

What's new in Kubernetes 1.33?

Mongkol Thongkraikaew

Head of Platform Engineering

Ascend Money

About Me

Mongkol Thongkraikaew

Head of Platform Engineering @ Ascend Money

AWS Community Builder 2025 (Container)

Exp : 8+ years with container technology

Open Source Lovers , Cloud-Native Lovers , Beef Lovers ,
Beer Lovers

Tech Blogger



: Cloudation 101



: Mongkol Thongkraikaew



Medium : @mongkol.ttm



Agenda

Kubernetes v1.33: Octarine

Enhancements in Kubernetes v1.33: Highlights

Deprecations & Removals Features

Demo





Kubernetes v1.33: Octarine



Kubernetes 1.33 : Octarine

Released **Wednesday, 23rd April, 2025**


The theme for Kubernetes v1.33 is
Octarine: The Color of Magic ✨

Inspired by Terry Pratchett's *Discworld* series.

This release highlights the open source magic that Kubernetes enables across the ecosystem



Summary of Enhancement

64 stars  in the background = 64 KEPs (Kubernetes Enhancement Proposals)

18 enhancement to **Stable**

20 enhancement to **Beta**

24 enhancement to **Alpha**

2 deprecated / withdrawn

Complete Lists :

<http://bit.ly/k8s133-enhancements>





Enhancement Highlights



Highlight #1 : Sidecar containers ([KEP-753](#))

Sidecar containers in Kubernetes are like little helper containers that live inside the same Pod as your main application. They're there to add extra features like logging, monitoring, and security, without you having to mess with your primary application's code. This feature is enabled by default from Kubernetes version 1.29 onwards.

Kubernetes Implements sidecars as a special class of init container, with `restartPolicy: Always`, to make sure the sidecars start before the main container and keep running, or automatically stop after the main container exits.

Status:

Alpha : v1.28

Beta : v1.29

Stable : **v1.33**

SIG-Node



Highlight #2 : nftables backend for kube-proxy

([KEP-3866](#))

Kube-proxy on Linux currently uses *iptables*, which has been the main packet filtering system but is now outdated and largely unmaintained. Development has shifted to *nftables*, which offers better features and performance.

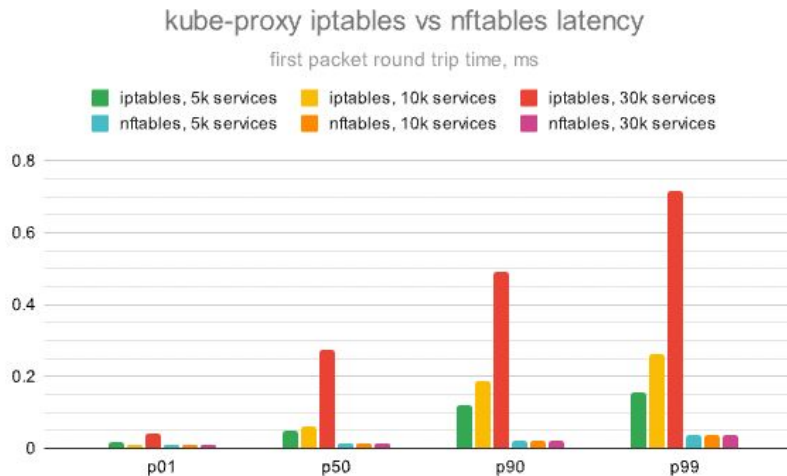
This KEP proposes a new official *nftables* backend for kube-proxy, aiming to eventually replace both *iptables* and *ipvs* backends and become the default Linux mode.

Status:

Alpha : v1.29

Beta : v1.31

Stable : **v1.33**



Blog Post : [NFTables mode for kube-proxy](#)

SIG-Network



Highlight #3 : Multiple Service CIDRs ([KEP-1880](#))

Enable dynamic scaling of Service IP addresses by introducing a new allocation mechanism. This involves two new API objects: `ServiceCIDR` and `IPAddress`. Users can then increase the number of available Service IPs on the fly by creating more `ServiceCIDRs`.

And also allow for renumbering Service IPs, and provide ways to find out the current Service IP range and which IP addresses are already allocated.

Status:

Alpha : v1.29

Beta : v1.31

Stable : **v1.33**

```
~ > kubectl get servicecidr
NAME          CIDRS
kubernetes    10.96.0.0/16
```

```
~ > kubectl get ipaddress
NAME          PARENTREF
10.96.0.1     services/default/kubernetes
10.96.0.10    services/kube-system/kube-dns
10.96.177.35  services/default/example-service
```

SIG-Network



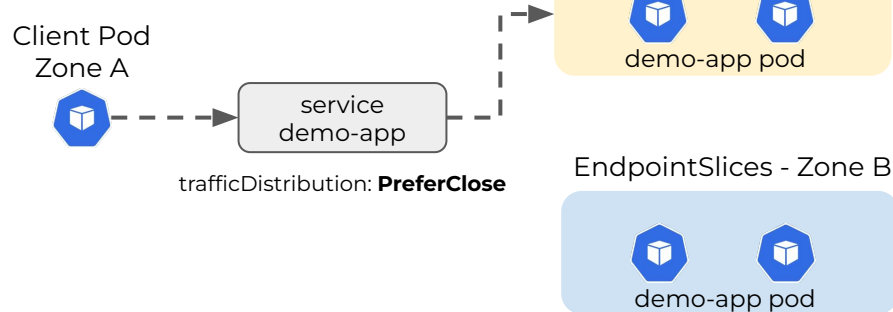
Highlight #4 : Traffic Distribution for Services

(KEP-4444)

This release graduate topology-aware routing and traffic distribution to GA, which would allow to optimize service traffic in multi-zone cluster. *trafficDistribution* field is added to service spec

```
apiVersion: v1
kind: Service
metadata:
  name: demo-svc
spec:
  type: ClusterIP
  trafficDistribution: PreferClose
  ports:
    - name: tcp
      port: 80
      protocol: TCP
      targetPort: 8080
  selector:
    app: demo-app
```

PreferClose, Prioritize routing traffic to endpoints in the same zone as the client. If no endpoints are available within the zone, traffic will be routed to other zones.



Status:

Alpha : v1.30

Beta : v1.31

Stable : **v1.33**

SIG-Network



Highlight #5: MatchLabelKeys to Pod Affinity and Pod Anti Affinity ([KEP-3633](#))

The scheduler currently treats *Terminating* Pods as active, causing topology spreading imbalances during rollouts.

The stable ***matchLabelKeys*** and ***mismatchLabelKeys*** fields in Pod affinity allow finer control over Pod co-location and separation. These complement *labelSelector* to improve scheduling flexibility for rolling updates and service isolation based on global settings.

Status:

Alpha : v1.29

Beta : v1.30

Stable : **v1.33**

Example Deployment manifest

```
...
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
            - key: app
              operator: In
              values:
                - database
          topologyKey: topology.kubernetes.io/zone
        matchLabelKeys: # ADDED
      - pod-template-hash
```

[pod-template-hash](#)

[Issue #105661](#)

SIG-Scheduling



Highlight #6: Recursive read-only (RRO) mounts

([KEP-3857](#))

The current `readOnly` volumes aren't recursively read-only, which could lead to data being compromised; for example, even if `/mnt` is mounted as read-only, its submounts like `/mnt/usbstorage` aren't read-only. This KEP aims to make read-only volumes recursively read-only.

Pod manifest

```
spec:
  volumes:
    - name: foo
      hostPath:
        path: /mnt
        type: Directory
  containers:
    - volumeMounts:
        - mountPath: /mnt
          name: foo
          mountPropagation: None
          readOnly: true
      # NEW
      recursiveReadOnly: IfPossible
```

Status:

Alpha : v1.30

Beta : v1.31

Stable : **v1.33**

Requires kernel ≥ 5.12 , with one of the following OCI runtimes:

- `runc` ≥ 1.1
- `crun` ≥ 1.4

SIG-Node

Add `RecursiveReadOnly` to the VolumeMount struct.

Available status: `Disabled` | `IfPossible` | `Enabled`



Highlight #7 : Subresource support in kubectl

([KEP-2590](#))

The `--subresource` argument is now generally available (GA) for kubectl subcommand such as `get`, `patch`, `edit`, `apply` and `replace`, allowing user to fetch and update subresources for all resource that support them.

Status:

Alpha : v1.24

Beta : v1.27

Stable : **v1.33**

```
$ kubectl get deployment example-deployment --subresource=status
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
example-deployment	1/1	1	1	26h

```
$ kubectl get deployment example-deployment --subresource=scale
```

NAME	DESIRED	AVAILABLE	AGE
example-deployment	1	1	26h

```
#resize resource cpu limit of pod
```

```
$ kubectl patch pod example-pod --subresource=resize --patch \
  '{"spec":{"containers":[{"name":"pause", "resources":{"requests": {"limits":{"cpu":"800m"}}}]}}'
pod/example-pod patched
```

[kubectl reference](#)

SIG-CLI



Highlight #8 : Zero-second sleeps for container PreStop hooks ([KEP-4818](#))

Creating a container with a PreStop lifecycle hook with sleep of **0** seconds will throw a validation error like so:

```
Invalid value: 0: must be greater than 0 and less than terminationGracePeriodSeconds (30)
```

Status:

Alpha : v1.32

Beta : **v1.33**

This KEP aims to add support for setting a value of zero with the sleep action of the PreStop hook to do a no-op. which is useful when a preStop hook is required but no delay is desired.

example sleep with shell command

```
...
spec:
  containers:
    - name: nginx
      image: nginx:1.16.1
      lifecycle:
        preStop:
          exec:
            command:
              ['sh', '-c', 'sleep 10s']
```

pod manifest

```
...
spec:
  containers:
    - name: nginx
      image: nginx:1.16.1
      lifecycle:
        preStop:
          sleep:
            seconds: 0
```

Feature gate name:

[PodLifecycleSleepActionAllowZero](#)

this feature gate which is disabled by default.

SIG-Node



Highlight #9 : Custom container stop signals

([KEP-4960](#))

Kubernetes by default sends a SIGTERM to all containers while killing them. This KEP proposes to add support to configure custom stop signals for containers from the *Pod spec*.

Currently you can only configure this by defining `STOPSIGNAL` in the container image definition file before you build the image.

Status:

Alpha : **v1.33**

Example Dockerfile

```
FROM nginx:alpine

# Set the stop signal to SIGQUIT, which nginx
uses for graceful shutdown
STOPSIGNAL SIGQUIT

# COPY my-nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
```

Pod manifest

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      image: nginx:custom-1.0.1
      lifecycle:
        stopSignal: SIGUSR1
```

#Override **STOPSIGNAL** of image

Feature gate name:
[ContainerStopSignals](#)

this feature gate which is disabled by default.

SIG-Node



Highlight #10 : Option for kubectl with .kuberc for user preferences ([KEP-3104](#))

kuberc is a configuration file that kubectl can now read on startup to apply **user-defined preferences**. The primary goal is to separate cluster credentials and server configuration from user preferences, which is primarily for cluster connection details and authentication.

Status:
Alpha : **v1.33**

Example kuberc configure

```
apiVersion: kubectl.config.k8s.io/v1alpha1
kind: Preference
aliases:
- name: kgp
  command: get
  appendArgs:
  - pods

# Force the --interactive=true flag for kubectl delete
overrides:
- command: delete
  flags:
  - name: interactive
    default: "true"
```

Example scripts

```
# Enable kuberc
# default path : ~/.kube/kuberc
$ export KUBECTL_KUBERC=true

$ kubectl kgp -n default
NAME          READY  STATUS   RESTARTS  AGE
example-pod   1/1    Running   0          83m

$ kubectl delete pod example-pod
You are about to delete the following 1 resource(s):
pod/example-pod
Do you want to continue? (y/n):
```

SIG-CLI



Highlight #11 : In-place resource resize for vertical scaling of Pods ([KEP-1287](#))

Before this enhancement, container resources defined in a Pod's spec were **immutable**, and updating any of these details within a Pod template would trigger Pod replacement (required restarting the Pod).

In-place Pod resizing allows you to change the CPU and memory requests and limits of running Pod **without any downtime**, seamless scale-down when the traffic is low, and even allocating larger resources during startup, which can then be reduced once the initial setup is complete.

```
# Try resize pod via kubectl

$ kubectl edit pod <pod-name> --subresource resize

# requires kubectl v1.32+
```

Status:

Alpha : v1.27
Beta : **v1.33**

Feature gate name:

[InPlacePodVerticalScaling](#)

In v1.33 is **enabled** by default.

SIG-Node
SIG-Autoscaling

Blog Post : [In-Place Pod Resize Graduated to Beta](#)



Deprecations & Removals



Deprecations & Removals

#1 Deprecation of stable Endpoint API ([KEP-4974](#))

SIG-Network

Since EndpointSlices became GA in 1.21, which effectively replaced the Endpoint API to address some limitations. The EndpointSlices API has introduced new features such as [dual-stack networking](#) or [traffic distribution](#), That make the original Endpoint API can official deprecation.

Goal

- Officially declare v1.Endpoints to be deprecated.
- Add **warnings** to alert users of the fact that Endpoints is deprecated.
- Ensure that all core Kubernetes code uses EndpointSlices rather than Endpoints
- etc.

```
$ kubectl get endpoints myservice
Warning: v1 Endpoints is deprecated in v1.33+; use discovery.k8s.io/v1 EndpointSlice
NAME           ENDPOINTS           AGE
myservice      10.180.3.17:443     1h

$ kubectl get endpointslice -l kubernetes.io/service-name=myservice
NAME           ADDRESSTYPE  PORTS  ENDPOINTS           AGE
myservice-7vzhx  IPv4         443    10.180.3.17         21s
myservice-jcv8s  IPv6         443    2001:db8:0123::5    21s
```

Blog Post : [Continuing the transition from Endpoints to EndpointSlices](#)



Deprecations & Removals

#2 Remove Kube-Proxy version information in node status ([KEP-4004](#))

Following its deprecation in v1.31 `.status.nodeInfo.kubeProxyVersion` field for node, its was removed in v1.33.

This field was set by kubelet, which does not actually know the kube-proxy version, or even if kube-proxy is running.

SIG-Node

#3 Removal of host network support for Windows pods ([KEP-3503](#))

The original feature was introduced as alpha in v1.26, but due to unexpected behavior with containerd, the Kubernetes project decided to withdraw the related KEP. Support was completely removed in v1.33.

SIG-Windows



Title	...	Stage	...	Title	...	Stage	...	
1 Fine grained Kubelet API authorization #2862		Beta		26 Separate kubect! user preferences from cluster configs #3104		Alpha		51 CSI Differential Snapshot for Block Volumes #3314
2 Take taints/tolerations into consideration when calculating PodTopolog... #3094		Stable		27 Consider Terminating Pods in Deployments #3973		Alpha		52 DRA: Resource Claim Status with possible standardized network inte
3 ClusterTrustBundles (previously Trust Anchor Sets) #3257		Beta		28 Speed up recursive SELinux label change #1710		Beta		53 Split L3 Cache Topology Awareness in CPU Manager #5109
4 Introduce MatchLabelKeys to Pod Affinity and Pod Anti Affinity #3633		Stable		29 Add subresource support to kubect! #2590		Stable		54 DRA: Partitionable Devices #4815
5 Bound service account token improvements #4193		Stable		30 Backoff Limit Per Index For Indexed Jobs #3850		Stable		55 DRA: device taints and tolerations #5055
6 Projected service account tokens for Kubelet image credential providers #4412		Alpha		31 Job success/completion policy #3998		Stable		56 PreferSameNode Traffic Distribution (formerly PreferLocal traffic po
7 Add CPUManager policy option to restrict reservedSystemCPUs to sys... #4540		Beta		32 Declarative Validation Of Kubernetes Native Types With validation-gen #5073		Beta		57 Pop pod from backoffQ when activeQ is empty #5142
8 DRA: Prioritized Alternatives in Device Requests #4816		Alpha		33 Configurable tolerance for Horizontal Pod Autoscalers #4951		Alpha		58 Traffic Distribution for Services #4444
9 Add storage capacity scoring #4049		Alpha		34 Sidecar Containers #753		Stable		59 Recursive Read-only (RRO) mounts #3857
10 Asynchronous preemption in the scheduler #4832		Beta		35 Coordinated Leader Election #4355		Beta		60 Allow zero value for Sleep Action of PreStop Hook #4818
11 Generic data populators #1495		Stable		36 Ordered namespace deletion #5080		Alpha		61 Fine-grained SupplementalGroups control #3619
12 Multiple Service CIDRs #1880		Stable		37 CRD Validation Ratcheting #4008		Stable		62 Expose Node labels via downward API #4742
13 Topology Aware Routing #2433		Stable		38 Host network support for Windows pods #3503		Deprecatio...		63 Streaming Encoding for LIST Responses #5116
14 Portworx file in-tree to CSI driver migration #2589		Stable		39 node: cpumanager: add options to reject non SMT-aligned workload #2625		Stable		64 Snapshottable API server cache #4988
15 Always Honor PersistentVolume Reclaim Policy #2644		Stable		40 Support PSI based on cgroupv2 #4205		Alpha		
16 nftables kube-proxy backend #3866		Stable		41 DRA: structured parameters #4381		Beta		
17 Deprecate status.nodeInfo.kubeProxyVersion field #4004		Stable		42 VolumeSource: OCI Artifact and/or image #4639		Beta		
18 Relaxed DNS search string validation #4427		Beta		43 Pod Generation #5067		Alpha		
19 IP/CIDR validation improvements #4858		Alpha		44 In-Place Update of Pod Resources #1287		Beta		
20 Mutable CSINode Allocatable Property #4876		Alpha		45 Add CPUManager policy option to distribute CPUs across NUMA node... #2902		Beta		
21 Deprecate v1.Endpoints #4974		Deprecatio...		46 add ProcMount option #4265		Beta		
22 Remove gitRepo volume driver #5040		Alpha		47 Tune CrashLoopBackOff #4603		Alpha		
23 API for external signing of Service Account tokens #740		Alpha		48 Container Stop Signals #4960		Alpha		
24 Ensure secret pulled images #2535		Alpha		49 Support for Direct Service Return (DSR) and overlay networking in Win... #5100		Beta		
25 DRA: AdminAccess for ResourceClaims and ResourceClaimTemplates #5018		Alpha		50 Support User Namespaces in pods #127		Beta		

Complete Lists :
<http://bit.ly/k8s133-enhancements>





Demo



More Information

[Kubernetes v1.33: Octarine](#)

[Kubernetes 1.33 Release Information](#)

[Kubernetes 1.33 Enhancements Tracking](#)



Thanks you

Mongkol Thongkraikaew

Head of Platform Engineering
Ascend Money