

NINE · CLOUD NAVIGATORS

Day 2 Operations with Kubernetes

What to consider when running Kubernetes in production

Table of Contents

1. Kubernetes: The De-Facto Standard for Container Orchestration

2. Adopting Kubernetes in Your Business

3. Containers and Kubernetes: Increasing Velocity

4. Kubernetes in Business: A Valuable Technology for Companies

5. The Kubernetes Application Lifecycle

6. Implementation: What Companies Need to Consider

7. Conclusion

8. About Nine Internet Solutions AG

Kubernetes: The De-Facto Standard for Container Orchestration

In just over a decade, Kubernetes has grown from a Google-internal infrastructure project into the world-wide de-facto standard for efficiently providing, scaling, and managing containerised workloads. According to the **CNCF Annual Survey 2024**, 80% of organisations now use Kubernetes in production, up from 66% the previous year.¹ Engineering teams across industries rely on it to ship features faster, maintain high availability, and run cost-efficient infrastructure at any scale.

Successful Implementation of Day-2 Operations

When introducing Kubernetes to their own IT systems, many companies are soon faced with a major challenge: day-2 operations. This is the point where the platform is ready for use and is integrated into daily business. Maintenance, monitoring, upgrades, and optimisation tasks arise, and from this moment onwards they require continuous resources and expertise.

This is why businesses should make resources and internal know-how available early on to ensure that the transition to running Kubernetes goes as smoothly as possible. Day-2 operations do not end with a successful deployment, they last for the entire product lifecycle. Only continuous system analysis, upgrades, and patching create efficiency when running applications in Kubernetes.

This whitepaper presents practical guidance on the processes you can expect when running Kubernetes, what you need to consider, and how you can manage your platform sustainably with the right tools and partners.

¹ Cloud Native Computing Foundation (CNCF) Annual Survey 2024. 750 respondents.

Adopting Kubernetes in Your Business

Businesses are actively looking for options that allow them to optimise their software architecture and future-proof it at the same time. Integrating containerised applications is one such option. Kubernetes is frequently used as a framework to manage applications in a standardised manner. Although the open-source system has revolutionised the IT world, its real-life implementation is often quite complex.

Running Kubernetes requires a new working mode, as familiar workflows have to be disrupted and adapted.

Aside from process-related change, a change in the workforce is needed as well: the usefulness and goals of the new working methods must be transparent and understood across all departments to enable a successful transition.

Actively Implement the Change Process and Think into the Future

Many tasks necessary in Kubernetes-related workflows are not covered by Kubernetes itself. Businesses need to be aware that further tools are needed and that these also need to be met with acceptance and understanding among the workforce. When all factors are considered right from the beginning, an overly long change process can be avoided and the full potential offered by Kubernetes can be harnessed sustainably.

The challenges a company may face

- Finding qualified personnel who are already experienced in implementing and operating Kubernetes at scale
- Developing an exact plan concerning the tasks related to day-2 operations
- Creating and maintaining an overview of the timeframe and costs involved
- Keeping pace with the rapid Kubernetes release cycle (three minor releases per year)

Containers and Kubernetes: Increasing Velocity

Instead of focusing on their core competencies and engaging in value creation, companies often spend much time ensuring that their applications simply run. In software development, fast processes play an important role: to be ahead of the competition, provide users with a first release, or to receive feedback regarding bug fixes early on. Kubernetes enables a largely automated process, as long as there is a managed cluster and the application is configured correctly. This means that cost is kept to a minimum, **iterations are shorter, and time to market is significantly reduced.**

Containers: Strong Software Packages

Due to the ever more complex nature of applications, the responsibilities of IT departments are increasingly growing, as is the complexity of application architecture, frameworks, and environments. Containers can help, since developers can define the software runtime environment, which is abstracted from the underlying infrastructure.

When providing containers, the applications and all related components are packaged together. The result: applications are less dependent on their environment, and can be run at any time and on different hosting systems. In order to benefit from these advantages, what is needed is container orchestration, and thus the use of a system like Kubernetes.

The advantages of containerised applications

- More efficient workflow and shorter feedback cycles
- Lower operational overhead compared to traditional VMs
- Less storage used and reduced resource requirements through higher application density
- Testing and production environments are identical: no more "works on my machine"
- Portable across different infrastructure providers and cloud environments

The OCI Standard: Images That Run Everywhere

Container images today follow the Open Container Initiative (OCI) standard, which means images built with Docker, Podman, or any other OCI-compliant tool run without modification across all modern Kubernetes environments. Since Kubernetes v1.24, the default container runtime is **containerd**, a lightweight, production-grade runtime that powers over 95% of all Kubernetes clusters.² The Docker runtime layer (dockershim) has been fully removed from Kubernetes, though existing Docker images continue to work without any changes.

² CNCF Annual Survey 2024.

Kubernetes: A Short Explainer

The open-source system Kubernetes allows for the management and orchestration of containerised workloads and services. It coordinates not only compute and network infrastructure, but also storage for user workloads, providing a container-centric management environment. Its strengths become particularly apparent when it comes to automating processes: it works as an ecosystem acting as a central access point for components and tools.

This allows for easier roll-out, scaling, and management of applications. With scalable architecture, containers can be orchestrated across several machines – anytime and anywhere – while Kubernetes handles scheduling, self-healing, and load distribution automatically.

Kubernetes in Business: A Valuable Technology for Companies

Due to the increasing prominence of Kubernetes worldwide, the open-source system can be found in the IT departments of businesses of any size. According to the CNCF Annual Survey 2024, **80% of organisations now run Kubernetes in production**. Companies benefit from the flexibility of being able to adapt and customise Kubernetes, using this to drive innovative projects and to tackle the growing challenges of monitoring, scaling, and communication.

What Are the Advantages of Using Kubernetes?

A shorter time to market

Opting for Kubernetes and containers leads to homogeneous development, testing, and production environments. This actively supports deployment automation, so new releases can be published faster and with greater confidence.

Improved portability and no vendor lock-in

Applications in containers can be managed via Kubernetes independently of their environment. Kubernetes can deploy containers regardless of the underlying infrastructure provider, meaning applications can run in geographic proximity to customers – including in Switzerland, for data sovereignty requirements – and avoid vendor lock-in.

Lower costs through re-usability

One major added value of Kubernetes is that components and stacks can be re-used. Instead of building them from scratch every time, the team can rely on existing containers and configurations – often packaged as Helm charts – and adjust them accordingly. This saves time and money, and creates efficient workflows.

More stability and higher availability

Kubernetes supports companies in decreasing operational expenditure through automation. By automating processes that validate code quality, check Kubernetes configurations, and scan for container and code vulnerabilities, engineering teams gain clarity on the implications of any changes. A rich ecosystem of tools integrates with Kubernetes to optimise incident management through dashboards, metrics, log observability, and alerting platforms.

Horizontal scaling and load balancing

Using a container orchestration system enables higher resource utilisation, since workloads can be scaled and distributed horizontally across several servers. Automated scaling leads to optimal resource planning, while service availability can be maintained during traffic peaks.

De-coupled application and operating system

The well-defined interface between container and operating system means it is generally of no consequence which underlying Linux distribution is used. The operating system can be updated or even replaced without affecting the containers.

Swiss data sovereignty

For Swiss companies in regulated industries – banking, insurance, healthcare, and the public sector – data residency is a non-negotiable requirement. Kubernetes deployed on Swiss infrastructure keeps all data subject exclusively to Swiss law, with no exposure to the US CLOUD Act or other extraterritorial legislation. This is a key reason why organisations choose Swiss-based managed Kubernetes providers over hyperscaler offerings.

The Kubernetes Application Lifecycle

The transition to Kubernetes and container-based applications brings long-term change to well-known procedures. To create a holistic environment in which real and sustainable success can be seen, the change process has to be planned and implemented correctly from the beginning. The process itself can be broken down into two main phases:

CI/CD (Continuous Integration / Continuous Deployment)

Continuous Integration (CI)

Complex processes are needed for a new code version to make its way from a local machine to a cluster in production. From a technical perspective, pipelines need to be in place to ensure that all automated tests work successfully. After that, container images have to be built and pushed to a registry, while the Kubernetes configuration needs to be patched accordingly.

Ideally, the entire CI process should run in the background, while the necessary changes are pushed to one central place via a version control system (VCS). The CI system is automatically activated to test the amended code in a production-like environment. Kubernetes containers should be seen as immutable: their contents should not be changed at runtime. Container images ensure that the container in the CI pipeline is identical to the one in production.

Image scanning is essential: A robust CI pipeline includes automated scanning of container images for known vulnerabilities (CVEs). Tools such as Trivy, Grype, or integrated registry scanners catch security issues before they reach production.

Continuous Deployment (CD) and GitOps

CD presents a continuation of CI, as new versions are continuously published. The software is being continually updated without the need for a maintenance window. Modern CD workflows increasingly follow the **GitOps** pattern: the desired state of the cluster is declared in Git, and a tool such as ArgoCD or Flux continuously reconciles the live state with what is defined in the repository. This creates a fully auditable, version-controlled deployment process and simplifies rollbacks.

Observability

With a fully automated testing and deployment pipeline in place, the process can move on to the observability phase.

Continuous, small improvements ensure availability

Monitoring the application to identify outages early on is critical. Service Reliability Engineering (SRE) is based on the idea that service availability is a prerequisite for success. Companies should define Service Level Indicators (SLIs) and Service Level Objectives (SLOs) and implement them via monitoring and alerting systems. This allows for the creation of an error budget³ which defines how many outages or service disruptions are acceptable.

A modern Kubernetes observability stack typically consists of:

- **Prometheus** for metrics collection and alerting rules
- **Grafana** for dashboards and visualisation
- **Loki** or **Elasticsearch** for centralised log aggregation
- **Alertmanager** for routing and managing alerts

Secret management

Kubernetes Secrets are base64-encoded by default, which is not encryption. For production environments, external secret management – via tools such as the **External Secrets Operator** integrating with HashiCorp Vault, AWS Secrets Manager, or Google Secret Manager – is strongly recommended.

Automatic restarts mitigate small bugs

Kubernetes automatically restarts containers that fail liveness or readiness probes, mitigating non-critical bugs until they are fixed. For this to work correctly, the application's performance profile must be defined in advance: CPU requests and

limits, memory limits, and readiness/liveness probe configurations. Without proper resource configuration, applications may fail to schedule or enter CrashLoopBackOff.

Disaster recovery creates security

While Kubernetes absorbs certain outages within the underlying structure, companies should rely on disaster recovery as a back-up plan for cases where an entire cluster becomes unavailable. Cluster configurations and data must be backed up automatically, transferred to a different location, and retained for a defined period. Additional software is necessary to automate this process reliably.

³ Error budget = maximum amount of time a technical system can fail without contractual consequences. In SRE setups, when an error budget is exhausted, the team refocuses on platform stability over feature development.

Keeping Up with the Kubernetes Release Cycle

Kubernetes releases three minor versions per year. Each version is supported for approximately 14 months (N-2 policy). This means companies must upgrade their clusters regularly, as skipping too many releases creates security exposure and eventual incompatibility.

In traditional development, maintenance windows are used to collect changes and publish a new release every quarter. With Kubernetes, rolling out smaller, more frequent updates is far superior: errors can be fixed during business hours in a faster and more efficient manner, and individual changes are easier to trace and roll back.

Implementation: What Companies Need to Consider

Implementing Kubernetes is complex, as demonstrated by the lifecycle phases above. Nevertheless, its growing adoption shows that it is worth the effort. Thorough preparation is essential to ensure that you do not lose track of your goals in the implementation phase.

Checklist: Preparation Tips

Demand management: What are the requirements for the transition to Kubernetes? What are the necessary technical prerequisites?

Workflow consulting: How can a successful workflow be achieved? Who owns CI/CD and who is responsible for the platform?

Running Kubernetes and lifecycle management: What are the steps needed for the product lifecycle – including cluster upgrades – and how can they be implemented? On what cadence will upgrades be performed?

Tool choice: What additional tools are necessary to ensure optimal functionality for both containers and Kubernetes (CI/CD, secret management, GitOps, image registry, observability)?

Running additional services: What does monitoring the systems look like, and who is responsible? Who ensures uptime and takes care of comprehensive lifecycle management, including regular cluster updates? Can Managed Kubernetes ease the burden here?

Optimisation measures: The ecosystem regularly offers new features and tools. Who is responsible for maintenance and updates? How is the cluster kept compliant with the latest security policies?

Why Finding the Right Software Tools Matters

When running Kubernetes, many other actions are required for a successful operation. Tasks such as monitoring and the regular maintenance of Kubernetes clusters represent an enormous expenditure for companies, and should not be underestimated. Clusters should be managed well, as even small adjustments such as scaling should not be made without sufficient knowledge. If the necessary expertise is missing, companies may face additional costs or performance losses.

Functions needed to run Kubernetes successfully

- **The ability to monitor metrics** to keep an eye on your application 24/7 and find out whether it performs within set boundaries. Prometheus, Grafana, and Alertmanager have become the standard observability stack.
- **A robust logging system** to display logs from earlier container runs, save logs in one central place, and correlate them with metrics. Loki is the most widely adopted log aggregation solution for Kubernetes environments.
- **A scaling mechanism** that offers autoscaling options based on CPU, RAM, and custom metrics, from HorizontalPodAutoscaler to cluster-level node autoscaling.
- **Reliable certificate management** to automatically provide TLS certificates, including automated Let's Encrypt certificates via cert-manager.
- **A complete cluster backup** to ensure that in an emergency, saved data and cluster configurations can be restored elsewhere.
- **Active 24/7 monitoring** to take the right measures when IT problems occur. When partnering with a service provider such as us, we take care of these tasks for you.
- **Regular cluster updates** to avoid faulty configurations, outages, or security vulnerabilities. Kubernetes releases three minor versions per year, each requiring careful planning and execution.

- **Secret management** via the External Secrets Operator or similar tools, ensuring that sensitive configuration values never live unencrypted in your Git repositories.
- **GitOps tooling** such as ArgoCD or Flux, enabling fully declarative, auditable deployments with automatic drift detection and rollback capabilities.
- **A private container registry** to store and distribute container images securely within your infrastructure, without dependency on public registries.

Experienced Service Providers Support Companies

To make sure you never lose track when running Kubernetes and containers, IT partnerships with experienced service providers can be just the right fit. Experienced experts support you in your transition to container orchestration technologies and ensure that all systems and functionalities work as expected on a day-to-day basis. Such partnerships not only reduce operational burden but also cost, as no lengthy experimentation is needed.

We also offer targeted training – free introductory [Kubernetes webinars](#) (one hour, fundamentals and NKE in practice) and the full-day [Nine Kubernetes Academy](#) (CHF 800 / CHF 600 for NKE customers) – to help teams get up to speed with Kubernetes quickly.

Conclusion

For businesses, running Kubernetes means thorough preparation and long-term thinking. This includes identifying all potential challenges surrounding the software release pipeline, as well as determining who will be responsible for platform lifecycle management, cluster upgrades, observability, and security.

As this whitepaper shows, running Kubernetes comes with a high degree of complexity. The ecosystem evolves rapidly: Kubernetes releases three new minor versions per year, and best practices around security, CD tooling, and runtime have changed significantly over the past few years. To implement it successfully in the long run, companies should consider making the necessary resources available early on.

Many companies rely on a trustworthy partner and service provider who can support them during implementation as well as day-to-day operations, and who can take on tasks not directly related to their value-creating core business. This ensures they can continue to focus on the development of their own applications while following cloud-native best practices, without the added burden of running additional services critical for their daily business.

For companies with data sovereignty requirements – in Switzerland, this includes any organisation in finance, healthcare, insurance, or the public sector – selecting a managed Kubernetes provider with Swiss data residency is not optional. It is a compliance requirement. Running Kubernetes on Swiss infrastructure ensures that all data remains subject exclusively to Swiss law, with no exposure to extraterritorial legislation such as the US CLOUD Act.

About Nine Internet Solutions AG

We are a leading Swiss provider of managed cloud and infrastructure services, with over 25 years of experience. We offer full platform management in the public and private cloud, both hosted in Switzerland. We are ISO 27001 and ISO 9001 certified and employ around 35 people. Our engineers hold Certified Kubernetes Administrator (CKA) and Google Cloud Professional Architect certifications.

We stand for Swiss data sovereignty, highest availability, 24/7 monitoring, personal support and full scalability.

Would you like to find out more about running Kubernetes successfully in your business?

We are happy to answer any questions and support you from day one to day two and beyond.

[Learn more about NKE](#)

Nine Internet Solutions AG

Badenerstrasse 47 · 8004 Zurich · Switzerland
info@nine.ch · +41 44 637 40 00 · nine.ch