

REFERENCES

- [1] “Stateful vs stateless — redhat.com,” <https://www.redhat.com/en/topics/cloud-native-apps/stateful-vs-stateless>.
- [2] H. Grgic, B. Mihaljević, and A. Radovan, “Comparison of garbage collectors in java programming language,” in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 1539–1544.
- [3] P. Pufek, H. Grgić, and B. Mihaljević, “Analysis of garbage collection algorithms and memory management in java,” in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2019, pp. 1677–1682.
- [4] T. Lindholm, F. Yellin, G. Bracha, and A. Buckley, *The Java virtual machine specification*. Pearson Education, 2014.
- [5] “Memory Management in Java - Javatpoint — javatpoint.com,” <https://www.javatpoint.com/memory-management-in-java>.
- [6] “Understanding Memory Management — docs.oracle.com,” https://docs.oracle.com/cd/E13150_01/jrocket_jvm/jrocket/geninfo/diagnos/garbage_collect.html.
- [7] “JIT in Java | Java JIT - Javatpoint — javatpoint.com,” <https://www.javatpoint.com/jit-in-java>.
- [8] S. Tavakolisomeh, R. Bruno, and P. Ferreira, “BestGC: An Automatic GC Selector,” *IEEE Access*, vol. 11, pp. 72 357–72 373, 2023.
- [9] Stefan Johansson. (2023) Jdk 21: The gcs keep getting better. Stefan Johansson’s Blog. [Online]. Available: <https://kstefanj.github.io/2023/12/13/jdk-21-the-gcs-keep-getting-better.html>
- [10] R. Lakshmanan. (2024) Cms gc algorithm removed from java 14? GC Easy - Universal Java GC Log Analyser. [Online]. Available: <https://blog.gceasy.io/cms-gc-algorithm-removed-from-java-14/>
- [11] D. H. Kurniawan, C. A. Stuardo, R. A. OS, and H. S. Gunawi, “Notification and prediction of heap management pauses in managed languages for latency stable systems,” *To appear*, n.d.
- [12] “Jep 404: Generational shenandoah (experimental),” <https://openjdk.org/jeps/404>.

- [13] Inside.Java. (2023) Introducing generational zgc. [Online]. Available: <https://inside.java/2023/11/28/gen-zgc-explainer/>
- [14] “JEP 318: Epsilon: A No-Op Garbage Collector (Experimental) — openjdk.java.net,” <https://openjdk.java.net/jeps/318>.
- [15] “Using JConsole - Java SE Monitoring and Management Guide — docs.oracle.com,” <https://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html>.
- [16] “VisualVM: Home — visualvm.github.io,” <https://visualvm.github.io/>.
- [17] Oracle. (n.d.) Jrockit to hotspot migration guide. Oracle Help Center. [Online]. Available: <https://docs.oracle.com/en/java/javase/11/jrockit-hotspot/logging.html>
- [18] F. Xian, W. Srisa-an, and H. Jiang, “Garbage collection: Java application servers’ achilles heel,” *Science of Computer Programming*, vol. 70, no. 2-3, pp. 89–110, 2008.
- [19] D. Fireman, R. Lopes, and J. A. B. Monteiro, “Using load shedding to fight tail-latency on runtime-based services,” in *Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, 2017, available via Google Scholar: https://scholar.google.com/scholar?as_q=Using+load+shedding+to+fight+tail-latency+on+runtime-based+services&as_occt=title&hl=en&as_sdt=_.
- [20] J. Zhao, “Improving performance of garbage collection for data-intensive applications in cloud systems,” Ph.D. dissertation, University of Colorado Colorado Springs, 2024.
- [21] S. M. Blackburn, P. Cheng, and K. S. McKinley, “Myths and realities: The performance impact of garbage collection,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 25–36, 2004.
- [22] A. O. Portillo-Dominguez, M. Wang, J. Murphy, and D. Magoni, “Adaptive gc-aware load balancing strategy for high-assurance java distributed systems,” in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*. IEEE, 2015, pp. 68–75.
- [23] A. O. Portillo-Dominguez, P. Perry, D. Magoni, M. Wang, and J. Murphy, “Trini: an adaptive load balancing strategy based on garbage collection for clustered java systems,” *Software: Practice and Experience*, vol. 46, no. 12, pp. 1705–1733, 2016.

- [24] S. Saraswati, S. Chatterjee, and R. Ramachandra, “Steal-a-gc: Framework to trigger gc during idle periods in distributed systems,” in *2016 IEEE 23rd International Conference on High Performance Computing (HiPC)*. IEEE, 2016, pp. 392–400.
- [25] N. Veretelnyk. (2021) Garbage collection: V8’s orinoco - nikolay veretelnyk - medium. [Online]. Available: <https://medium.com/@nikolay.veretelnik/garbage-collection-v8s-orinoco-452b70761f0c>
- [26] D. Fireman, J. Brunet, R. Lopes, D. Quaresma, and T. E. Pereira, “Improving tail latency of stateful cloud services via gc control and load shedding,” in *2018 IEEE International Conference on Cloud Computing Technology and Science (Cloud-Com)*, 2018, pp. 121–128.
- [27] Z. Zhuang, C. Tran, H. Ramachandra, and B. Sridharan, “Eliminating os-caused large jvm pauses for latency-sensitive java-based cloud platforms,” in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. IEEE, 2016, pp. 694–701.
- [28] “Kubernetes Documentation — kubernetes.io,” <https://kubernetes.io/docs/home/>.
- [29] “Docker overview — docs.docker.com,” <https://docs.docker.com/get-started/overview/>.
- [30] T. Hu and Y. Wang, “A kubernetes autoscaler based on pod replicas prediction,” in *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*. IEEE, 2021, pp. 238–241.
- [31] “Understand the OutOfMemoryError Exception — docs.oracle.com,” <https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/memleaks002.html>, [Accessed 10-12-2024].
- [32] “GitHub - kubernetes/client-go: Go client for Kubernetes. — github.com,” <https://github.com/kubernetes/client-go>.
- [33] “ReplicaSet — kubernetes.io,” <https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/>, [Accessed 12-12-2025].
- [34] C. Stenberg, “Frameworks for lifecycle management of stateful applications on top of kubernetes: Testing and evaluation,” Dissertation, 2022.
- [35] M. Sebrechts, T. Ramlot, S. Borny, T. Goethals, B. Volckaert, and F. De Turck, “Adapting kubernetes controllers to the edge: on-demand control planes using wasm and wasi,” in *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, 2022, pp. 195–202.

- [36] “Kubernetes Metrics (v1beta1) — kubernetes.io,” <https://kubernetes.io/docs/reference/external-api/metrics.v1beta1/>.
- [37] C. Carrión, “Kubernetes scheduling: Taxonomy, ongoing issues and challenges,” *ACM Comput. Surv.*, vol. 55, no. 7, Dec. 2022. [Online]. Available: <https://doi.org/10.1145/3539606>
- [38] L. Abdollahi Vayghan, M. A. Saied, M. Toeroe, and F. Khendek, “Deploying microservice based applications with kubernetes: Experiments and lessons learned,” in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018, pp. 970–973.
- [39] O. Mart, C. Negru, F. Pop, and A. Castiglione, “Observability in kubernetes cluster: Automatic anomalies detection using prometheus,” in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2020, pp. 565–570.
- [40] B. B. Rad, H. J. Bhatti, and M. Ahmadi, “An introduction to docker and analysis of its performance,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 17, no. 3, March 2017, manuscript received March 5, 2017, revised March 20, 2017.
- [41] “Configure Liveness, Readiness and Startup Probes — kubernetes.io,” <https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/>.
- [42] “Design of Shutdown Hooks API — docs.oracle.com,” <https://docs.oracle.com/javase/8/docs/technotes/guides/lang/hook-design.html>.
- [43] “kube-apiserver — kubernetes.io,” <https://kubernetes.io/docs/reference/command-line-tools-reference/kube-apiserver/>.
- [44] A. Muppada, “A Hands-On Guide to Kubernetes Custom Resource Definitions (CRDs) With a Practical Example — muppadaanvesh,” <https://medium.com/@muppadaanvesh/a-hand-on-guide-to-kubernetes-custom-resource-definitions-crds-with-a-practical-example-%EF%B8%8F-84094861e90b>.
- [45] M. Devops, “CRDs in Kubernetes — minimaldevops.com,” <https://minimaldevops.com/crds-in-kubernetes-c38037315548>.

- [46] “Extend the Kubernetes API with CustomResourceDefinitions — kubernetes.io,” <https://kubernetes.io/docs/tasks/extend-kubernetes/custom-resources/custom-resource-definitions/>.
- [47] “GitHub - anveshmuppeda/kubernetes: Kubernetes Complete Hands-On Guides — github.com,” <https://github.com/anveshmuppeda/kubernetes>.
- [48] “Operating etcd clusters for Kubernetes — kubernetes.io,” <https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/>.
- [49] “Installation | Rancher Desktop Docs — docs.rancherdesktop.io,” <https://docs.rancherdesktop.io/getting-started/installation/>.
- [50] “Working with Containers | Rancher Desktop Docs — docs.rancherdesktop.io,” <https://docs.rancherdesktop.io/tutorials/working-with-containers>.
- [51] “GitHub - kubernetes-sigs/metrics-server: Scalable and efficient source of container resource metrics for Kubernetes built-in autoscaling pipelines. — github.com,” <https://github.com/kubernetes-sigs/metrics-server>.
- [52] “pkg package - sigs.k8s.io/controller-runtime/pkg - Go Packages — pkg.go.dev,” <https://pkg.go.dev/sigs.k8s.io/controller-runtime/pkg>.
- [53] “Releases · kubernetes-sigs/kubebuilder — github.com,” <https://github.com/kubernetes-sigs/kubebuilder/releases>.
- [54] “- YouTube — youtube.com,” <https://www.youtube.com/watch?v=rfXk6svglrA>.
- [55] “Using RBAC Authorization — kubernetes.io,” <https://kubernetes.io/docs/reference/access-authn-authz/rbac/>.
- [56] G. Rostami, “Role-based access control (rbac) authorization in kubernetes,” *Journal of ICT Standardization*, vol. 11, no. 3, pp. 237–260, 2023.
- [57] “Introducing Kubebuilder: an SDK for building Kubernetes APIs using CRDs — kubernetes.io,” <https://kubernetes.io/blog/2018/08/10/introducing-kubebuilder-an-sdk-for-building-kubernetes-apis-using-crds/>.
- [58] “Containerizing Test Tooling: Creating your Dockerfile and Makefile — docker.com,” <https://www.docker.com/blog/containerizing-test-tooling-creating-your-dockerfile-and-makefile/>.
- [59] K. Aykurt, R.-M. Ursu, J. Zerwas, P. Krämer, N. Asadi, L. Wong, and W. Kellerer, “Hypa: Hybrid horizontal pod autoscaling with automated model updates,” in *2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2023, pp. 8–14.

- [60] “Horizontal Pod Autoscaling — kubernetes.io,” <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>.
- [61] V. Tiwari, S. Upadhyay, J. K. Goswami, and S. Agrawal, “Analytical evaluation of web performance testing tools: Apache jmeter and soapui,” in *2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT)*, 2023, pp. 519–523.
- [62] Q. Cooper, D. Krishnamurthy, and Y. Amannejad, “Budget aware performance test selection for microservices,” in *2024 IEEE 17th International Conference on Cloud Computing (CLOUD)*, 2024, pp. 376–385.