What's New in Kubernetes 1.16

Presenters

Kenny Coleman

1.16 Enhancements Lead

@kendrickcoleman

Lachlan Evenson

1.16 Release Lead

@LachlanEvenson

Taylor Dolezal

1.16 Communications Lead / Moderator

@onlydole

Agenda

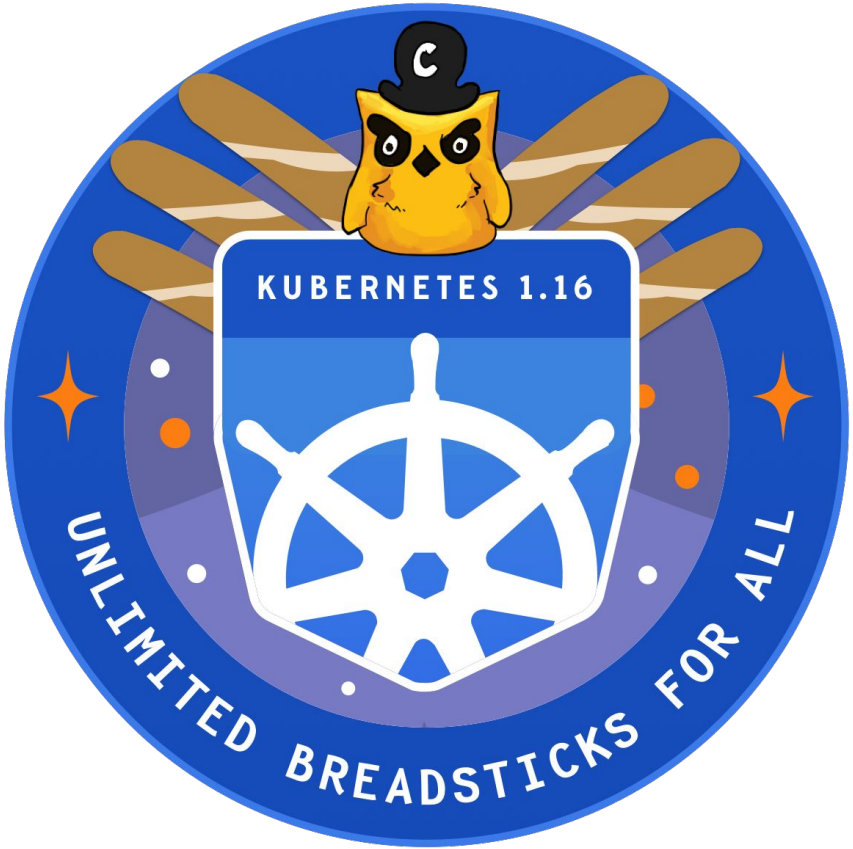
Major Feature: CRDs Moving to Stable

Major Feature: IPv4/IPv6 Dual Stack Support

Major Feature: Extend allowed PVC DataSources aka Volume Cloning

1.16 Enhancements Overview

Q&A



1.16 Enhancements

CLOUD NATIVE
COMPUTING FOUNDATION

Overview

- 31 total enhancements tracked in 1.16
  - 8 Stable Enhancements
  - 8 Graduating to Beta
  - 15 Introduced Alpha features

Highlights

CLOUD NATIVE
COMPUTING FOUNDATION

CRDs Moving to Stable

- To Follow Next in SIG API Machinery

IPv4/IPv6 Dual Stack Support

- Alpha in 1.16
- IPv6 support was added to Kubernetes in 1.9 but it requires IPv6 throughout. Clusters can could run in either IPv4-only, IPv6-only, or in a "single-pod-IP-aware" dual-stack configuration.
- Now in 1.16:
  - Awareness of multiple IPv4/IPv6 address assignments per pod
  - Native IPv4-to-IPv4 in parallel with IPv6-to-IPv6 communications to, from, and within a cluster
  - Functionality tested with the Bridge CNI plugin, PTP CNI plugin, and Host-Local IPAM plugins as references

- https://github.com/kubernetes/enhancements/issues/563

Extend allowed PVC DataSources aka Volume Cloning

- Beta in 1.16

- Adding support for specifying existing PVCs in the DataSource field to indicate a user would like to Clone a Volume

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
    name: pvc-2
    namespace: myns
spec:
  capacity:
    storage: 10Gi
  dataSource:
    kind: PersistentVolumeClaim
    name: pvc-1
```

- https://github.com/kubernetes/enhancements/issues/989

API MACHINERY

CLOUD NATIVE
COMPUTING FOUNDATION

CustomResourceDefinitions

- Graduating to Stable

- CustomResourceDefinitions (CRDs) are the way to extend the Kubernetes API to include custom resource types that behave like the native resource types. CRDs have been in Beta since Kubernetes 1.7

- https://github.com/kubernetes/enhancements/issues/95

Publish CRD OpenAPI Schema

- Graduated to Stable

- Publishing CRD OpenAPI enables client-side validation, schema explanation and client generation for CRs. It covers the gap between CR and native Kubernetes APIs, which already support OpenAPI documentation.
- For every CRD served, publish Paths (operations that we support on resource and subresources) and Definitions (for both CR object and CR list object) in OpenAPI documentation to fully demonstrate the existence of the API.
- For CRDs with schema defined, the CR object Definition should include both CRD schema and native Kubernetes ObjectMeta and TypeMeta properties.
- For CRDs without schema, the CR object definition will be as complete as possible while still maintaining compatibility with the openapi spec and with supported kubernetes components

- https://github.com/kubernetes/enhancements/issues/692

Subresources for Custom Resources

- Graduated to Stable

- Adds `/status` and `/scale` subresources for CustomResources.

- https://github.com/kubernetes/enhancements/issues/571

Defaulting and Pruning for Custom Resources

- Pruning is Stable / Defaulting is Beta in 1.16

- Defaulting is implemented for most native Kubernetes API types and plays a crucial role for API compatibility when adding new fields. CustomResources do not support this natively.
- This adds support for specifying default values for fields via OpenAPI v3 validation schemas in the CRD manifest.

```
properties:
  foo:
    type: string
    default: "abc"
```

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
spec:
  preserveUnknownFields: false
  ...
```

- CustomResources store arbitrary JSON data without following the typical Kubernetes API behaviour to prune unknown fields. This makes CRDs different, but also leads to security and general data consistency concerns because it is unclear what is actually stored in etcd.
- This will add pruning of all fields which are not specified in the OpenAPI validation schemas given in the CRD.

- https://github.com/kubernetes/enhancements/issues/575

Webhook Conversion for Custom Resource Definitions

- Graduated to Stable

- Support for version-conversion of Kubernetes resources defined via Custom Resource Definitions (CRD)

- CRD users want to be certain they can evolve their API before they start down the path of developing a CRD + controller

- CRD supports multiple version but no conversion between them (something called nopConverter which only change the apiVersion of the CR). With this proposal, it introduced a conversion mechanism for CRDs based on an external webhook. Detail API changes, use cases and upgrade/downgrade scenarios are discussed.

- https://github.com/kubernetes/enhancements/issues/598

Admission Webhooks

- Graduated to Stable

- Admission webhook is a way to extend kubernetes by putting hook on object creation/modification/deletion. Admission webhooks can mutate or validate the object.

- Extended to single objects

- https://github.com/kubernetes/enhancements/issues/492

Add Watch Bookmarks support

- Graduated to Beta

- Make restarting watches cheaper from kube-apiserver performance perspective.
- Different scalability tests observed that restarting watches may cause significant load on kube-apiserver when watcher is observing a small percentage of changes (due to field or label selector). In extreme cases, reestablishing such watcher may even lead to falling out of history window and "resource version too old" errors
- Reduce load on apiserver by minimizing amount of unnecessary watch events that need to be processed after restarting a watch.
- Reduce amount of undesired "resource version too old" errors on reestablishing a watch.
- A new type of watch event called Bookmark. Watch event with type Bookmark will represent information that all the objects up to a given resourceVersion has been processed for a given watcher.

- https://github.com/kubernetes/enhancements/issues/956

Server-Side Apply

- Graduated to Beta

`kubectly apply` will move to the control plane

- Example problems today:
  - User does POST, then changes something and applies: surprise!
  - User does an apply, then kubectl edit, then applies again: surprise!
  - User does GET, edits locally, then apply: surprise!
  - User tweaks some annotations, then applies: surprise!
- "Apply" is intended to allow users and systems to cooperatively determine the desired state of an object
  - Be robust to changes made by other users, systems, defaulters, and object schema evolution.
  - Be agnostic about prior steps in a CI/CD system (and not require such a system).
  - Have low cognitive burden

kubectl apply.....--server-side

- https://github.com/kubernetes/enhancements/issues/555

Deprecate and Remove SelfLink

- Net New Alpha

- `SelfLink` is a URL representing a given object. It is part of `ObjectMeta` and `ListMeta` which means that it is part of every single Kubernetes object.
- This field will be deprecated and removed in 1 year following the Deprecation Policy

- https://github.com/kubernetes/enhancements/issues/1164

CLOUD PROVIDER

CLOUD NATIVE
COMPUTING FOUNDATION

Building Kubernetes Without In-Tree Cloud Providers

- Net New Alpha

- The in tree cloud-provider implementations are being removed which involves a large amount of code that is used in many places in tree. In order to prepare for this, a build has to available to determine what removal entails and verify that Kubernetes will continue to function correctly.

- https://github.com/kubernetes/enhancements/issues/1179

CLUSTER LIFECYCLE

Kubeadm for Windows

- Net New Alpha

- A tool to allow users to take a Windows machine and join it to an existing Kubernetes cluster with a single command. The user should also be able to reset the node with a single command. Install prerequisites and enable kubeadm to bring the Windows node to a Ready state

- Only proposes enablement of support for Windows worker nodes using kubeadm.

- https://github.com/kubernetes/enhancements/issues/995

Advanced configurations with kubeadm (using Kustomize)

- Net New Alpha

- A new kubeadm feature that will allow users to bootstrap a Kubernetes cluster with static pods customizations not supported by the Kubeadm component configuration.

- https://github.com/kubernetes/enhancements/issues/1177

INSTRUMENTATION

CLOUD NATIVE
COMPUTING FOUNDATION

Kubernetes Metrics Overhaul

- Net New Alpha

- A number of metrics that Kubernetes is instrumented with do not follow the official Kubernetes instrumentation guidelines. In order to have consistently named and high quality metrics, this effort aims to make working with metrics exposed by Kubernetes consistent with the rest of the ecosystem
  - cAdvisor instrumentation changes
  - Changing API latency histogram buckets
  - Kubelet metric changes
  - Kube-scheduler, proxy, and api-server metric changes
  - Client-go metric changes

- https://github.com/kubernetes/enhancements/issues/1206

NETWORK

Finalizer Protection for Service LoadBalancers

- Graduated to Beta

- Finalizer protection to ensure the Service resource is not fully deleted until the correlating load balancer resources are deleted.
- Any service that has type=LoadBalancer (both existing and newly created ones) will be attached a service LoadBalancer finalizer, which should be removed by service controller upon the cleanup of related load balancer resources

- https://github.com/kubernetes/enhancements/issues/980

EndpointSlice API

- Net New Alpha

- The current Core/V1 Endpoints API comes with severe performance/scalability drawbacks affecting multiple components in the control-plane. A new EndpointSlice API aiming to replace Core/V1 Endpoints API for most internal consumers, including kube-proxy. The new EndpointSlice API aims to address existing problems as well as leaving room for future extension.

- https://github.com/kubernetes/enhancements/issues/752

NODE

CLOUD NATIVE
COMPUTING FOUNDATION

Ephemeral Containers

- Net New Alpha

A mechanism to run a container with a temporary duration that executes within namespaces of an existing pod. Ephemeral Containers are initiated by a user and intended to observe the state of other pods and containers for troubleshooting and debugging purposes.

- https://github.com/kubernetes/enhancements/issues/277

Pod Overhead

- Net New Alpha

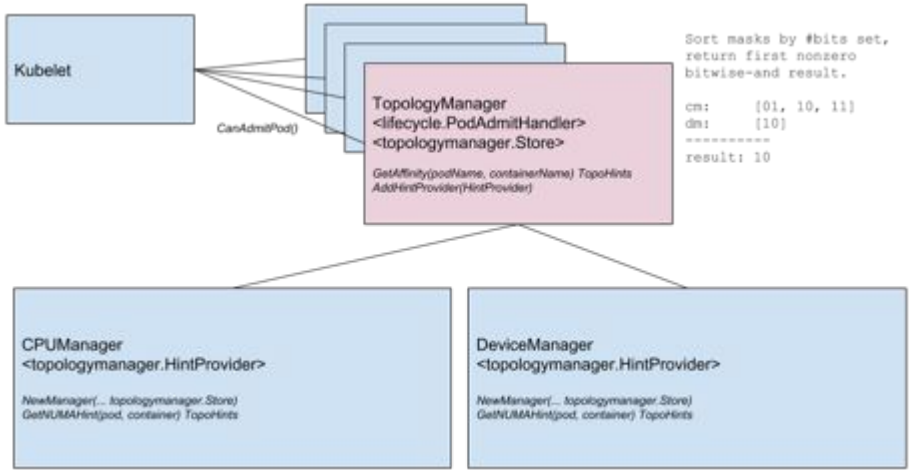
- Sandbox runtimes introduce a non-negligible overhead at the pod level which must be accounted for effective scheduling, resource quota management, and constraining

- https://github.com/kubernetes/enhancements/issues/688

Node Topology Manager

- Net New Alpha

- An increasing number of systems leverage a combination of CPUs and hardware accelerators to support latency-critical execution and high-throughput parallel computation.
- CPU isolation and memory and device locality are required. However, in Kubernetes, these optimizations are handled by a disjoint set of components.
- This proposal provides a mechanism to coordinate hardware resource assignments for different components in Kubernetes.



- https://github.com/kubernetes/enhancements/issues/693

Add pod-startup liveness-probe holdoff for slow-starting pods

- Net New Alpha

- Slow starting containers are difficult to address with the current status of health probes: they are either killed before being up, or could be left deadlocked during a very long time before being killed.

- This proposal adds a new probe called startupProbe that holds off all the other probes until the pod has finished its startup. In the case of a slow-starting pod, it could poll on a relatively short period with a high failureThreshold. Once it is satisfied, the other probes can start..

```
ports:
- name: liveness-port
  containerPort: 8080
  hostPort: 8080

livenessProbe:
  httpGet:
    path: /healthz
    port: liveness-port
  failureThreshold: 1
  periodSeconds: 10

startupProbe:
  httpGet:
    path: /healthz
    port: liveness-port
  failureThreshold: 30 (=initializationFailureThreshold)
  periodSeconds: 10
```

- https://github.com/kubernetes/enhancements/issues/950

RuntimeClass Scheduling

- Beta

- RuntimeClass scheduling enables native support for heterogeneous clusters where every node does not necessarily support every RuntimeClass. This feature allows pod authors to select a RuntimeClass without needing to worry about cluster topology.
  - Ie. Windows or Sandbox Workloads

- https://github.com/kubernetes/enhancements/issues/894

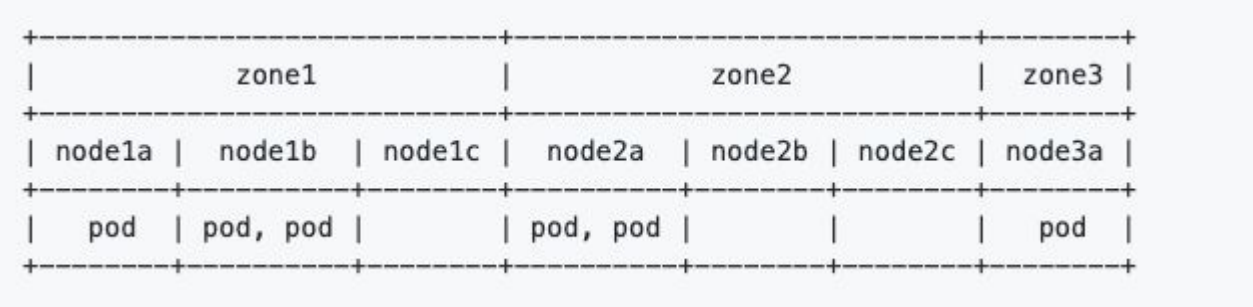
SCHEDULING

CLOUD NATIVE
COMPUTING FOUNDATION

Even Pods Spreading

- Net New Alpha

- EvenPodsSpreading feature gives users more fine-grained control on distribution of pods scheduling, so as to achieve better high availability and resource utilization

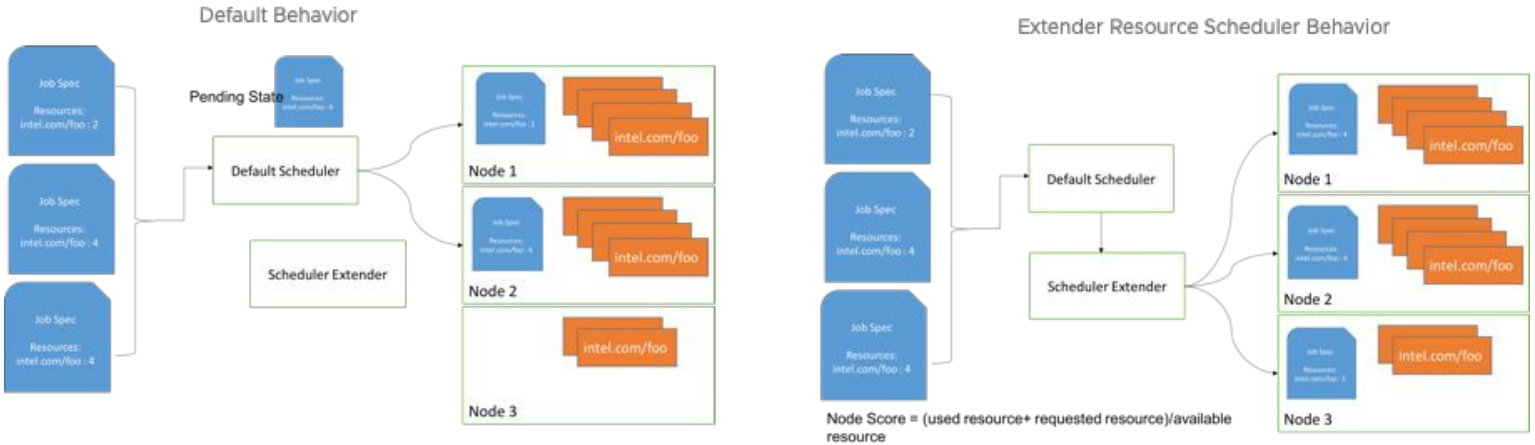


- https://github.com/kubernetes/enhancements/issues/894

ExtendingRequestedToCapacityRatio Priority Function to support Resource Bin Packing of Extended Resources

- Net New Alpha

- Extend RequestedToCapacityRatio Priority Function to allow users to use the best fit polices during scheduling. It will allow users to apply bin packing on core resources like CPU, Memory as well as extended resources like accelerators.



- https://github.com/kubernetes/enhancements/issues/964

STORAGE

CLOUD NATIVE
COMPUTING FOUNDATION

Add resizing support to CSI volumes

- Graduated to Beta

- expansion of CSI volumes used by PersistentVolumeClaims that support volume expansion as a plugin capability.
- To support resizing of CSI volumes an external resize controller will monitor all PVCs. If a PVC meets following criteria for resizing, it will be added to controller's work queue:
  - The driver name disovered from PVC should match name of driver currently known(by querying driver info via CSI RPC call) to external resize controller.
  - Once it notices a PVC has been updated and by comparing old and new PVC object, it determines more space has been requested by the user.
- Once PVC gets picked from workqueue, the controller will also compare requested PVC size with actual size of volume in PersistentVolume object. Once PVC passes all these checks, a CSI ControllerExpandVolume call will be made by the controller if CSI plugin implements ControllerExpandVolume RPC call.

- https://github.com/kubernetes/enhancements/issues/556

CSI Inline Volume Support

- Graduated to Beta

- Define API and high level design for in-line CSI volumes in Pod.
- Make in-line CSI volumes secure for using ephemeral volumes (such as Secrets or ConfigMap).
- Currently, CSI can be used only through PersistentVolume object. CSI drivers can be used to provide ephemeral volumes used to inject state, configuration, secrets, identity or similar information to pods, like Secrets and ConfigMap in-tree volumes do today. We don't want to force users to create PVs for each such volume, we should allow to use them in-line in pods as regular Secrets or ephemeral Flex volumes.

- https://github.com/kubernetes/enhancements/issues/596

Support for CSI Plugins on Windows Nodes

- Net New Alpha

- Kubernetes operators will be able to leverage modern CSI plugins to satisfy the persistent storage requirements of Windows workloads in Kubernetes.

- https://github.com/kubernetes/enhancements/issues/1122

WINDOWS

CLOUD NATIVE
COMPUTING FOUNDATION

Support GMSA for Windows workloads

- Beta

- Group Managed Service Accounts (GMSA) for Windows containers in Kubernetes. GMSA are a specific type of Active Directory account that provides automatic password management, simplified service principal name (SPN) management, and the ability to delegate the management to other administrators across multiple servers

- https://github.com/kubernetes/enhancements/issues/689

RunAsUserName for Windows

- Net New Alpha
- Windows specific options in Pod Security Context and Container Security Context
- The enhancements will cover fields pertinent to GMSA credential specs and the username with which to execute the container entry-point
- Not inclusive of GMSA functionality but only to support it later

- https://github.com/kubernetes/enhancements/issues/1043

What's coming next?

- Already 4 weeks into 1.17
  - Enhancements freeze has passed - Oct 15th
  - http://bit.ly/k8s117-enhancement-tracking
- Targeted GA is December 9th

Questions?

CLOUD NATIVE
COMPUTING FOUNDATION

Thank You

CLOUD NATIVE
COMPUTING FOUNDATION