GUIDE

Kubernetes v1.36: Enhanced Security and AI Workload Support

> ⚠️ **HIGH PRIORITY** — Important fixes. Upgrade soon.

> **Version:** 1.36 | **Released:** 2026-04-22 | **Upgrade from:** 1.35.x

Kubernetes v1.36, codenamed "Haru" (Spring in Japanese), has sprung into action, bringing a robust set of enhancements primarily focused on tightening security and bolstering support for the rapidly evolving landscape of AI/ML workloads. This release marks a significant step forward, promoting critical security features to General Availability (GA) and introducing capabilities that directly address the unique demands of modern, resource-intensive applications.

Release at a Glance

- **Security by Default:** `MutatingAdmissionPolicy` is now GA (v1) and enabled by default, fundamentally enhancing admission control security. This is a critical shift, requiring operators to understand its implications for their cluster's security posture.
- **AI/ML Workload Optimization:** Features like `OCI volumes support` and `Horizontal Pod Autoscaler (HPA) scale-to-zero` provide tangible benefits for AI/ML pipelines, improving data management and resource efficiency for intermittent inference tasks.
- **Deeper Performance Insights:** `Pressure Stall Information (PSI)` metrics reaching GA offer unparalleled visibility into resource saturation, empowering engineers to proactively tune performance before bottlenecks impact user experience.
- **Breaking Change Alert:** The `gitRepo` volume type has been permanently disabled due to security risks. Immediate migration to `initContainers` or `emptyDir` with `git` commands is required for affected workloads.

Headline New Features

Kubernetes v1.36 delivers a suite of powerful new capabilities, with a strong emphasis on maturity and stability for features previously in beta.

MutatingAdmissionPolicy Graduates to GA

The `MutatingAdmissionPolicy` API has been promoted to GA (v1) and is now enabled by default. This is a game-changer for cluster security, allowing for powerful, declarative modifications to incoming API requests before they are persisted. 📌 **Key Idea:** This feature allows cluster administrators to enforce policies that automatically inject sidecars, set default values, or modify resource configurations based on predefined rules, enhancing security and operational consistency.

Finer-Grained Impersonation Controls

This release introduces more granular controls for impersonation, allowing users to restrict permissions when impersonating others. This aligns with the principle of least privilege, enabling administrators to define precise boundaries for delegated access. ⚡ **Real-world insight:** In multi-tenant environments or for auditing purposes, this allows for safer debugging and administrative tasks by ensuring that an impersonating user cannot exceed the permissions they would have had if they weren't impersonating at all.

External KMS for Service Account Token Signing

Kubernetes v1.36 now supports integration with external Key Management Solutions (KMS) for service account token signing. This removes the `kube-apiserver`'s sole dependency on its internal key for signing, enhancing security and allowing for better key lifecycle management, rotation, and auditing through dedicated KMS providers (e.g., cloud KMS services, HSMs).

Stable User Namespaces

User Namespaces have reached stable status, providing improved isolation and security for diverse workloads. This feature maps container UIDs/GIDs to different UIDs/GIDs on the host, preventing privilege escalation and enhancing multi-tenancy. 🧠 **Important:** While a powerful security primitive, user namespaces can introduce complexity with host-path volumes and certain privileged operations. Careful planning is essential.

OCI Volumes Support

New support for OCI volumes simplifies the management of container images and data, particularly beneficial for complex AI/ML pipelines. This allows for direct consumption of OCI artifacts as volumes, streamlining data and model distribution. ⚡ **Quick Note:** This can significantly simplify workflows where models or large datasets are packaged as OCI artifacts and need to be mounted directly into pods without intermediate steps.

HPA Scale-to-Zero Capability

The Horizontal Pod Autoscaler (HPA) now supports scaling down to zero pods. This is a highly anticipated feature, especially for intermittent workloads like AI inference services, where resources can be fully deallocated when not in use, leading to significant cost savings.

```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: my-ai-inference
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: my-ai-inference
  minReplicas: 0 # New: Allows scaling down to zero
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

🔥 **Optimization / Pro tip:** Combine HPA scale-to-zero with a service mesh or ingress controller that can buffer requests while the deployment scales up from zero, ensuring a smooth user experience.

Security Enhancements

Kubernetes v1.36 takes a proactive stance on security, embedding stronger protections directly into the platform's core.

Admission Control Hardening with MutatingAdmissionPolicy GA

The promotion of `MutatingAdmissionPolicy` to GA and its default enablement is a cornerstone security improvement. This allows for dynamic, policy-driven modification of resources at the API admission stage. For example, you can automatically inject security sidecars, ensure all pods have a specific `PodSecurityContext`, or enforce resource limits.

```
# Example MutatingAdmissionPolicy (simplified)
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingAdmissionPolicy
metadata:
  name: enforce-security-context
spec:
  rules:
  - apiGroups: [""]
    apiVersions: ["v1"]
    operations: ["CREATE", "UPDATE"]
    resources: ["pods"]
    scope: "Namespaced"
    matchPolicy: Equivalent
    # This policy would inject a default security context if not present
    # In a real scenario, this would involve a webhook service to perform the
mutation
    # and the policy would point to that service.
    # For illustration, imagine it ensures runAsNonRoot: true
```

This declarative approach to security policy enforcement reduces the chance of misconfigurations and provides a consistent security baseline across the cluster.

Enhanced Impersonation Security

The introduction of finer-grained impersonation controls directly enhances the principle of least privilege. Previously, impersonating another user granted the impersonator all of the impersonated user's permissions. Now, an impersonator can explicitly request a subset of their own permissions to be applied during the impersonation, preventing accidental or malicious over-privileging.

External Key Management for Service Account Tokens

Moving service account token signing to external KMS solutions significantly reduces the attack surface on the `kube-apiserver`. It allows organizations to leverage their existing, hardened key management infrastructure, improving compliance, auditability, and the overall security posture of service account credentials. This is particularly crucial for clusters handling sensitive data or operating under strict regulatory requirements.

Stable User Namespaces for Workload Isolation

With User Namespaces now stable, Kubernetes offers a more robust mechanism for isolating workloads. By remapping UIDs/GIDs, containers gain an additional layer of protection against privilege escalation attempts, even if a vulnerability within the container allows for root access. This makes multi-tenant clusters inherently more secure.

No specific CVEs were identified for this release in the provided snippets. This release focuses on introducing significant new security features and hardening existing mechanisms, rather than patching specific vulnerabilities.

Performance Improvements

Kubernetes v1.36 not only brings new features but also solidifies the tools for understanding and optimizing cluster performance.

Pressure Stall Information (PSI) Metrics to GA

The graduation of `Pressure Stall Information (PSI)` metrics to GA is a game-changer for performance tuning. PSI provides crucial insights into resource saturation (CPU, memory, I/O) by reporting how much time tasks spend waiting for resources. Unlike traditional utilization metrics, PSI directly indicates contention and starvation, allowing engineers to identify resource bottlenecks before they lead to noticeable performance degradation.

⚡ **Real-world insight:** Instead of seeing 80% CPU utilization and wondering if it's a problem, PSI might show that 10% of CPU time is spent waiting due to contention, indicating a hidden bottleneck. This allows for more precise scaling decisions or resource allocation adjustments.

Improved Scalability Test Benchmarks

SIG Scalability has officially increased the supported resource size in scalability tests from 800MB to 1.5GB. This isn't just a number; it indicates that the Kubernetes control plane and underlying components have been tested and proven stable and performant at significantly larger scales. This provides greater confidence for deploying larger, more complex clusters and workloads.

Breaking Changes and Removed APIs

As with any major release, Kubernetes v1.36 includes breaking changes that require attention during the upgrade process. The most critical change is the permanent removal of a legacy volume type.

Permanent Disablement of `gitRepo` Volume Type

The `gitRepo` volume type has been permanently disabled due to long-standing security concerns. This volume type allowed pods to pull Git repositories directly at startup, which created a significant attack vector if the repository or the Git client had vulnerabilities.

Why it matters: This change means any existing Pods, Deployments, StatefulSets, or other workloads that rely on `gitRepo` volumes will fail to start or operate correctly after upgrading to v1.36.

Migration Strategy: Users must migrate to more secure alternatives. The recommended approaches are:

1. **Using `initContainers`:** Use an `initContainer` to clone the Git repository into an `emptyDir` volume. The main application container then mounts this `emptyDir`.
2. **Using `emptyDir` with `git` commands:** Similar to `initContainers`, but the main container executes the `git clone` command itself upon startup (less common for application code).

Before (Kubernetes v1.35.x and earlier):

```
apiVersion: v1
kind: Pod
metadata:
  name: my-gitrepo-pod
spec:
  containers:
  - name: web-server
    image: nginx
    volumeMounts:
    - name: git-volume
      mountPath: /app/repo
  volumes:
  - name: git-volume
    gitRepo:
      repository: https://github.com/my-org/my-repo.git
      revision: master
      directory: .
```

After (Kubernetes v1.36 and later):

```
apiVersion: v1
kind: Pod
metadata:
  name: my-gitrepo-pod-migrated
spec:
  initContainers:
  - name: git-cloner
    image: alpine/git # Or any image with git installed
    command: ["git", "clone", "https://github.com/my-org/my-repo.git", "/app/
repo"]
    volumeMounts:
    - name: shared-data
      mountPath: /app/repo
  containers:
  - name: web-server
    image: nginx
    volumeMounts:
    - name: shared-data
      mountPath: /app/repo
  volumes:
  - name: shared-data
    emptyDir: {} # An emptyDir to store the cloned repository
```

⚠️ **What can go wrong:** Failing to migrate `gitRepo` volumes will result in pod creation failures. Thoroughly audit your cluster for any workloads using this deprecated volume type before upgrading.

How to Upgrade

Given the high urgency and significant security enhancements, planning your upgrade to Kubernetes v1.36 is crucial. Always consult the official Kubernetes documentation and changelog for the most precise, version-specific instructions.

1. **Review the Official Changelog:** Before any upgrade, carefully read the official changelog for v1.36 to understand all changes, especially any other minor breaking changes or deprecations not highlighted here.
2. **Backup Your Cluster:** Always back up your `etcd` data and Kubernetes configurations before starting an upgrade.
3. **Upgrade `kubeadm` (if applicable):** If you're using `kubeadm`, upgrade the `kubeadm` binary on your control plane nodes first.

```
    # Update package lists
    sudo apt-get update
    # Upgrade kubeadm
    sudo apt-get install -y kubeadm=1.36.0-00
```

```
Or for `yum`/`dnf`:
```

```
    sudo yum update
    sudo yum install -y kubeadm-1.36.0
```

1. **Upgrade Control Plane:** Initialize the upgrade on a control plane node.

```
    sudo kubeadm upgrade plan v1.36.0
    sudo kubeadm upgrade apply v1.36.0
```

1. **Upgrade `kubelet` and `kubectl`:** After the control plane is upgraded, upgrade `kubelet` and `kubectl` on all nodes (control plane and worker nodes).

```
    # Upgrade kubelet and kubectl
    sudo apt-get install -y kubelet=1.36.0-00 kubectl=1.36.0-00
    sudo systemctl restart kubelet
```

```
Or for `yum`/`dnf`:
```

```
    sudo yum install -y kubelet-1.36.0 kubectl-1.36.0
    sudo systemctl restart kubelet
```

1. **Drain and Uncordon Worker Nodes:** For each worker node, drain it, upgrade `kubelet` and `kubectl`, then uncordon.

```
    # On control plane node:
    kubectl drain <node-name> --ignore-daemonsets

    # On worker node:
    sudo apt-get update
    sudo apt-get install -y kubelet=1.36.0-00 kubectl=1.36.0-00
    sudo systemctl restart kubelet

    # On control plane node:
```

```
kubectl uncordon <node-name>
```

1. **Post-Upgrade Verification:** Verify the cluster health, check logs, and ensure all critical workloads are running as expected. Pay close attention to any pods that previously used `gitRepo` volumes.

Ecosystem Impact

Kubernetes v1.36's focus on enhanced security and AI workload support will ripple through the cloud-native ecosystem, prompting updates and new integrations.

Security Tooling and Policy Engines

The GA of `MutatingAdmissionPolicy` will likely lead to an explosion of new policies and integrations within existing policy engines like Kyverno, OPA Gatekeeper, and Falco. These tools will leverage the default-enabled status to offer more sophisticated and automated security enforcement. Cloud providers will also update their managed Kubernetes services to reflect these new default security postures.

AI/ML Platforms and MLOps Tools

The `OCI volumes support` and `HPA scale-to-zero` capabilities are significant for the AI/ML community. MLOps platforms and data science tooling will likely integrate these features to streamline model deployment, data versioning, and cost optimization for inference services. Expect to see updates from vendors like Kubeflow, MLflow, and various cloud-specific AI services to leverage these improvements.

Observability and Monitoring

The promotion of `Pressure Stall Information (PSI)` metrics to GA will drive new features in monitoring and observability platforms. Tools like Prometheus, Grafana, Datadog, and New Relic will enhance their dashboards and alerting capabilities to expose and interpret PSI data, providing more actionable insights into resource contention.

Cloud Providers

Cloud providers offering managed Kubernetes services (EKS, AKS, GKE, etc.) will rapidly adopt v1.36. Users should expect new versions of their managed clusters to become available, incorporating the default security enhancements and new features. It's crucial to check their specific release notes for any provider-specific considerations or additional breaking changes.

The continuous evolution of Kubernetes, with releases like v1.36, underscores the project's commitment to security, scalability, and adaptability to emerging workloads like AI. Staying current with these releases is not just about gaining new features, but about maintaining a secure, efficient, and future-proof cloud-native infrastructure.