Elasticity in Cloud Computing

Elasticity in cloud computing refers to the ability of a system to dynamically scale resources up or down based on demand, ensuring optimal performance and cost efficiency (Armbrust et al., 2010).

This is achieved through automated resource allocation, allowing applications to handle varying workloads without manual intervention. For example, during peak traffic, additional virtual machines or containers can be provisioned, and scaled down during low demand to reduce costs.

Elasticity benefits applications by improving availability, as resources are adjusted to prevent overload, and enhancing fault tolerance, as systems can adapt to failures by redistributing resources. It also optimizes costs, as organizations pay only for resources used, making it ideal for unpredictable workloads (Mell & Grance, 2011).

Vendor Lock-In and Kubernetes

Vendor lock-in occurs when applications are tightly coupled to a specific cloud provider's proprietary services, making migration to other providers costly and complex (Vaquero et al., 2011).

This dependency arises from using provider-specific APIs, databases, or compute services, which are not portable. Kubernetes mitigates this by providing a standardized, open-source platform for container orchestration that operates consistently across public, private, and hybrid clouds (Burns et al., 2016).

By abstracting infrastructure details, Kubernetes allows applications to be deployed using portable container images and configurations, reducing reliance on vendor-specific services. This portability enables organizations to switch providers or adopt multi-cloud strategies with minimal rework, enhancing flexibility and cost control (Hoffmann et al., 2019).

Kubernetes on Private Infrastructure and Hybrid Cloud

Using Kubernetes on private infrastructure allows organizations to leverage existing on-premises resources while achieving cloud-like capabilities such as scalability and automated management (Bernstein, 2014).

Kubernetes clusters can orchestrate containers on private servers, optimizing resource utilization and enabling consistent application deployment. This is beneficial for sensitive workloads requiring compliance or low-latency access. Transitioning to a hybrid cloud is facilitated as Kubernetes provides a unified management layer across private and public clouds (Pahl, 2015).

Organizations can extend private clusters to public clouds using Kubernetes federation or service meshes, ensuring seamless workload portability and resource scaling. This hybrid approach balances cost, control, and scalability, allowing dynamic resource allocation while maintaining data sovereignty (Vaquero et al., 2011).

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50–58. https://doi.org/10.1145/1721654.1721672

Bernstein, D. (2014). Containers and cloud: From LXC to Docker to Kubernetes. IEEE Cloud Computing, 1(3), 81–84. https://doi.org/10.1109/MCC.2014.51

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. Communications of the ACM, 59(5), 50–57. https://doi.org/10.1145/2890784

Hoffmann, M., Lessner, T., & Lübken, J. (2019). A survey on cloud vendor lock-in mitigation strategies. Proceedings of the 9th International Conference on Cloud Computing and Services Science, 316–323. https://doi.org/10.5220/0007739603160323

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. National Institute of Standards and Technology, 53(6), 1–7. https://doi.org/10.6028/NIST.SP.800-145

Pahl, C. (2015). Containerization and the PaaS cloud. IEEE Cloud Computing, 2(3), 24–31. https://doi.org/10.1109/MCC.2015.51

Vaquero, L. M., Rodero-Merino, L., & Morán, D. (2011). Locking the sky: A survey on IaaS cloud lock-in. Computing, 91(1), 93–111. https://doi.org/10.1007/s00607-010-0140-9