What's New in Kubernetes 1.15

Presenters

Kenny Coleman

1.15 Enhancements Lead

Kim McMahon

Moderator

Agenda

Major Feature: Dynamic HA Clusters with kubeadm

Major Feature: Volume Cloning

Major Feature: CRDs!

1.15 Enhancements Overview

Q&A

1.15 Enhancements

Overview

- 25 total enhancements tracked in 1.15
  - 2 Stable Enhancements
  - 13 Graduating to Beta
  - 10 Introduced Alpha features

Highlights

CLOUD NATIVE
COMPUTING FOUNDATION

Ability to create dynamic HA clusters with kubeadm

- Graduated to Beta in 1.15

- Dynamic HA clusters can easily be created with the kubeadm tool using exactly the same `kubeadm init` and `kubeadm join` commands the users are familiar with, the only difference that you have to pass the `--control-plane` flag.

- `sudo kubeadm init --config=kubeadm-config.yaml --upload-certs`

- https://github.com/kubernetes/enhancements/issues/357

Extend allowed PVC DataSources aka Volume Cloning

- Net New Alpha 1.15

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

```
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

CRD Mania!

- -- To Follow Next in SIG API Machinery

API MACHINERY

CLOUD NATIVE
COMPUTING FOUNDATION

Admission Webhooks

- Updated Changes to continue Beta work in 1.15

- 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

Defaulting and Pruning for Custom Resources

- Net New Alpha in 1.15
- 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"
```

- 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.

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

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

Webhook Conversion for Custom Resource Definitions

- Graduated to Beta

- 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

Publish CRD OpenAPI Schema

- Graduated to Beta

- 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

Add Watch Bookmarks support

- Net New Alpha in 1.15

- 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

APPS

CLOUD NATIVE
COMPUTING FOUNDATION

PDB support for custom resources with scale subresource

- Graduated to Beta

- Pod Disruption Budget (PDB) is an important tool to control the number of voluntary disruptions for workloads on Kubernetes.
- As more users start deploying custom controllers/operators based on CRDs (EtcdCluster, MySQLReplicaSet...), it is inconvenient that they cannot take advantage of PDBs. This doesn't work today because the PDB controller needs to know the desired number of replicas specified in a controller and the PDB controller only knows how to find this from the four Kubernetes workload controllers mentioned above.
- Use the scale subresource to allow setting PDBs on any resource that implements the scale subresource.

- https://github.com/kubernetes/enhancements/issues/981

ARCHITECTURE

Add go module support to k8s.io/kubernetes

- Net New Stable

- Manage the vendor folder in kubernetes/kubernetes using go modules, and define go.mod module files for published components like k8s.io/client-go and k8s.io/api.

- In addition to simply keeping up with the go ecosystem, go modules provide many benefits:
  a. rebuilding vendor with go modules provided a 10x speed increase over Godep in preliminary tests
  b. go modules can reproduce a consistent vendor directory on any OS
  c. if semantic import versioning is adopted, consumers of Kubernetes modules can use two distinct versions simultaneously (if required by diamond dependencies)

- https://github.com/kubernetes/enhancements/issues/917

CLI

CLOUD NATIVE
COMPUTING FOUNDATION

kubectl get and describe should work well with extensions

- Graduated to Stable in 1.15

- Server-side Get and Partial Objects to GA. Being feature complete will begin the removal of remove the legacy printers in subsequent versions.

- kubectl gets columns back from the server, not the client, to allow extensions to work cleanly

- https://github.com/kubernetes/enhancements/issues/515

CLUSTER LIFECYCLE

kubeadm Config file graduation (v1beta2)

- Graduated to Beta

- For the 1.15 cycle, upgrade the config type from v1beta1 to v1beta2.

- The kubeadm config file was originally created as alternative to command line flags for kubeadm init and kubeadm join actions, but over time the number of options supported by the kubeadm config file has grown continuously, while the number of command line flags is intentionally kept under control and limited to the most common and simplest use cases.

- **V1beta2 - Add config options for new and existing kubeadm features**
- Over time kubeadm gains new features which may require the addition of new settings to the config format. One notable such feature, that was introduced after the release of v1beta1 is the Certificates copy for `join --control-plane`

- https://github.com/kubernetes/enhancements/issues/970

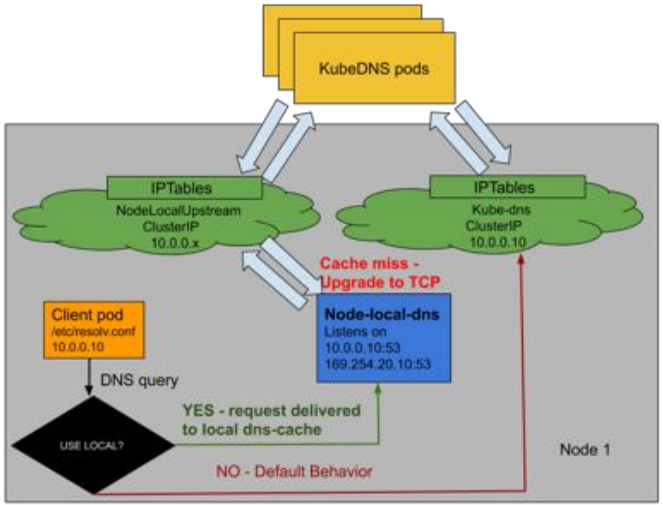
NETWORK

kubeadm Config file graduation (v1beta2)

- Graduated to Beta

- NodeLocal DNSCache is an addon that runs a dnsCache pod as a daemonset to improve clusterDNS performance and reliability.

- Begin implementation of HA. Use an additional listen IP for node-local-dns pod. Extend node-local-dns to listen on the kube-dns service IP as well. Requests to kube-dns service IP will be handled by node-local-dns pod when it is up. If it is unavailable, the requests will go to kube-dns endpoints instead.



- https://github.com/kubernetes/enhancements/issues/1024

Finalizer Protection for Service LoadBalancers

- Net New Alpha

- 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

NODE

CLOUD NATIVE
COMPUTING FOUNDATION

Quotas for Ephemeral Storage

- Net New Alpha

- Use filesystem quotas to monitor local ephemeral storage utilization.

- This enhancement utilizes filesystem project quotas to provide monitoring of resource consumption and *optionally enforcement of limits*. Project quotas offer a kernel-based means of monitoring *and restricting* filesystem consumption that can be applied to one or more directories.

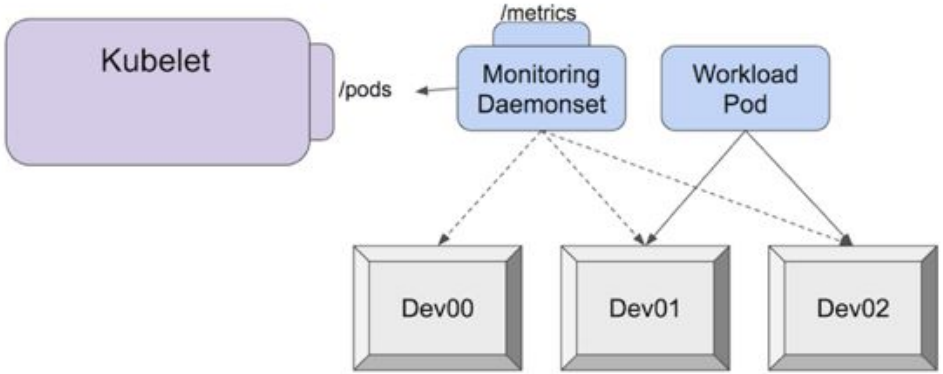
```
apiVersion: v1
kind: Pod
max:
metadata:
  name: "diskhog"
spec:
  containers:
  - name: "perl"
    resources:
      limits:
        ephemeral-storage: "2048Ki"
    image: "perl"
    command:
    - perl
    - -e
    - >
      my $file = "/data/a/a"; open OUT, ">$file" or die "Cannot open $file: $!\n"; unlink "$file" or die
    volumeMounts:
    - name: a
      mountPath: /data/a
  volumes:
  - name: a
    emptyDir: {}
```

- https://github.com/kubernetes/enhancements/issues/1029

Support 3rd party device monitoring plugins

- Graduate to Beta

- Device Monitoring requires external agents to be able to determine the set of devices in-use by containers and attach pod and container metadata for these devicesDynamic Audit Control provide a means of configuring the advanced auditing features post cluster provisioning.



- https://github.com/kubernetes/enhancements/issues/606

PID Limiting

- Graduated to Beta

- Enable isolation of pid resources. A mechanism to enable pod-to-pod PID isolation as well as node-to-pod PID isolation.
- **Pod to Pod Isolation**
  - To enable pid isolation among pods, the SupportPodPidsLimit feature gate is defined.
  - If enabled, the kubelet argument for pod-max-pids will write out the configured pid limit to the pod level cgroup to the value specified on Linux hosts. If -1, the kubelet will default to the node allocatable pid capacity.
- **Node to Pod Isolation**
  - To enable pid isolation from node to pods, the SupportNodePidsLimit feature gate is proposed. If enabled, pid reservations may be supported at the node allocatable and eviction manager subsystem configurations.
  - Node allocatable is a well-established feature concept in the kubelet that allows isolation of user pod resources from host daemons at the kubepods cgroup level that parents all end-user pods.

- https://github.com/kubernetes/enhancements/issues/751

SCALABILITY

PID Limiting

- Net New Alpha

- Add more structure to Event API and change deduplication logic so Events won't overload the cluster.

- This effort has two main goals - reduce performance impact that Events have on the rest of the cluster and add more structure to the Event object which is first and necessary step to make it possible to automate Event analysis

- https://github.com/kubernetes/enhancements/issues/606

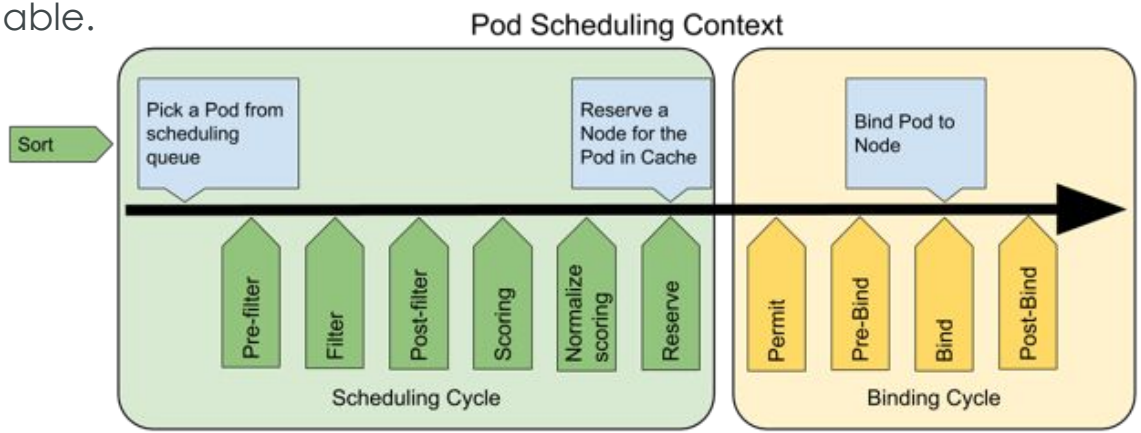
SCHEDULING

CLOUD NATIVE
COMPUTING FOUNDATION

Scheduling Framework

- Net New Alpha

- The scheduling framework is a new set of "plugin" APIs being added to the existing Kubernetes Scheduler. Plugins are compiled into the scheduler, and these APIs allow many scheduling features to be implemented as plugins, while keeping the scheduling "core" simple and maintainable.



- https://github.com/kubernetes/enhancements/issues/624

Add non-preempting option to PriorityClasses

- Graduated to Stable

- This feature adds a new option to PriorityClasses, which can enable or disable pod preemption
- Add a Preempting field to both PodSpec and PriorityClass. Setting the Preempting field in PriorityClass provides a straightforward interface, and allows ResourceQuotas to restrict preemption.

- https://github.com/kubernetes/enhancements/issues/902

STORAGE

CLOUD NATIVE
COMPUTING FOUNDATION

Support for Online Resizing of PVs

- Graduated to Beta

- Enable users to increase size of PersistentVolumes already mounted.
  - Release 1.9 only supported offline file system resizing for PVs in kubelet, as this operation is only executed inside the `MountVolume` operation. If a resizing request was submitted after the volume mounted, it won't be performed.
- Uses:
  - As a user I am running Mysql on a 100GB volume - but I am running out of space, I should be able to increase size of volume mysql is using without losing all my data. (online and with data)
  - As a user I am running an application on glusterfs. I should be able to resize the gluster volume without losing data or mount point. (online and with data and without taking pod offline)
- Implementation:
  - The key point of online file system resizing is how kubelet to discover which PVCs need file system resizing. We achieve this goal by adding a `volumeFSResizingAnnotation` annotation to pod.
- https://github.com/kubernetes/enhancements/issues/531

Provide environment variable expansion in sub path mounts

- Graduated to Beta

- Provide environment variable expansion in sub path mounts
- a way to dynamically generate host paths when mounting volumes. The subPath feature creates directories on demand, but the names assigned to those directories are static.
- Supporting the downward API variables would provide a good way to share storage and avoid collisions. Centralized log storage is one use case.

```
    env:
    - name: NAME
      valueFrom:
        fieldRef:
          fieldPath: metadata.name
    volumeMounts:
    - mountPath: /var/log/mysql
      name: logs
      subPath: $(NAME)
  volumes:
  - name: logs
    hostPath:
      path: /mnt/log-repo
```

```
/mnt/log-repo/mysql-1174854418-p8t02
/mnt/log-repo/mysql-1174854418-plsf8
/mnt/log-repo/mysql-3521978760-ns129
/mnt/log-repo/mysql-3521978760-s1m6x
/mnt/log-repo/mysql-3957873462-k4gg7
/mnt/log-repo/mysql-3957873462-rqsr1
```

- https://github.com/kubernetes/enhancements/issues/559

In-tree storage plugin to CSI Driver Migration

- More Alpha Work

- API changes to support migration of inline in-tree volumes to CSI #77703
- Handle CSI volume resize migration. #77994
- Translate StorageClass object instead of parameters. Add GCE PD Storage class translation logic. #77837

- https://github.com/kubernetes/enhancements/issues/625

ExecutionHook

- Net New Alpha

- Provide ExecutionHook API design to trigger hook commands in the containers for different use cases, e.g., volume snapshot and application snapshot.
- Introduce an API (ExecutionHook) for dynamically executing user's commands in a pod/container or a group of pods/containers and a controller (ExecutionHookController) to manage the hook lifecycle. ExecutionHook provides a general mechanism for users to trigger hook commands in their containers for their different use cases

- https://github.com/kubernetes/enhancements/issues/962

What's coming next?

- Already 4 weeks into 1.16
  - Enhancements freeze is July 30th
- Targeted GA is September 16th

Questions?

CLOUD NATIVE
COMPUTING FOUNDATION

Thank You

CLOUD NATIVE
COMPUTING FOUNDATION