KubeOps

Kubernetes Security in Critical Infrastructures

Best Practices for Resilience and Compliance

KubeOps

Kubernetes Security in Critical Infrastructures

Best Practices for Resilience and Compliance

Introduction

Kubernetes is a powerful tool for operating modern containerised applications. However, targeted measures are required to minimise risks and make the infrastructure resilient, especially in the KRITIS sector, where there are high requirements for security, availability and compliance.
Operators of critical infrastructures face the challenge of implementing Kubernetes securely and efficiently. Cyberattacks on IT systems in sensitive areas can have serious consequences - from financial damage to a loss of trust among partners and customers. It is therefore crucial to recognise security risks at an early stage and secure them with proven best practices.

This white paper offers practical recommendations for decision-makers and technical managers to protect Kubernetes clusters from threats, fulfil regulatory requirements and ensure a secure operating environment in the long term.

Security Measures for Kubernetes Environments

The following key messages provide a compact overview of central security measures for Kubernetes environments in critical infrastructures:

- **Minimise attack surfaces:** Air-gapped clusters, reverse proxies and precise network segmentation reduce security risks and prevent unauthorised access.
- **Ensure a secure software supply chain:** Trusted software sources, signed container images and regular security scans protect the Kubernetes environment from manipulation and attacks.
- **Secure deployment and operating processes:** Internet-free deployments, resource restrictions and monitoring mechanisms can be used to efficiently protect Kubernetes clusters.
- **Clearly define access rights:** Role-based access control (RBAC), namespaces and secrets management ensure that only authorised users and applications have access to critical resources.
- **Establish proactive security monitoring:** Automated threat detection with tools such as Prometheus, Falco and Sysdig helps to identify and respond to attacks in real time.

KubeOps

Table of Content

Reduction of the Attack Surface through Network and Internet Access

Kubernetes clusters are highly networked systems that usually communicate via various internal and external interfaces. Insufficient control of these network connections can lead to attackers gaining access to the cluster via the internet or internal vulnerabilities.

Why is this a problem?

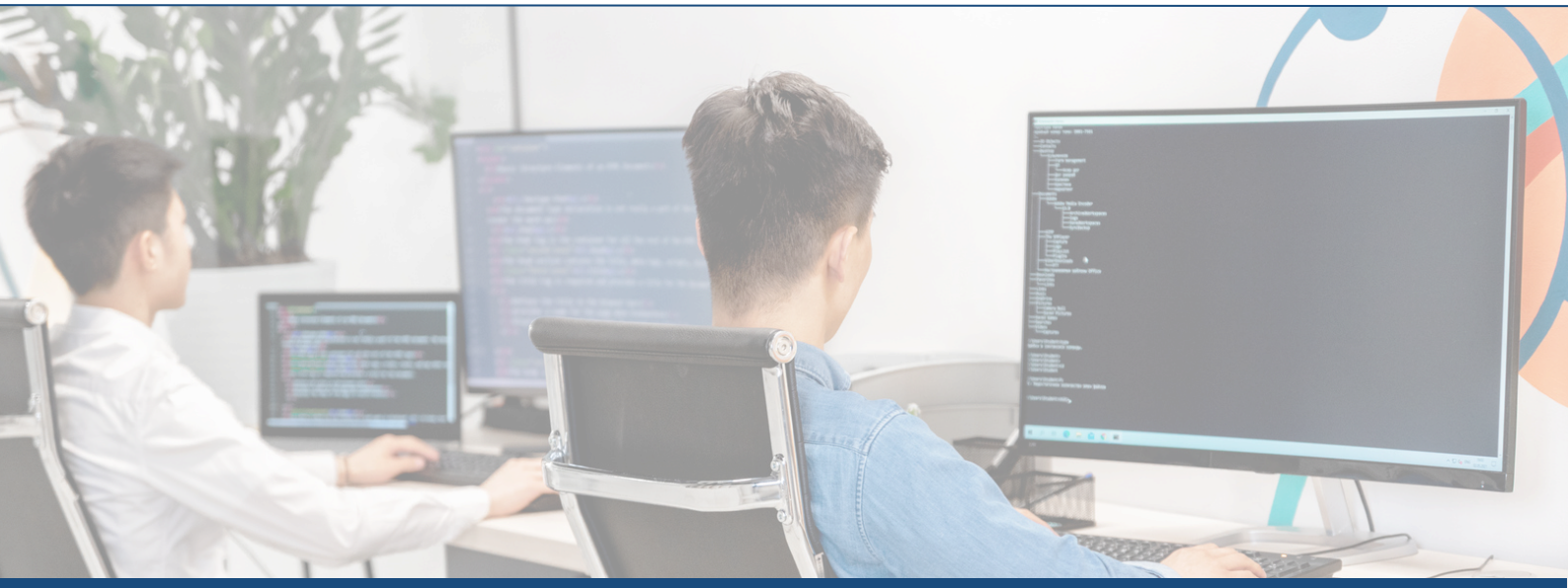
- By default, many Kubernetes clusters are configured to use external container registries or other cloud services - potentially opening doors for attacks.
- Misconfigured network policies allow uncontrolled internal and external connections that can be exploited by attackers.
- Many attacks on Kubernetes environments are based on access to unsecured API endpoints or insufficiently protected services.

Measures

→ **Air-Gapped cluster:**
A completely isolated cluster does not require an internet connection, thus minimising the attack surface.
Advantages: No external dependencies, complete control over resources.

→ **Reverse proxy for controlled Internet access:**
With a reverse proxy, data traffic can be specifically controlled and logged.
Additional benefit: Suspicious activities can be recognised and blocked.

→ **Ingress and egress rules:**
Use tools such as Calico to define network policies and prevent unauthorised data traffic.



Control of the Origin of Software and Dependencies

Most Kubernetes applications are based on container images that originate either from public registries (e.g. Docker Hub) or internal repositories. Each of these images can contain numerous dependencies on which the security of the entire cluster depends.

Why is this a problem?

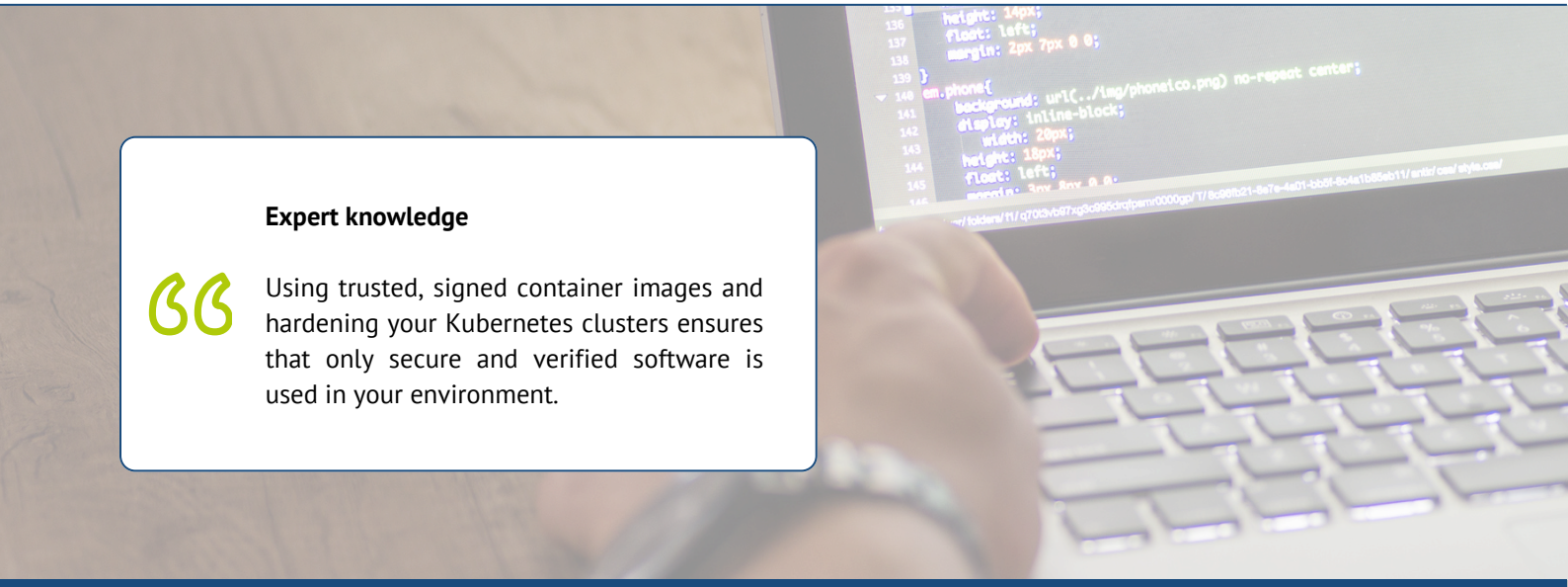
- Software sources that have not been checked: Malicious code or vulnerabilities in external images can get directly into the cluster.
- Outdated dependencies: If container images are not updated regularly, known vulnerabilities (CVEs) can remain unprotected.
- Manipulated software supply chains: Attacks on software supply chains have risen sharply in recent years as attackers increasingly attempt to penetrate development or deployment processes directly.

Measures

→ **Trusted software sources:**
Only use signed container images from reliable sources.
A separate container registry ensures that only verified images are used.

→ **Hardening the cluster:**
Rely on lean base images such as Alpine Linux to minimise potential vulnerabilities.
Regular security scans with tools such as Trivy or Clair identify vulnerabilities at an early stage.

→ **Supply chain security:**
Validate all software components and dependencies in your environment.
Implement a clear policy for dealing with open source software.



Expert knowledge

Using trusted, signed container images and hardening your Kubernetes clusters ensures that only secure and verified software is used in your environment.

Secure Deployment

Kubernetes enables applications to be deployed quickly and flexibly. However, without suitable security measures, deployment processes can become a gateway for threats.

Why ist this a problem?

- Uncontrolled deployments: If deployments are made directly from external sources (e.g. GitHub or Docker Hub), untrusted or manipulated images can be injected.
- Misuse of resources: Without defined CPU and memory limits, a single deployment can overload critical resources and destabilise the cluster.
- Lack of monitoring: Faulty or compromised deployments can go unnoticed if no suitable monitoring and checking mechanisms are implemented.

Measures

→ **Deployment without internet access:**
Isolate the deployment process to avoid dependencies on external resources.

→ **Signed container images:**
Tools such as Cosign enable the cryptographic signature of images and guarantee that only authorised software is used.

→ **Resource management:**
Set clear CPU and memory limits to ensure the stability of the cluster (Kubernetes Resource Management).

→ **Liveness and readiness probes:**
These mechanisms monitor the status of the containers and minimise downtimes.

→ **Security contexts for pods and containers:**
Set security policies for pods and containers to prevent applications from running with root privileges or being granted unnecessary permissions.

→ **Namespaces in the Kubernetes cluster:**
Isolate resources with namespaces to securely separate applications and teams within a cluster.

→ **Secure management of sensitive data:**
Store sensitive data in Kubernetes Secrets to manage passwords, API keys and certificates encrypted and protected from unauthorised access.

→ **Flexible configuration management:**
Use ConfigMaps for external management of configurations to make changes without having to restart containers or rebuild images.

Infrastructure and Setup of a Secure Kubernetes Cluster

A secure Kubernetes cluster starts with the underlying infrastructure. Misconfigurations in networks, access rights or storage systems can cause serious security vulnerabilities.

Why is this a problem?

- Lack of network separation: Without appropriate segmentation, attackers can move laterally within the cluster and compromise other systems.
- Vulnerabilities in API security: The Kubernetes API is the heart of the cluster - inadequate security can allow attackers to gain administrative rights.
- Insufficient authorisation management: Authorisations for users and services are often assigned too broadly, which increases the risk of insider attacks or unauthorised access.

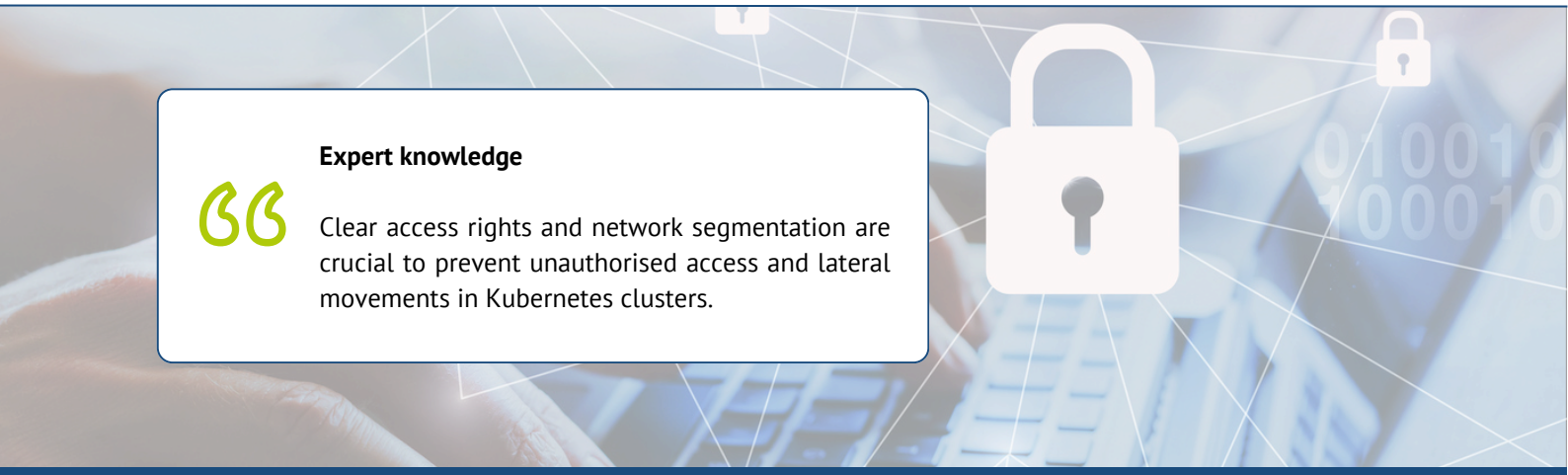
Measures

→ **Secure cluster infrastructure**
Harden nodes, segment networks and implement security measures to secure Kubernetes clusters against external and internal threats and ensure a secure operating environment.

→ **Network segmentation and isolation:**
Segment your network with firewalls and policies to isolate cluster areas, prevent unauthorised traffic and increase security.

→ **Use of Transport Layer Security (TLS):**
Encrypt all cluster communication with TLS to protect the integrity and confidentiality of sensitive data.

→ **Role-based access control (RBAC):**
Use RBAC to granularly control access rights so that only authorised users and applications can access critical resources.



Expert knowledge

Clear access rights and network segmentation are crucial to prevent unauthorised access and lateral movements in Kubernetes clusters.

Monitoring and Continuous Safety Assessment

Even the best security measures lose their effectiveness if threats are not recognised and responded to. Continuous monitoring and regular security checks are essential to secure Kubernetes clusters in the long term.

Why is this a problem?

- Lack of real-time monitoring: Without a comprehensive monitoring setup, attacks or misconfigurations can go unnoticed.
- Irregular security checks: Many companies only carry out security analyses sporadically - giving attackers plenty of time to exploit vulnerabilities.
- Lack of automation: Manual checks are inefficient and error-prone - a modern security concept must be based on automated scans and threat detection.

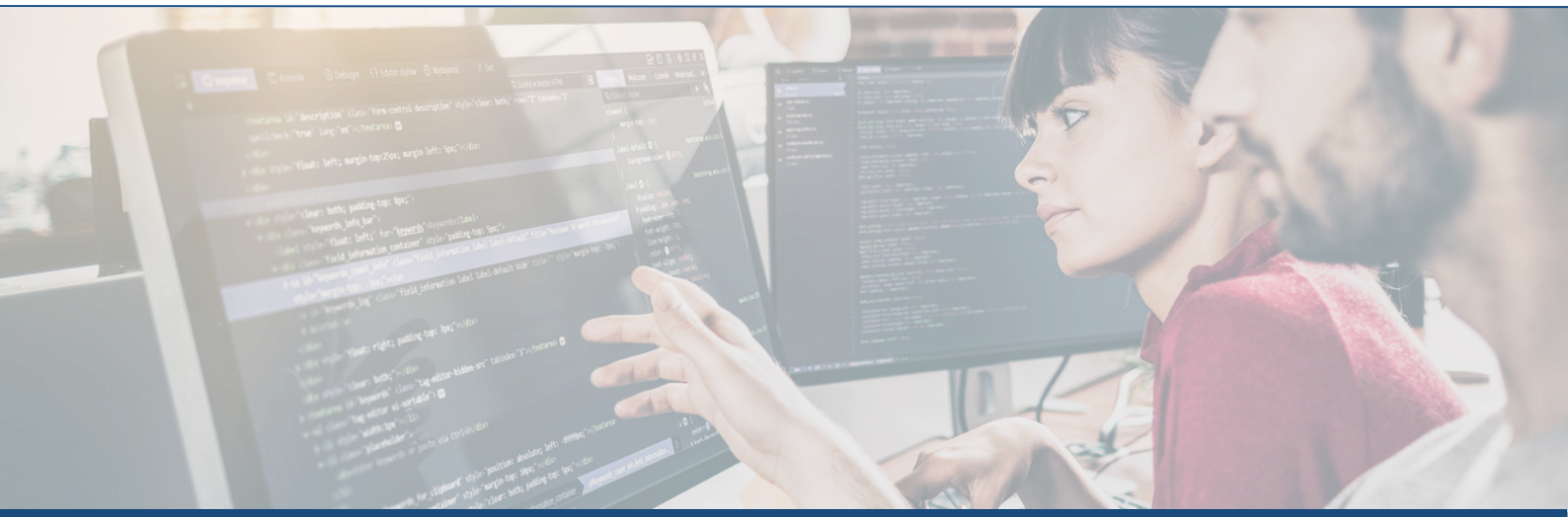
Measures

→ **Monitoring and surveillance:**
With tools such as Prometheus and Grafana, which provide real-time data on cluster performance, problems can be detected at an early stage and a quick response can be made to unusual behaviour.

→ **Continuous security assessments**
Carry out regular security reviews to identify risks at an early stage, adapt security measures and protect the infrastructure in the long term.

→ **Automated security scans**
Use regular scans with tools such as Clair and Trivy to detect and eliminate vulnerabilities in container images and running containers at an early stage.

→ **Threat detection:**
Solutions such as Falco and Sysdig enable real-time monitoring and analysis of security incidents.



Conclusion

Core strategies for secure Kubernetes environments

This white paper has shown how a combination of best practices and modern security approaches can significantly increase the security of Kubernetes clusters.
The strategies listed offer a comprehensive approach to making the operation of Kubernetes environments secure and future-proof.

- **Air gapped approaches:** Isolating Kubernetes clusters to minimise external threats.
- **Trusted software sources:** Use of signed and verified software from secure sources.
- **Infrastructure hardening:** Strengthening the cluster infrastructure through targeted measures to reduce vulnerabilities.
- **Continuous monitoring:** Proactive monitoring and threat detection through automated security assessments.
- **Future-proof security strategies:** Anticipation of future challenges through continuous adaptation of security measures.
- **Integration of proven tools:** Utilisation of Calico and Multus for a secure and scalable network architecture.
- **Holistic security approach:** Focus on preventive measures and continuous optimisation to ensure the secure operation of modern IT infrastructures.

These approaches provide a solid foundation for the protection of Kubernetes environments and help to ensure that IT infrastructures not only meet current security requirements, but are also prepared for future challenges. The continuous focus on security and prevention ensures the long-term stability and reliability of Kubernetes clusters.

Expert knowledge

Operating Kubernetes securely means minimising attack surfaces and keeping an eye on risks. A well-secured infrastructure with clear network boundaries, encrypted communication and strict access rights lays the foundation. It is just as important to only use trustworthy software and to keep containers lean and secure. Automated scans and continuous monitoring help to recognise problems at an early stage. As threats are constantly evolving, the security strategy should also be regularly adapted to ensure that the environment remains stable and protected in the long term.

Excursus: Container Hardening - an Essential Building Block in Kubernetes Security

The security of a Kubernetes cluster begins not only at the network and infrastructure level, but also directly with the containers themselves. Container hardening includes measures aimed at making container images as secure as possible in order to minimise attack surfaces.

Why is container hardening important?

❌ Containers often contain unnecessary packages and libraries that may have security vulnerabilities.

❌ Without restrictions, many containers run with excessive authorisations, which can endanger the entire cluster in the event of an attack.

❌ Manipulated or outdated images represent a considerable risk, especially if they originate from external, untrustworthy sources.

Best practices for container hardening

✅ Use minimal base images: Images such as Distroless or Alpine Linux significantly reduce attack vectors.

✅ Use signed and verified container images: Tools such as Cosign ensure that only authenticated images are loaded into Kubernetes.

✅ Favour rootless containers: Setting runAsNonRoot in Kubernetes deployments prevents attackers from gaining root rights.

✅ Perform automated security scans: Scanners such as Trivy or Clair help to identify and close known vulnerabilities (CVEs) at an early stage.

Well thought-out container hardening is one of the most effective measures for minimising security risks in Kubernetes environments.

About KubeOps

KubeOps GmbH was founded in 2019 as a subsidiary of ARWINET GmbH.

Our mission is to enable KRITIS organisations to build a robust container infrastructure quickly and efficiently. We understand the specific requirements of our customers, support them in setting up secure, resilient Kubernetes clusters and ensure their stable operation.

By using open source Kubernetes and carefully integrated components, we create automated, highly available and hardened clusters that are independent of vendor lock-ins to maximise our customers' flexibility and security.

We also offer training and certification to enhance your Kubernetes expertise.