KubeOps

Kubernetes-Sicherheit in kritischen Infrastrukturen

Best Practices für Resilienz und Compliance

KubeOps

Kubernetes-Sicherheit in kritischen Infrastrukturen

Best Practices für Resilienz und Compliance

Einleitung

Kubernetes ist ein leistungsstarkes Werkzeug für den Betrieb moderner containerisierter Anwendungen. Besonders im KRITIS-Sektor, wo hohe Anforderungen an Sicherheit, Verfügbarkeit und Compliance bestehen, sind jedoch gezielte Maßnahmen erforderlich, um Risiken zu minimieren und die Infrastruktur widerstandsfähig zu gestalten.
Betreiber kritischer Infrastrukturen stehen vor der Herausforderung, Kubernetes sicher und effizient zu implementieren. Cyberangriffe auf IT-Systeme in sensiblen Bereichen können schwerwiegende Folgen haben – von finanziellen Schäden bis hin zu einem Vertrauensverlust bei Partnern und Kunden. Daher ist es entscheidend, Sicherheitsrisiken frühzeitig zu erkennen und mit bewährten Best Practices abzusichern.

Dieses Whitepaper bietet praxisnahe Handlungsempfehlungen für Entscheider und technische Verantwortliche, um Kubernetes-Cluster vor Bedrohungen zu schützen, regulatorische Anforderungen zu erfüllen und eine langfristig sichere Betriebsumgebung zu gewährleisten.

Sicherheitsmaßnahmen für Kubernetes-Umgebungen

Die folgenden Kernbotschaften geben einen kompakten Überblick über zentrale Sicherheitsmaßnahmen für Kubernetes-Umgebungen in kritischen Infrastrukturen:

- **Angriffsflächen minimieren:** Air-Gapped-Cluster, Reverse-Proxies und präzise Netzwerksegmentierung reduzieren Sicherheitsrisiken und verhindern unbefugten Zugriff.
- **Sichere Software-Lieferkette gewährleisten:** Vertrauenswürdige Softwarequellen, signierte Container-Images und regelmäßige Sicherheits-Scans schützen die Kubernetes-Umgebung vor Manipulationen und Angriffen.
- **Deployment- und Betriebsprozesse absichern:** Durch internetfreie Deployments, Ressourcenbeschränkungen und Monitoring-Mechanismen lassen sich Kubernetes-Cluster effizient schützen.
- **Zugriffsrechte klar definieren:** Mit rollenbasierter Zugriffskontrolle (RBAC), Namespaces und Secrets-Management wird sichergestellt, dass nur autorisierte Nutzer und Anwendungen Zugriff auf kritische Ressourcen haben.
- **Proaktives Sicherheits-Monitoring etablieren:** Automatisierte Bedrohungserkennung mit Tools wie Prometheus, Falco und Sysdig hilft, Angriffe in Echtzeit zu identifizieren und darauf zu reagieren.

KubeOps

Inhaltsverzeichnis

Reduzierung der Angriffsfläche durch Netzwerk- und Internetzugriffe

Kubernetes-Cluster sind hochvernetzte Systeme, die in der Regel über verschiedene interne und externe Schnittstellen kommunizieren. Eine unzureichende Kontrolle dieser Netzwerkverbindungen kann dazu führen, dass Angreifer über das Internet oder interne Schwachstellen Zugang zum Cluster erhalten.

Warum ist das ein Problem?

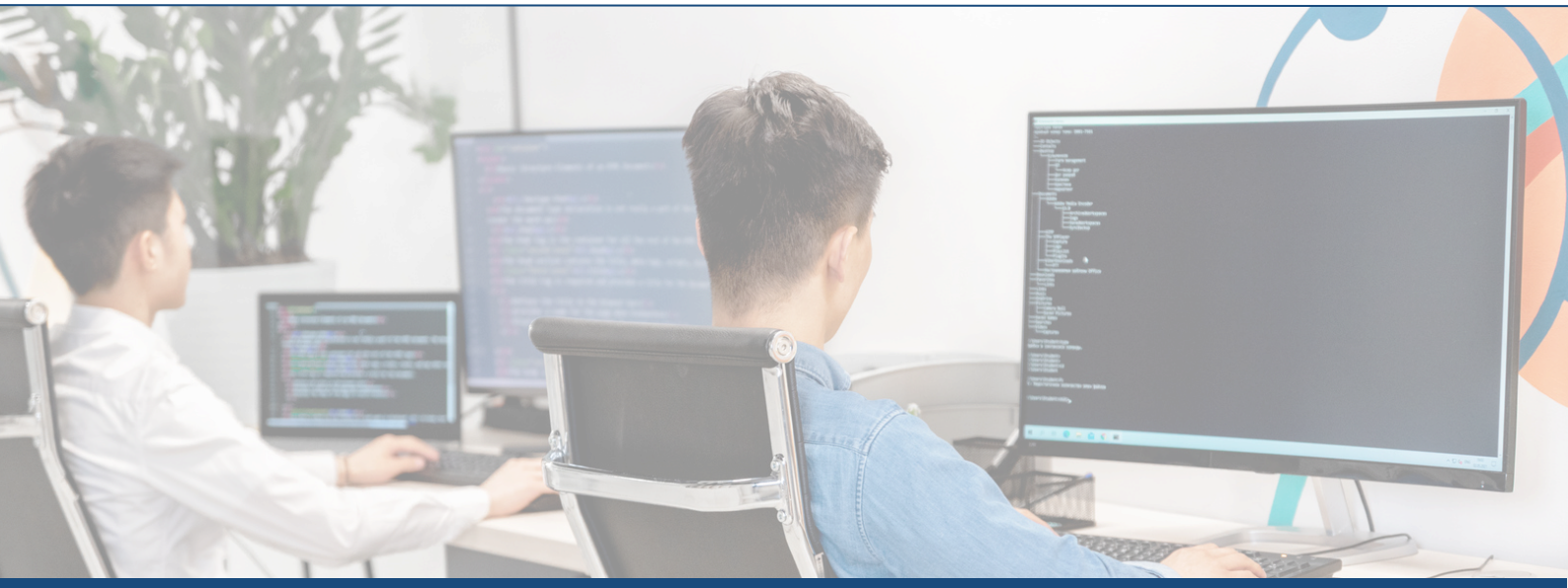
- Standardmäßig sind viele Kubernetes-Cluster so konfiguriert, dass sie externe Container-Registries oder andere Cloud-Dienste nutzen – das öffnet potenziell Türen für Angriffe.
- Fehlkonfigurierte Netzwerkrichtlinien ermöglichen unkontrollierte interne und externe Verbindungen, die von Angreifern ausgenutzt werden können.
- Viele Angriffe auf Kubernetes-Umgebungen basieren auf dem Zugriff auf ungesicherte API-Endpoints oder unzureichend geschützte Services.

Maßnahmen

→ **Air-Gapped-Cluster:**
Ein vollständig isoliertes Cluster benötigt keine Internetverbindung und minimiert so die Angriffsfläche.
Vorteile: Keine externen Abhängigkeiten, vollständige Kontrolle über Ressourcen.

→ **Reverse-Proxy für kontrollierten Internetzugriff:**
Mit einem Reverse-Proxy lässt sich der Datenverkehr gezielt steuern und Protokollieren.
Zusatznutzen: Verdächtige Aktivitäten können erkannt und blockiert werden.

→ **Ingress- und Egress-Regeln:**
Nutzen Sie Tools wie Calico, um Netzwerkrichtlinien zu definieren und unbefugten Datenverkehr zu verhindern.



Kontrolle der Herkunft von Software und Abhängigkeiten

Die meisten Kubernetes-Anwendungen basieren auf Container-Images, die entweder aus öffentlichen Registries (z. B. Docker Hub) oder internen Repositories stammen. Jedes dieser Images kann zahlreiche Abhängigkeiten enthalten, von denen die Sicherheit des gesamten Clusters abhängt.

Warum ist das ein Problem?

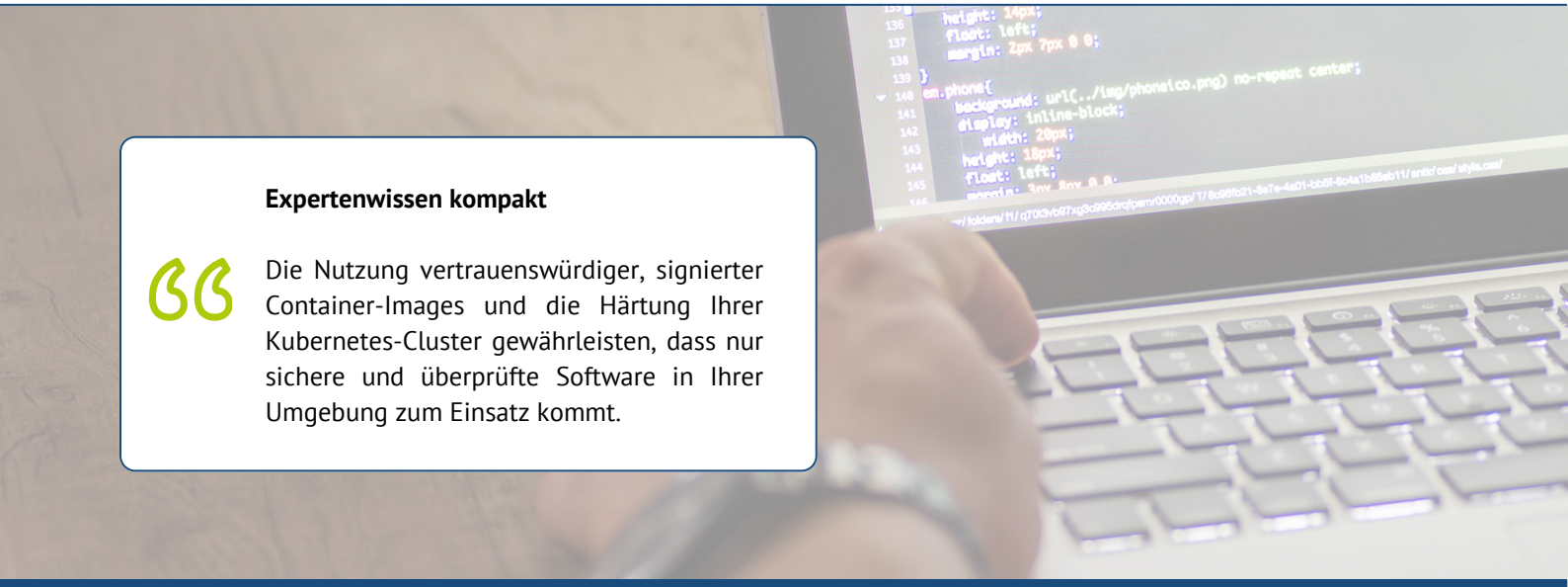
- Nicht überprüfte Softwarequellen: Schadcode oder Schwachstellen in externen Images können direkt in den Cluster gelangen.
- Veraltete Abhängigkeiten: Wenn Container-Images nicht regelmäßig aktualisiert werden, können bekannte Sicherheitslücken (CVEs) ungeschützt bleiben.
- Manipulierte Software-Lieferketten: Angriffe auf Software-Supply-Chains sind in den letzten Jahren stark gestiegen, da Angreifer zunehmend versuchen, direkt in die Entwicklungs- oder Bereitstellungsprozesse einzudringen.

Maßnahmen

→ **Vertrauenswürdige Softwarequellen:**
Verwenden Sie ausschließlich signierte Container-Images aus zuverlässigen Quellen.
Eine eigene Container-Registry gewährleistet, dass nur geprüfte Images verwendet werden.

→ **Härtung der Cluster:**
Setzen Sie auf schlanke Base Images wie Alpine Linux, um potenzielle Schwachstellen zu minimieren.
Regelmäßige Sicherheits-Scans mit Tools wie Trivy oder Clair identifizieren Schwachstellen frühzeitig.

→ **Supply Chain Security:**
Validieren Sie alle Software-Komponenten und Abhängigkeiten in Ihrer Umgebung.
Implementieren Sie eine klare Richtlinie für den Umgang mit Open-Source-Software.



Expertenwissen kompakt

Die Nutzung vertrauenswürdiger, signierter Container-Images und die Härtung Ihrer Kubernetes-Cluster gewährleisten, dass nur sichere und überprüfte Software in Ihrer Umgebung zum Einsatz kommt.

Sicheres Deployment

Kubernetes ermöglicht eine schnelle und flexible Bereitstellung von Anwendungen. Doch ohne geeignete Sicherheitsmaßnahmen können Deployment-Prozesse zum Einfallstor für Bedrohungen werden.

Warum ist das ein Problem?

- Unkontrollierte Deployments: Wenn Deployments direkt aus externen Quellen (z. B. GitHub oder Docker Hub) erfolgen, können nicht vertrauenswürdige oder manipulierte Images eingeschleust werden.
- Ressourcenmissbrauch: Ohne festgelegte CPU- und Speicherlimits kann ein einzelnes Deployment kritische Ressourcen überlasten und den Cluster destabilisieren.
- Fehlende Überwachung: Fehlerhafte oder kompromittierte Deployments können unbemerkt bleiben, wenn keine geeigneten Monitoring- und Prüfmechanismen implementiert sind.

Maßnahmen

→ **Deployment ohne Internetzugang:**
Isolieren Sie den Deployment-Prozess, um Abhängigkeiten von externen Ressourcen zu vermeiden.

→ **Signierte Container-Images:**
Tools wie Cosign ermöglichen die kryptografische Signatur von Images und garantieren, dass nur autorisierte Software eingesetzt wird.

→ **Ressourcenmanagement:**
Setzen Sie klare CPU- und Speicherlimits, um die Stabilität des Clusters zu gewährleisten (Kubernetes Resource Management).

→ **Liveness- und Readiness-Probes:**
Diese Mechanismen überwachen den Zustand der Container und minimieren Ausfallzeiten.

→ **Sicherheitskontexte für Pods und Container:**
Setzen Sie Sicherheitsrichtlinien für Pods und Container, um zu verhindern, dass Anwendungen mit Root-Rechten ausgeführt werden oder unnötige Berechtigungen erhalten.

→ **Namespaces im Kubernetes-Cluster:**
Isolieren Sie Ressourcen mit Namespaces, um Anwendungen und Teams innerhalb eines Clusters sicher voneinander zu trennen.

→ **Sichere Verwaltung sensibler Daten:**
Speichern Sie sensible Daten in Kubernetes Secrets, um Passwörter, API-Schlüssel und Zertifikate verschlüsselt und vor unbefugtem Zugriff geschützt zu verwalten.

→ **Flexible Konfigurationsverwaltung:**
Nutzen Sie ConfigMaps für die externe Verwaltung von Konfigurationen, um Änderungen vorzunehmen, ohne Container neu starten oder Images neu bauen zu müssen.

Infrastruktur und Aufbau eines sicheren Kubernetes-Clusters

Ein sicherer Kubernetes-Cluster beginnt bei der zugrunde liegenden Infrastruktur. Fehlkonfigurationen in Netzwerken, Zugriffsrechten oder Speichersystemen können schwerwiegende Sicherheitslücken verursachen.

Warum ist das ein Problem?

- Fehlende Netzwerktrennung: Ohne geeignete Segmentierung können Angreifer sich lateral innerhalb des Clusters bewegen und weitere Systeme kompromittieren.
- Schwachstellen in der API-Sicherheit: Die Kubernetes-API ist das Herzstück des Clusters – eine unzureichende Absicherung kann es Angreifern ermöglichen, administrative Rechte zu erlangen.
- Unzureichendes Berechtigungsmanagement: Oft werden zu weitreichende Berechtigungen für Benutzer und Services vergeben, was das Risiko von Insider-Angriffen oder unautorisierten Zugriffen erhöht.

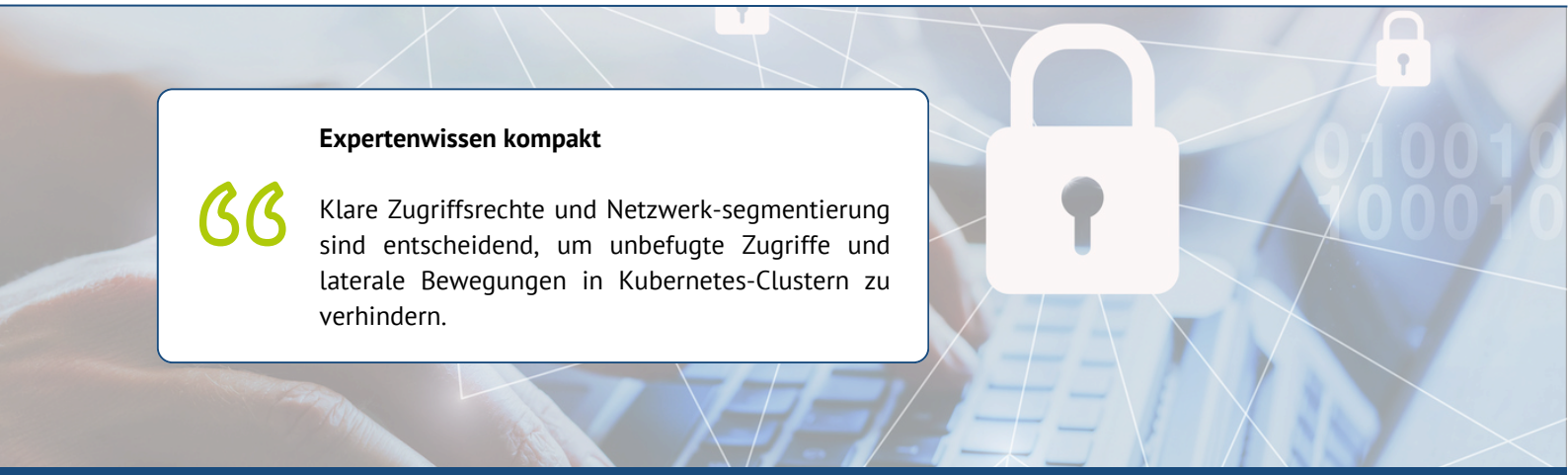
Maßnahmen

→ **Sichere Cluster-Infrastruktur**
Härten Sie die Nodes, segmentieren Sie Netzwerke und implementieren Sie Sicherheitsmaßnahmen, um Kubernetes-Cluster gegen externe und interne Bedrohungen abzusichern und eine sichere Betriebsumgebung zu gewährleisten.

→ **Netzwerksegmentierung und -isolierung:**
Segmentieren Sie Ihr Netzwerk mit Firewalls und Policies, um Cluster-Bereiche zu isolieren, unautorisierten Datenverkehr zu verhindern und die Sicherheit zu erhöhen.

→ **Verwendung von Transport Layer Security (TLS):**
Verschlüsseln Sie die gesamte Cluster-Kommunikation mit TLS, um die Integrität und Vertraulichkeit sensibler Daten zu schützen.

→ **Rollenbasierte Zugriffskontrolle (RBAC):**
Setzen Sie RBAC ein, um Zugriffsrechte granular zu steuern, sodass nur autorisierte Benutzer und Anwendungen auf kritische Ressourcen zugreifen können.



Expertenwissen kompakt

Klare Zugriffsrechte und Netzwerk-segmentierung sind entscheidend, um unbefugte Zugriffe und laterale Bewegungen in Kubernetes-Clustern zu verhindern.

Monitoring und kontinuierliche Sicherheitsbewertung

Auch die besten Sicherheitsmaßnahmen verlieren ihre Wirksamkeit, wenn Bedrohungen nicht erkannt und darauf reagiert wird. Ein kontinuierliches Monitoring und regelmäßige Sicherheitsüberprüfungen sind essenziell, um Kubernetes-Cluster langfristig abzusichern.

Warum ist das ein Problem?

- Fehlende Echtzeitüberwachung: Ohne ein umfassendes Monitoring-Setup können Angriffe oder Fehlkonfigurationen unbemerkt bleiben.
- Unregelmäßige Sicherheitsprüfungen: Viele Unternehmen führen Sicherheitsanalysen nur sporadisch durch – das lässt Angreifern viel Zeit, Schwachstellen auszunutzen.
- Mangelnde Automatisierung: Manuelle Überprüfungen sind ineffizient und fehleranfällig – ein modernes Sicherheitskonzept muss auf automatisierten Scans und Bedrohungserkennung basieren.

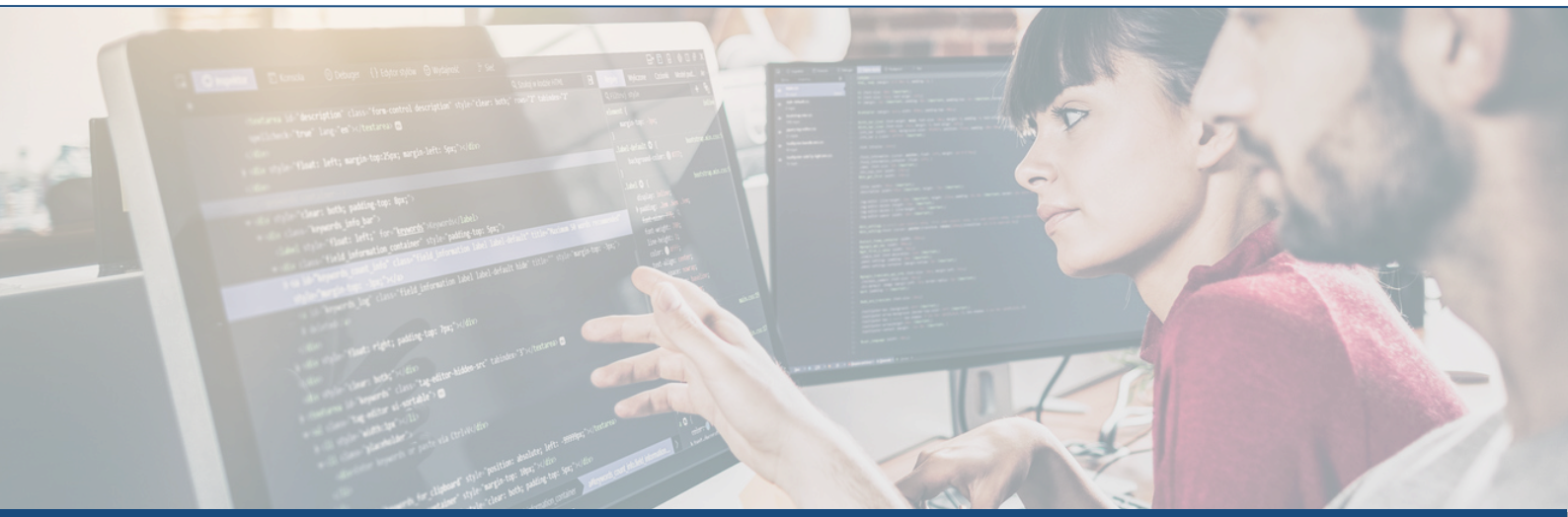
Maßnahmen

→ **Monitoring und Überwachung:**
Mit Tools wie Prometheus und Grafana, die Echtzeitdaten zur Cluster-Performance liefern, können Probleme frühzeitig erkannt und auf ungewöhnliches Verhalten schnell reagiert werden.

→ **Kontinuierliche Sicherheitsbewertungen**
Führen Sie regelmäßige Sicherheitsüberprüfungen durch, um Risiken frühzeitig zu identifizieren, Sicherheitsmaßnahmen anzupassen und die Infrastruktur dauerhaft zu schützen.

→ **Automatisierte Sicherheits-Scans**
Setzen Sie regelmäßige Scans mit Tools wie Clair und Trivy ein, um Schwachstellen in Container-Images und laufenden Containern frühzeitig zu erkennen und zu beheben.

→ **Bedrohungserkennung:**
Lösungen wie Falco und Sysdig ermöglichen die Echtzeitüberwachung und Analyse von Sicherheitsvorfällen.



Fazit

Kernstrategien für sichere Kubernetes-Umgebungen

Dieses Whitepaper hat gezeigt, wie durch eine Kombination von Best Practices und modernen Sicherheitsansätzen die Sicherheit von Kubernetes-Clustern erheblich gesteigert werden kann.
Die aufgeführten Strategien bieten einen umfassenden Ansatz, um den Betrieb von Kubernetes-Umgebungen sicher und zukunftssicher zu gestalten.

- **Air Gapped-Ansätze:** Isolation von Kubernetes-Clustern zur Minimierung externer Bedrohungen.
- **Vertrauenswürdige Softwarequellen:** Einsatz signierter und geprüfter Software aus sicheren Quellen.
- **Infrastrukturhärtung:** Stärkung der Cluster-Infrastruktur durch gezielte Maßnahmen zur Reduzierung von Schwachstellen.
- **Kontinuierliches Monitoring:** Proaktive Überwachung und Bedrohungserkennung durch automatisierte Sicherheits- bewertungen.
- **Zukunftssichere Sicherheitsstrategien:** Antizipation zukünftiger Herausforderungen durch kontinuierliche Anpassung der Sicherheitsmaßnahmen.
- **Integration bewährter Tools:** Nutzung von Calico und Multus für eine sichere und skalierbare Netzwerkarchitektur.
- **Ganzheitlicher Sicherheitsansatz:** Fokus auf präventive Maßnahmen und fortlaufende Optimierung zur Gewährleistung eines sicheren Betriebs moderner IT-Infrastrukturen.

Diese Ansätze bieten eine solide Grundlage für den Schutz von Kubernetes-Umgebungen und tragen dazu bei, dass IT-Infrastrukturen nicht nur den aktuellen Sicherheitsanforderungen gerecht werden, sondern auch auf zukünftige Herausforderungen vorbereitet sind. Der kontinuierliche Fokus auf Sicherheit und Prävention gewährleistet die langfristige Stabilität und Zuverlässigkeit der Kubernetes-Cluster.

Expertenwissen kompakt

Kubernetes sicher zu betreiben, bedeutet, Angriffsflächen zu minimieren und Risiken im Blick zu behalten. Eine gut abgesicherte Infrastruktur mit klaren Netzwerkgrenzen, verschlüsselter Kommunikation und strikten Zugriffsrechten legt die Basis. Genauso wichtig ist es, nur vertrauenswürdige Software zu nutzen und Container schlank und sicher zu halten. Automatisierte Scans und kontinuierliches Monitoring helfen, Probleme frühzeitig zu erkennen. Da sich Bedrohungen ständig weiterentwickeln, sollte auch die Sicherheitsstrategie regelmäßig angepasst werden – so bleibt die Umgebung langfristig stabil und geschützt.

Exkurs: Container-Härtung - Ein essenzieller Baustein in der Kubernetes Sicherheit

Die Sicherheit eines Kubernetes-Clusters beginnt nicht nur auf Netzwerk- und Infrastrukturebene, sondern direkt bei den Containern selbst. Container-Härtung umfasst Maßnahmen, die darauf abzielen, Container-Images so sicher wie möglich zu gestalten, um Angriffsflächen zu minimieren.

Warum ist Container-Härtung wichtig?

❌ Container enthalten oft unnötige Pakete und Bibliotheken, die Sicherheitslücken aufweisen können.

❌Ohne Einschränkungen laufen viele Container mit zu hohen Berechtigungen, was bei einem Angriff das gesamte Cluster gefährden kann.

❌ Manipulierte oder veraltete Images stellen ein erhebliches Risiko dar, insbesondere wenn sie aus externen, nicht vertrauenswürdigen Quellen stammen.

Best Practices zur Container-Härtung

✅ Minimale Base Images verwenden: Images wie Distroless oder Alpine Linux reduzieren Angriffsvektoren erheblich.
✅ Signierte und geprüfte Container-Images nutzen: Tools wie Cosign stellen sicher, dass nur authentifizierte Images in Kubernetes geladen werden.
✅ Rootless-Container bevorzugen: Durch das Setzen von runAsNonRoot in Kubernetes-Deployments wird verhindert, dass Angreifer Root-Rechte erhalten.
✅ Automatisierte Sicherheits-Scans durchführen: Scanner wie Trivy oder Clair helfen, bekannte Schwachstellen (CVEs) frühzeitig zu identifizieren und zu schließen.

Eine durchdachte Container-Härtung ist eine der effektivsten Maßnahmen, um Sicherheitsrisiken in Kubernetes-Umgebungen zu minimieren.

Über KubeOps

Die KubeOps GmbH wurde 2019 als Tochterunternehmen der ARWINET GmbH gegründet.

Unsere Mission ist es, KRITIS-Organisationen zu einem schnellen und effizienten Aufbau einer robusten Container-Infrastruktur zu befähigen. Wir verstehen die spezifischen Anforderungen unserer Kunden, unterstützen sie beim Aufbau sicherer, resilienter Kubernetes-Cluster und gewährleisten deren stabilen Betrieb.

Durch den Einsatz von Open-Source-Kubernetes und sorgfältig integrierten Komponenten schaffen wir automatisierte, hochverfügbare und gehärtete Cluster, die unabhängig von Herstellerbindungen sind, um so die Flexibilität und Sicherheit unserer Kunden zu maximieren.

Darüber hinaus bieten wir Ihnen Schulungen und Zertifizierungen zur Erweiterung Ihres Kubernetes-Fachwissen.