Sarcouncil Journal of Engineering and Computer Sciences

ISSN(Online): 2945-3585

Volume- 05| Issue- 04| 2026

Research Article **Received:** 15-03-2026 | **Accepted:** 10-04-2026 | **Published:** 25-04-2026

Energy-Efficient Kubernetes Workloads in Multi-Region AWS: A Sustainable Design Guide

Veeresh Nunavath

Independent Researcher, University of Southern Indiana & Indiana.

Abstract: The rapid increase in the use of cloud-native applications has increased the environmental footprint of distributed computing, especially in multi-region Kubernetes deployments running on platforms such as Amazon Web Services (AWS). This paper gives a detailed description of how to design and run energy-efficient Kubernetes workloads across geographically separated AWS regions. It considers energy-conscious scheduling, efficiency-optimized autoscaling, telemetry-enabled observability, sustainable software design, and policy-based governance through applying concepts of sustainable computing in concert with container-based orchestration techniques. The article focuses on matching real-time use of resources with real-time environmental conditions (including carbon intensity, as well as regional energy mix) without sacrificing service reliability or compliance. In an integration of new studies and best practices, the research sets the stage for how to approach the process of scalable, resilient, and environmentally responsible Kubernetes operations.

Keywords: Energy-efficient computing; Kubernetes; Multi-region cloud architecture; AWS sustainability; Green software design.

INTRODUCTION

The increased usage of cloud computing has revolutionized how a modern enterprise can be organized; it provides scale, latitude, and availability to the world never been seen before. Nevertheless, such a transformation to digitalization has also contributed to a powerful rise in the worldwide energy consumption of data centers. Containerized workloads orchestrated through Kubernetes can be resource-demanding and ecologically unsound, at least on a cloud infrastructure. It is unknown how Kubernetes workloads are deployed in more than one AWS region to provide availability, performance, and regulatory compliance, and as more enterprises adopt these deployments, their environmental impact has become a consideration. This has created the necessity of developing sustainable architectural paradigms that factor in energy efficiency in the multi-region Kubernetes deployment plans at the ground level [Pinto, G. *et al*., 2014; Qi, H. *et al*., 2022; Siddik, M. A. B. *et al*., 2021]. Energy efficiency in cloud-native systems is no longer a nice-to-have optimization: it is a long-term strategic necessity. Regulatory frameworks, sustainability targets, and the growing climate change awareness have contributed to the growth in demand for greener IT solutions. Amazon, or AWS, providers have been taking steps to integrate green energy sources into their data centers. Nevertheless, it is also upon organizations to design and use their workloads in a manner that reduces energy consumption. Kubernetes, or other container orchestrators, introduce additional layers of abstraction and dynamic resource management that, once optimized, may result in significant power savings when workloads are distributed across multiple geographies [Castillejo-Cuberos, A. *et al*., 2023; Vaičys, J. *et al*., 2024; Gill, S. S., & Buyya, R. 2018].

The current paper highlights the design of energy-efficient Kubernetes workloads on AWS, referring to multi-region deployments. The recommendations range from the literature on the selection of infrastructure, the design of containers, the scheduling of workloads, autoscaling, and monitoring. We discuss effective methods that can help decrease energy consumption, optimize resource expenditures, and calibrate cloud activities to the goals of sustainability. First, we give an introduction to the concept of multi-region architecture in Kubernetes, then discuss the principles of energy optimization, paving the way for a sustainability-driven design of workloads that can follow the global environmental and economic principles [Dong, Z. *et al*., 2014; Jezdović, I. *et al*., 2021].

Multi-Region Kubernetes Architecture in AWS

In order to provide a context for sustainable workload design, it is important to build a first insight into the interaction of Kubernetes in the AWS regions, as illustrated in Figure 1. The Multi-Region Kubernetes deployment is the one that implies managing containerized applications across the clusters in the geographically distributed AWS data centers. This architecture provides redundancy, low-latency access, disaster recovery,

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

***Corresponding Author:** Veeresh Nunavath

DOI- https://doi.org/10.5281/zenodo.22848083

and data residency laws. Various utilities are offered by AWS to manage this architecture, such as Amazon Elastic Kubernetes Service (EKS), Amazon Route 53 for DNS-based load balancing, AWS Transit Gateway, and inter-region VPC peering [Böhm, S., & Wirtz, G. 2022; Alharthi, S. *et al*., 2024]. The trade-off with multi-region configurations is, however, the cost in terms of energy and resources in ensuring high availability. Continuous or consistent replication of data to different regions, always-on failovers, and regional redundancy may add compute and network consumption. All such operations that are not designed well end up as a source of unnecessary energy consumption. It is crucial, therefore, that functional efficiency and sustainable environmental concerns be incorporated in facilitating the assessment of the architecture [Sannidhanam, A. H. 2025; Lei, S. *et al*., 2022]. In moving past infrastructure design into energy-efficient operating environments, one quickly finds that sustainable Kubernetes workload strategies will not only consider how applications are deployed but also where and when. The above concerns bring us to the seriousness of energy-aware workload placement and scheduling in a multi-region setting.

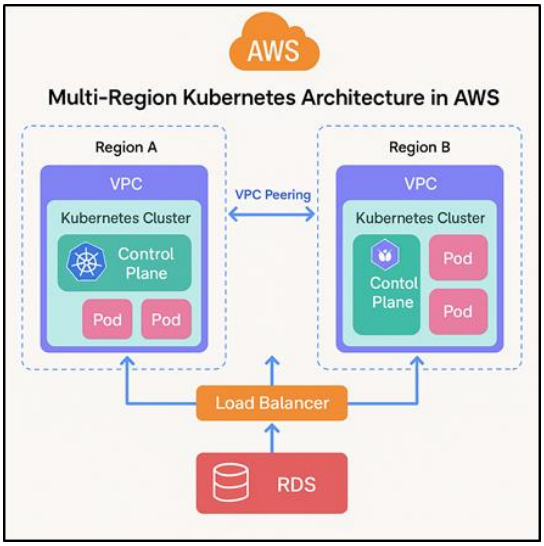

Figure 1: Multi-Region Kubernetes Architecture in AWS showcasing two regional VPCs, each hosting a Kubernetes cluster with control planes and pods. The regions are connected via VPC Peering, with a centralized load balancer distributing traffic and a shared RDS instance ensuring consistent data access across both regions.

Energy-Aware Scheduling and Placement of Kubernetes Workloads

Placement and scheduling are core to deciding how efficient a Kubernetes workload will be. Conventional scheduling algorithms are performance-centered and resource-based. Notably, energy-aware scheduling adds further parameters to the equation, including real-time carbon intensity of electrical grids, the availability of renewable energy in certain locations, and the energy efficiency of hardware underneath. AWS data centers are not homogeneous in terms of their power mix, and can also make use of reducing the total amount of emissions by running workloads in less carbon-intensive regions [Xu, M., & Buyya, R. 2020; Read, N., & Cosgrove, P. 2021]. Kubernetes can be expanded with custom schedulers, or Kubernetes APIs can be used to integrate with external data sources to achieve energy-aware scheduling. As an example, external environmental APIs and real-time energy metrics delivered by AWS could be employed to find the most appropriate region in which to execute a task. Also, using node affinity rules together with taints/tolerations and cloud carbon footprint can be used to steer compute-intensive workloads to run in renewable-powered regions, or during off-peak times when energy is more sustainable [Buyya, R. *et al*., 2024; Barney, A. *et al*., 2022].

Besides, dynamic workload placement, or workload migration among regions on the basis of energy measures, may result in considerable savings. All these will, however, be offset by the energy cost of transporting the data over regions. Therefore, the data gravity and the limit of latency should be investigated with a focus on energy optimization. This co-interaction of compute placement, data locality, and power consumption poses a challenging optimization problem that

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher

needs to be solved at both Kubernetes and cloud architecture levels [Anh, N. H. 2024; Xu, S. *et al*., 2021]. Developing these scheduling approaches further, one can note that advanced autoscaling and resource allocation schemes should also be integrated into sustainable Kubernetes design. The methods allow determining the demand that resources are allocated to, and therefore minimize wastage and over-allocation.

While we've established the significance of energy-aware workload placement, a data-driven understanding of regional carbon intensity helps inform more precise decisions in Kubernetes scheduling. The following table presents a comparative overview of estimated carbon intensities across key AWS regions, highlighting their relative suitability for sustainable deployment.

Table 1: Carbon Intensity Comparison of Selected AWS Regions

AWS Region	**Location**	**Grid Carbon Intensity (gCO_2/kWh)**	**Renewable Energy Share (%)**	**Sustainability Rating (1-5)**
us-west-1	N. California	240	55	4
eu-central-1	Frankfurt	310	42	3
ap-south-1	Mumbai	700	12	1
eu-north-1	Stockholm	45	98	5
us-east-1	N. Virginia	320	40	3
ap-northeast-1	Tokyo	430	25	2
sa-east-1	São Paulo	210	68	4

Autoscaling and Resource Optimization for Sustainability

Autoscaling is a cornerstone of energy-efficient cloud operations. Kubernetes supports Horizontal Pod Autoscaler (HPA), Vertical Pod Autoscaler (VPA), and Cluster Autoscaler (CA), all of which can dynamically scale workloads based on real-time demand. However, energy-aware autoscaling goes a step further by incorporating environmental factors into scaling decisions. For instance, workloads can be scaled down more aggressively during high-carbon periods or scaled up in regions currently powered by renewable energy surpluses [Drivas, I. C. *et al*., 2019; He, Z. *et al*., 2022].

Inappropriately defined resource requests and limits can cause major energy inefficiencies. Over-provisioning resources (allocating more CPU and memory than is necessary) means idle power consumption and reduced use of resources. In turn, limited under-provisioning can contribute to design autoscaling overheads. Proper profiling- and telemetry-based right-sizing is also essential to enable the best use of energy. This involves scrutinizing the CPU throttling patterns, memory saturation points, and container lifecycle patterns to adjust the settings [Saboor, A. *et al*., 2022; Thein, T. *et al*., 2020]. Also, cluster autoscaling options can favor the use of spot instances or ARM-based processors that, in most cases, do not require the amount of power that is required by similar x86 counterparts. The synergy will be provided upon having spot instances in regions with the dominance of renewable power, generating savings in price and sustainability. When appropriately integrated into the system, this set of methods forms a robust backbone of the next stage of energy-efficient processes, sustainable observability, and monitoring.

Beyond strategic scheduling, operational efficiencies through tuning Kubernetes autoscaling and resource allocation mechanisms can drastically reduce energy use. The following table summarizes empirically observed energy savings from various optimization techniques.

Table 2: Energy-Saving Potential of Optimization Strategies in Kubernetes (Empirical Estimates)

Optimization Strategy	**Description**	**Estimated Energy Savings (%)**
Aggressive Downscaling during Idle Hours	Reducing pod count when load is low	15-25%
Use of ARM-based Nodes	Switching from x86 to ARM architecture	10-20%
Spot Instance Allocation in Green Zones	Running workloads in spot markets in low-carbon regions	20-30%
Precise Resource Requests &	Avoiding overprovisioning of CPU/memory	5-10%

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher

Limits		
Batch Job Scheduling During Low Carbon	Timing non-critical jobs based on renewable availability	10-18%
Container Image Size Optimization	Reducing build and deployment overheads	3-7%

Sustainable Observability and Monitoring Practices

The key to ensuring that a Kubernetes environment is energy efficient boils down to observability. It follows that traditional observability stacks (e.g., Prometheus, Grafana, Loki) are performance-, reliability-, and security-oriented, and in the context of sustainability observability, should include energy consumption, carbon intensity, and resource utilisation efficiency metrics as shown in Figure 2. By monitoring these indicators, teams are able to interrelate workload behavior and energy consumption and locate inefficiencies in the application lifecycle [Bicer, Y. *et al*., 2022; García-Mireles, G. A. *et al*., 2026].

Energy-focused observability would involve the incorporation of third-party tools and APIs, e.g., those that provide carbon emissions data per AWS region. The customer carbon footprint tracking tool that AWS currently offers, supplemented with publicly available datasets on this electricity grid carbon intensity, can provide a real-time visualization of the impact on the environment. Such telemetry might be utilized to dynamically change workload behaviour, such as delaying non-essential batch jobs during times when the intensity of carbon use is high or moving stateless services to less carbon-intensive regions [Mathew, V. *et al*., 2012; Allam, H. 2025]. Elaborate telemetry can also expose workload design inefficiencies that are under the hood, like containers having poor ratios of CPU to memory they consume, resulting in inefficient utilization of the underlying hardware. Dashboards focused on presenting energy-related KPIs (kilowatt-hours per pod-hour, carbon grams per request, and so on) have become the must-have tool of developers and operators who wish to achieve energy efficiency targets. Such insights can then be used to provide optimization loops to autoscaling, scheduling, and container lifecycle management [Zhou, L. *et al*., 2023; Si, J. *et al*., 2019].

These monitoring practices, when aligned with automation and DevOps processes, pave the way for energy-aware Continuous Integration/Continuous Delivery (CI/CD) pipelines. By enforcing sustainability gates in deployment workflows, organizations can ensure that energy-intensive changes are flagged and optimized before going live. Transitioning from observability, the next logical step is understanding how application design and container configuration influence overall sustainability in Kubernetes environments.

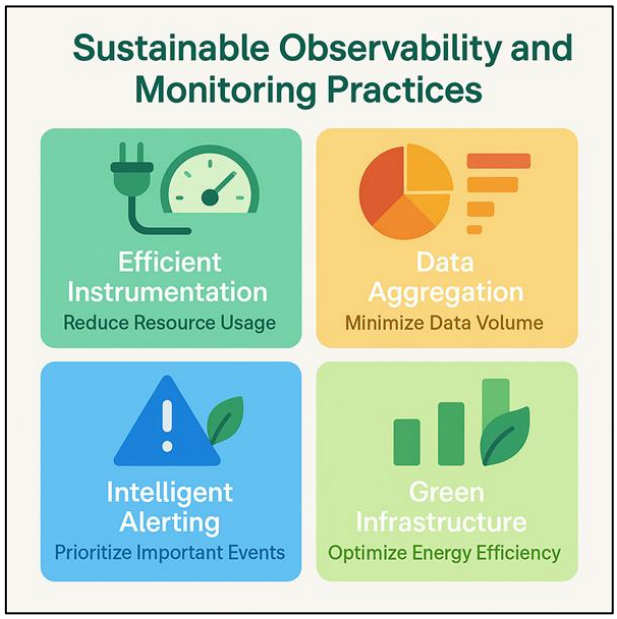

Figure 2: Sustainable Observability and Monitoring Practices illustrated through four key pillars: Efficient Instrumentation for reduced resource usage, Data Aggregation to minimize data volume, Intelligent Alerting for prioritizing critical events, and Green Infrastructure to enhance energy efficiency in monitoring systems.

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher

Green Application Design and Container Optimization

It is important because the sustainability of Kubernetes workloads is largely affected by the way that they themselves are architected. Software-layer energy efficiency starts with lean and write-performant application code. The resource-intensive monolithic applications produce bloated containers, resulting in excessive idle consumption. Conversely, microservices that have well-defined interfaces, low and loosely coupled dependencies, and asynchronous communication patterns are likely to be more efficient in a distributed setting [Meneguette, R. I., & Boukerche, A. 2019; Kaiser, S. *et al*., 2022]. Another important field to optimize is container images. Shrinking the image by minimizing dependencies, using lighter base images, as well as removing redundant layers, results in faster boot time, fewer bytes to be transferred over the network, and less storage I/O, which translates to lower energy consumption. Bi-stage construction and caching mechanisms can be used to facilitate container manufacturing and delivery [Lei, S. *et al*., 2022; Drivas, I. C. *et al*., 2019].

Moreover, applications should be energy-aware in their operation and interpretation. This involves the ability to gracefully degrade in high-carbon times, the ability to idle or enter low-power states, and the ability to dynamically configure service behavior based on telemetry using APIs or feature toggles. Stateless designs make migration of workload across greener territories sustainable, and also easy to implement with respect to the lack of consistency in states, and increase availability and sustainability. A combination of both these forms of application-level and infrastructure-level would enable an overall sustainability posture. However, the full power of these practices will become achievable only when intertwined with the DevOps and GitOps workflows, so that energy efficiency will constitute the default attribute of the delivery process.

Integrating Sustainability into DevOps and GitOps Pipelines

Modern Kubernetes operations are increasingly driven by automation pipelines that manage code deployments, infrastructure provisioning, and configuration management. Embedding sustainability into these pipelines ensures that energy efficiency is not an afterthought but a continuous part of the development lifecycle. In CI/CD, this can be accomplished by adding quality gates for container size, build efficiency, and energy benchmarking results. Workloads failing to meet sustainability thresholds, e.g., those with high carbon-per-build ratios, can be flagged for review or rejected. These gates can be enforced using tools that assess code complexity, dependency bloat, and compute overhead [He, Z. *et al*., 2022; Thein, T. *et al*., 2020].

GitOps, which involves managing Kubernetes states declaratively via Git repositories, allows for greater control over workload placement and scaling rules. By integrating sustainability policies into GitOps manifests, such as specifying preferred green regions or sustainable instance types, organizations can maintain energy-efficient configurations at scale. Audit trails provided by Git repositories also facilitate tracking sustainability improvements over time [García-Mireles, G. A. *et al*., 2026; Allam, H. 2025]. Automated workflows can also be extended to schedule non-urgent tasks based on carbon-aware cron jobs. For example, data processing workloads can be queued to run when the grid is less carbon-intensive or when renewable energy is abundant. This dynamic scheduling reinforces a culture of environmentally responsible computing without sacrificing business outcomes.

Having established how automation enhances sustainability, we now examine the broader implications of compliance, governance, and regulatory alignment in the context of energy-efficient Kubernetes workloads.

Compliance, Governance, and Regulatory Considerations

The shift towards sustainability in cloud-native architecture is reinforced by evolving regulatory landscapes and corporate governance frameworks. Organizations operating Kubernetes workloads across AWS regions must ensure compliance with both environmental and data governance laws. This is especially critical in regions governed by the EU Green Deal, California's Climate Disclosure Regulations, and ESG (Environmental, Social, and Governance) reporting mandates [Siddik, M. A. B. *et al*., 2021; Read, N., & Cosgrove, P. 2021].

Design policies: sustainable design choices, like choosing green AWS regions, minimizing inter-region data transfers, or using zones powered by renewable energy, can directly help support compliance goals. The regional sustainability data is published by AWS itself and can be followed up

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher

on in internal audits and external reporting to justify design decisions already. Sustainability standards can be enforced across clusters by using policy-as-code through implementing Kubernetes Admission Controllers or OPA (Open Policy Agent) [Sannidhanam, A. H. 2025; Xu, M., & Buyya, R. 2020]. Reporting and traceability of sustainability metrics should also be a part of governance frameworks. It is possible to aggregate data on energy use per application, carbon footprint per release, and regional consumption statistics that can be compiled into leadership review sustainability dashboards. They are in support of ESG disclosures, assist in meeting carbon neutrality targets, and exhibit climate-related targ*et al*ignment. More importantly, there should be a balance between energy-efficient governance and service-level goals. Although energy reduction is one of the primary objectives, availability, latency, and customer experience come first. That is why compliance approaches must be devised to enable flexibility in workload dynamics without reducing regulatory compliance. What will be done next is only subject to our imagination, as the future is full of possibilities for what further innovations can be made in sustainable Kubernetes workloads in cloud-native environments.

FUTURE OUTLOOK AND INNOVATIONS IN SUSTAINABLE CLOUD-NATIVE WORKLOADS

The future of energy-efficient Kubernetes workloads lies at the intersection of cloud innovation, machine learning, and environmental intelligence. Emerging technologies promise to make energy optimization even more granular and automated. For example, AI-driven schedulers are being developed to analyze historical energy data, predict carbon intensity trends, and place workloads accordingly. These predictive models can dynamically optimize cluster configurations in real time [Si, J. *et al*., 2019; Meneguette, R. I., & Boukerche, A. 2019].

Another innovation that is set to reduce idling compute time is serverless Kubernetes or event-driven architecture. Serverless paradigms can significantly reduce energy consumption by executing functions only when requested and automatically freeing resources. Likewise, development of green computing hardware, including ARM-based processors and energy-optimized GPUs, is also being more widely utilized by cloud providers and is being offered to users in AWS [Kaiser, S. *et al*., 2022]. Edge computing is also bringing its part into sustainability as it facilitates cutting down on transfers between regions. Bringing lightweight workloads near the user reduces the latency and the network power consumption. Together with energy-dispersing edge nodes and solar micro-data centers, this can redraw the sustainability solar panel of distributed applications. Further, there is industry-wide action to standardize sustainability measurements and practices as represented by the Green Software Foundation. Such frameworks will probably form part of future cloud certification and audit. Kubernetes itself is similarly developing to bear native energy-conscious capabilities by continuing its contributions to the open-source environment. These new developments are promising, but their performance is also dependent on the strength of organizational devotion to the practice of sustainability and continuous interaction between the developers, cloud planning specialists, and climate analysts. Companies able to code green and resilient cloud infrastructure into the DNA of their operations will be at the forefront of making the digital infrastructure more responsible and green.

CONCLUSION

A multi-region AWS environment with energy-efficient Kubernetes workloads is a complex yet essential step that organisations must take to ensure that they minimise their impact on the environment without compromising the resilience and performance of their applications. The technical details of such care, such as energy-aware scheduling and observability, as well as optimized application design and governance, can all be found in this guide as ways to combine technical rigor with ecological responsibility. The workload toward sustainable cloud-native architecture is with deliberate overdesign- to pick sustainable areas, proportionate workloads, automatic environment-friendly deployments, and assign constant environmental monitoring metrics. With cloud infrastructure growing at a global scale, making sustainability the core of Kubernetes operations will be essential in guaranteeing long-term sustainability, regulatory compliance, and business competitiveness. The opportunities are great, and with innovation, working together, and dedication, energy-efficient Kubernetes systems can become the norm and not the exception, a new day imbued with sustainable cloud computing.

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher

REFERENCES

1. Pinto, G., Castor, F., & Liu, Y. D. "Mining questions about software energy consumption." *Proceedings of the 11th working conference on mining software repositories*. (2014).
2. Qi, H., Guo, Y., Hou, D., Xing, Z., Ren, W., Cong, L., & Di, X. "SDN-based dynamic multi-path routing strategy for satellite networks." *Future Generation Computer Systems* 133 (2022): 254-265.
3. Siddik, M. A. B., Shehabi, A., & Marston, L. "The environmental footprint of data centers in the United States." *Environmental Research Letters* 16.6 (2021): 064017.
4. Castillejo-Cuberos, A., Cardemil, J. M., & Escobar, R. "Techno-economic assessment of photovoltaic plants considering high temporal resolution and non-linear dynamics of battery storage." *Applied Energy* 334 (2023): 120712.
5. Vaičys, J., Gudžius, S., Jonaitis, A., Račkienė, R., Blinov, A., & Peftitsis, D. "A case study of optimising energy storage dispatch: Convex optimisation approach with degradation considerations." *Journal of Energy Storage* 97 (2024): 112941.
6. Gill, S. S., & Buyya, R. "A taxonomy and future directions for sustainable cloud computing: 360 degree view." *ACM Computing Surveys (CSUR)* 51.5 (2018): 1-33.
7. Dong, Z., Zhuang, W., & Rojas-Cessa, R. "Energy-aware scheduling schemes for cloud data centers on google trace data." *2014 IEEE Online Conference on Green Communications (OnlineGreencomm)*. IEEE, (2014).
8. Jezdović, I., Popović, S., Radenković, M., Labus, A., & Bogdanović, Z. "A crowdsensing platform for real-time monitoring and analysis of noise pollution in smart cities." *Sustainable Computing: Informatics and Systems* 31 (2021): 100588.
9. Böhm, S., & Wirtz, G. "Cloud-edge orchestration for smart cities: A review of kubernetes-based orchestration architectures." *EAI Endorsed Trans. Smart Cities* 6.18 (2022): e2.
10. Alharthi, S., Alshamsi, A., Alseiari, A., & Alwarafy, A. "Auto-scaling techniques in cloud computing: Issues and research directions." *Sensors* 24.17 (2024): 5551.
11. Sannidhanam, A. H. "Autonomous AI Agents for Cloud Infrastructure Operations." *Journal of Contemporary Science and Technology Management* 1.01 (2025): 83-100.
12. Lei, S., Pozo, D., Wang, M. H., Li, Q., Li, Y., & Peng, C. "Power economic dispatch against extreme weather conditions: The price of resilience." *Renewable and Sustainable Energy Reviews* 157 (2022): 111994.
13. Xu, M., & Buyya, R. "Managing renewable energy and carbon footprint in multi-cloud computing environments." *Journal of Parallel and Distributed Computing* 135 (2020): 191-202.
14. Read, N., & Cosgrove, P. "Comments on:“Carbon emissions and costs associated with subsidizing New York nuclear instead of replacing it with renewables”." *Journal of Cleaner Production* 290 (2021): 125835.
15. Buyya, R., Ilager, S., & Arroba, P. "Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads." *Software: Practice and Experience* 54.1 (2024): 24-38.
16. Barney, A., Polatidis, H., & Haralambopoulos, D. "Decarbonisation of islands: A multi-criteria decision analysis platform and application." *Sustainable Energy Technologies and Assessments* 52 (2022): 102115.
17. Anh, N. H. "Hybrid cloud migration strategies: balancing flexibility, security, and cost in a multi-cloud environment." *Transactions on Machine Learning, Artificial Intelligence, and Advanced Intelligent Systems* 14.10 (2024): 14-26.
18. Xu, S., Koren, I., & Krishna, C. M. "Adaptive workload adjustment for cyber-physical systems using deep reinforcement learning." *Sustainable Computing: Informatics and Systems* 30 (2021): 100525.
19. Drivas, I. C., Sakas, D. P., Giannakopoulos, G. A., & Kyriaki-Manessi, D. "Search engines’ visits and users’ behavior in websites: optimization of users engagement with the content." *International Conference on Business Intelligence & Modelling*. Cham: Springer International Publishing, (2019).
20. He, Z., Ning, D., Gou, Y., & Zhou, Z. "Wave energy converter optimization based on differential evolution algorithm." *Energy* 246 (2022): 123433.
21. Saboor, A., Hassan, M. F., Akbar, R., Shah, S. N. M., Hassan, F., Magsi, S. A., & Siddiqui, M. A. "Containerized microservices orchestration and provisioning in cloud computing: A conceptual framework and

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher

future perspectives." *Applied Sciences* 12.12 (2022): 5793.
22. Thein, T., Myo, M. M., Parvin, S., & Gawanmeh, A. "Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers." *Journal of King Saud University Computer and Information Sciences* 32.10 (2020): 1127-1139.
23. Bicer, Y., Sajid, M. U., & Al-Breiki, M. "Optimal spectra management for self-power producing greenhouses for hot arid climates." *Renewable and Sustainable Energy Reviews* 159 (2022): 112194.
24. García-Mireles, G. A., Moraga, M. Á., García, F., & Calero, C. "Sustainability in the Field of Software Engineering: A Tertiary Study." *ACM Transactions on Software Engineering and Methodology* 35.5 (2026): 1-86.
25. Mathew, V., Sitaraman, R. K., & Shenoy, P. "Energy-aware load balancing in content delivery networks." *2012 Proceedings IEEE INFOCOM*. IEEE, (2012).
26. Allam, H. "Policy-Driven Engineering: Automating Compliance Across DevOps Pipelines." *International Journal of Emerging Trends in Computer Science and Information Technology* 6.1 (2025): 89-100.
27. Zhou, L., Nandal, A., Ansari, I. S., Polat, K., Polak, L., Dhaka, A., & Alenezi, F. "Secrecy outage probability and limiting threshold criteria in an optimal SNR regime by maximizing desired transfer rate." *Internet of Things* 22 (2023): 100789.
28. Si, J., Rong, H., Cheng, Z., Li, Z., Cheng, J., & Zhong, C. "Impact of channel correlation on secrecy performance over Rayleigh fading channels." In *ICC 2019-2019 IEEE International Conference on Communications (ICC)* (2019). 1-6. IEEE.
29. Meneguette, R. I., & Boukerche, A. "An efficient green-aware architecture for virtual machine migration in sustainable vehicular clouds." *IEEE Transactions on Sustainable Computing* 5.1 (2019): 25-36.
30. Kaiser, S., Haq, M. S., Tosun, A. Ş., & Korkmaz, T. "Container technologies for arm architecture: A comprehensive survey of the state-of-the-art." *IEEE Access* 10 (2022): 84853-84881.

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:
Nunavath, V. "Energy-Efficient Kubernetes Workloads in Multi-Region AWS: A Sustainable Design Guide" *Sarcouncil Journal of Engineering and Computer Sciences* 5.4 (2026): pp 144-151.

Copyright © 2022 The Author(s): This work is licensed under a Creative Commons Attribution- NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License

Publisher: SARC Publisher