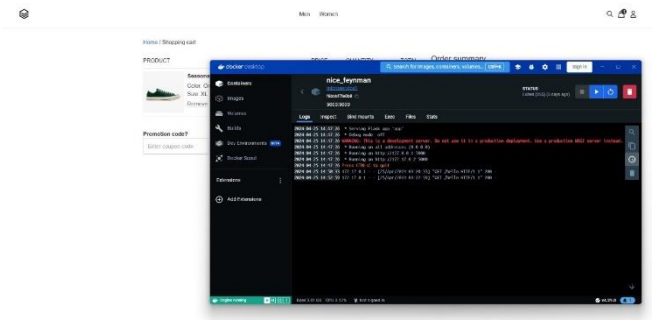

FIG.2. Connected remote EC2 instance and deployed docker containers

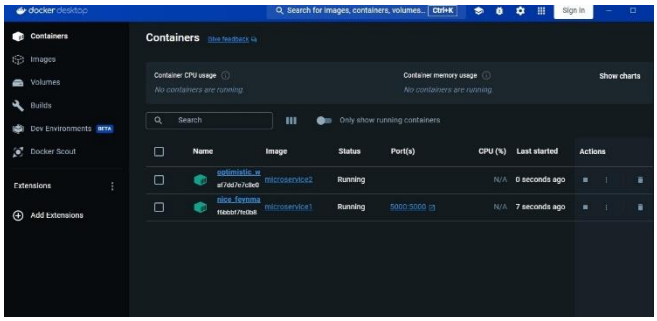

FIG.3. Docker Hub instances running

In Fig. 3. Docker Desktop a user-friendly interface for managing Docker containers on local machines is used. The Containers tab within Docker Desktop provides a visual representation of running containers, simplifying container management tasks. Users can easily access the Containers tab from the Docker Desktop application, which displays essential information such as container status, resource usage, and networking details. To deploy containers, navigate to the Containers tab, either pull pre-built images from Docker Hub or build custom images locally using Dockerfiles. Once images are selected or created, we can start containers directly from the Containers tab with a single click, specifying ports, volumes, and other configurations as needed.

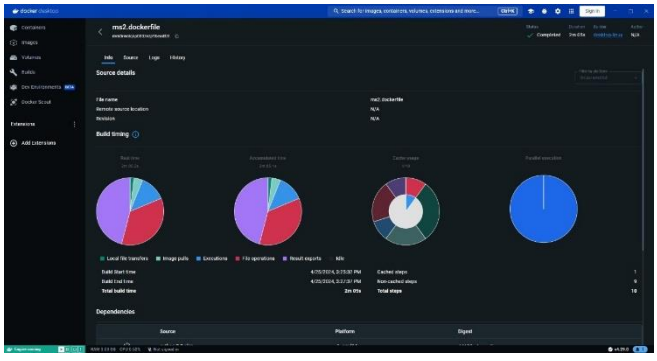

FIG . 4. Microservice Instance Analysis

Fig. 4. shows the analytics of the Docker instance in Docker Desktop, a visual representation of Dockerfile instances. It provides as a platform to view, create, and modify

Dockerfiles directly within the interface. This intuitive interface simplifies the process of building Docker images from Dockerfiles, streamlining the development process. It also has a visual pie chart for easy monitoring of resources and downtime. Coupled with the resource broker configuration one can easily manage and scale large web applications.

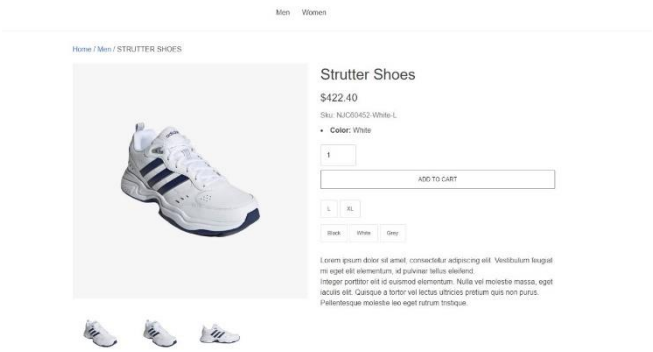

FIG . 5. DEMO EVERSHOP WEBSITE RUNNING

Fig. 5. show a demo interface of the evershop demo app, the web app we have deployed and orchestrated using AWS, Docker and the custom resource broker. The webapps were divided based on the functionality, here i) Shopping Page, ii) Shopping Cart, iii) Billing Page and finally iv) Database. This also allows the individual components to be run, tested, upgraded and monitored separately improving the developer experience, reliability and overall stability of the web application.

This section focuses on the outcomes of the survey on green computing practices and their effects. Respondents rated their familiarity with green computing on a scale from 1 to 5, with 1 being the highest and 5 the lowest. Most respondents rated their familiarity between 2 and 3, indicating a moderate to high awareness level. This suggests that while there is a general understanding of green computing, there is still room for increasing awareness and deeper understanding. When asked to describe green computing, many respondents selected specific practices such as using energy-efficient hardware. 43.4% of the participants couldn't fully identify all the correct practices among the most common green computing practices. This indicates a partial understanding and application of green computing. In terms of practices, common activities included using energy-efficient hardware and enabling power-saving modes, with fewer respondents engaging in practices like reducing paper receipts and green manufacturing processes for hardware. Another significant finding is the misconception regarding cloud computing. Many respondents considered using cloud computing as a substitute to using personal hardware as a green computing alternative. This oversight may have arisen due the poor understanding of cloud services among the general public. Datacenters are massive energy consuming sites and can even utilize significant portions of a city's water and electricity [17]. This underscores the need for more targeted education and awareness efforts about cloud and datacenter infrastructure. Additionally, suggestions by respondents for improving adoption, such as "more awareness by companies" and "government policies" highlight the need for increased initiatives needed to be taken by the corporations to boost understanding and adoption of green computing. Furthermore, the effectiveness of corporate initiatives in promoting green computing was rated between 1 and 3, suggesting that current efforts have huge scope to be effective or recognized by consumers.

Analyzing the correlation between age groups or urban and non-urban areas along with green computing awareness revealed interesting trends. Younger respondents those in urban areas showed higher familiarity with green computing practices. Urban respondents generally rated corporate initiatives more favorably, indicating better access to information and resources related to green computing. This is also aided by the fact that urban consumers are more likely to be customers of these corporations and are aware of green computing initiatives. In contrast, older age groups and non-urban respondents displayed lower understanding of green computing. To enhance the adoption of green computing, educational institutions, corporations and policymakers should focus on programs, schemes to improve green computing practices in India. Workshops and training sessions can be provided to educate employees and students. Awareness campaigns should emphasize the full scope of green computing, including its environmental and long-term economic benefits. Corporations should develop and communicate clear policies on green computing, like reducing e-waste or sustainable turn in programs to reuse hardware. Transparency in electricity usage, carbon footprint must also be a mandate as it will help converting them into a metric to target. Adoption of practices beyond just using energy-efficient hardware needs to be considered. Additionally, providing incentives for adopting green technologies, such as subsidies or tax brackets for such spending can motivate more businesses to adopt green computing.

V. CONCLUSION

In the current landscape of digital infrastructure, the energy-intensive operation of large data servers poses significant environmental challenges, especially amidst escalating concerns over water stress, energy consumption, and climate change impacts. The pressure from corporations to cool massive data centers in water-stressed areas highlights the critical need for sustainable practices in technology. Our research focuses on integrating green coding principles, efficient algorithms, and optimal resource utilization to deliver seamless services while prioritizing sustainability. By implementing autoscaling strategies that dynamically adjust resource allocation based on actual demand, we minimize resource wastage and enhance operational efficiency. The unified resource broker serves as a sophisticated system manager, capable of fine-tuning clusters to meet application requirements effectively, thereby reducing environmental impact and ensuring cost-effective scalability in modern applications. This approach not only addresses the urgency of climate action but also emphasizes the importance of responsible resource management in the digital age. The survey conducted indicates a level of awareness regarding green computing, particularly among younger and urban respondents, indicating a solid base to expand on. There is enormous room for improvement in understanding and implementing comprehensive green practices, where education and regulations set will make a significant impact. Corporate transparency, clear green computing strategies, and incentives can help to accelerate wider adoption and beneficial environmental impact. Future work involves advancements in container orchestration platforms, integration of AI and machine learning, enhancing security

protocols, broader adoption in edge computing, and continued innovation in the unified resource broker technology to further enhance capabilities, scalability, and environmental sustainability, positioning it as a leading solution for efficient and eco-friendly microservices deployment in cloud environments.

REFERENCES

[1] Srirama, S.N., Adhikari, M. and Paul, S., 2020. Application deployment using containers with auto-scaling for microservices in cloud environment. *Journal of Network and Computer Applications*, *160*, p.102629.

[2] Gias, A.U., Casale, G. and Woodside, M., 2019, July. ATOM: Model-driven autoscaling for microservices. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 1994-2004). IEEE.

[3] Coulson, N.C., Sotiriadis, S. and Bessis, N., 2020. Adaptive microservice scaling for elastic applications. *IEEE Internet of Things Journal*, *7*(5), pp.4195-4202.

[4] Abdullah, M., Iqbal, W., Berral, J.L., Polo, J. and Carrera, D., 2020. Burst-aware predictive autoscaling for containerized microservices. *IEEE Transactions on Services Computing*, *15*(3), pp.1448-1460.

[5] Gotin, M., Lösch, F., Heinrich, R. and Reussner, R., 2018, March. Investigating performance metrics for scaling microservices in cloudiot-environments. In *Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering* (pp. 157-167).

[6] Luo, S., Xu, H., Ye, K., Xu, G., Zhang, L., Yang, G. and Xu, C., 2022, November. The power of prediction: microservice auto scaling via workload learning. In *Proceedings of the 13th Symposium on Cloud Computing* (pp. 355-369).

[7] Khaleq, A.A. and Ra, I., 2021. Intelligent autoscaling of microservices in the cloud for real-time applications. *IEEE Access*, *9*, pp.35464-35476.

[8] Yu, G., Chen, P. and Zheng, Z., 2019, July. Microscaler: Automatic scaling for microservices with an online learning approach. In *2019 IEEE International Conference on Web Services (ICWS)* (pp. 68-75). IEEE.

[9] Abdullah, M., Iqbal, W., Mahmood, A., Bukhari, F. and Erradi, A., 2020. Predictive autoscaling of microservices hosted in fog microdata center. *IEEE Systems Journal*, *15*(1), pp.1275-1286.

[10] https://time.com/5814276/google-data-centers-water/

[11] https://cc-techgroup.com/data-center-energy-consumption/

[12] https://www.forbes.com/sites/forbesbooksauthors/2023/06/21/how-green-is-your-data-center-optimize-code-to-reduce-greenhouse-gas-emissions/

[13] Nadaf, A., 2023, May. Green Computing: Optimized Underutilized Host detection in IQR Vm Allocation Policy. In 2023 4th International Conference on Intelligent Engineering and Management (ICIEM) (pp. 1-4). IEEE.

[14] Rajagopal, S.M., Supriya, M. and Buyya, R., 2023. Resource Provisioning using Meta-heuristic Methods for IoT Microservices with Mobility Management. IEEE Access.

[15] Nemmini, S., Abhishek, S., Anjali, T. and Raj, S., 2023, November. Fortifying Information Security: Security Implications of Microservice and Monolithic Architectures. In 2023 16th International Conference on Security of Information and Networks (SIN) (pp. 1-6). IEEE.

[16] Dinesh Kumar, K. and Umamaheswari, E., 2024. An efficient proactive VM consolidation technique with improved LSTM network in a cloud environment. Computing, 106(1), pp.1-28.

[17] https://www.nbcnews.com/tech/internet/drought-stricken-communities-push-back-against-data-centers-n1271344

[18] https://datareportal.com/reports/digital-2023-india

[19] Nadaf, A., 2023, May. Green Computing: Optimized Underutilized Host detection in IQR Vm Allocation Policy. In 2023 4th International Conference on Intelligent Engineering and Management (ICIEM) (pp. 1-4). IEEE.

[20] Saha, S. and Beena, M., 2022, December. Power Cognizant Optimization Techniques for Green Cloud Systems. In 2022 OITS International Conference on Information Technology (OCIT) (pp. 507-512). IEEE

[21] Tiwari, S. and Beena, B.M., 2022, November. Energy Cognizant Scheduling in Three-Layer Cloud Architecture: A Systematic Review. In 2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) (pp. 1-6). IEEE.