How Kubernetes Initializers Work

Di Xu
Kubernetes Member
@dixudx

Contents

- Admission Controllers
- Use Cases
- What Are Initializers
- How Initializers Work
- Developing Your Own Initializer

Admission Controllers

Comm only Used

AlwaysPullImages, DefaultStorageClass, DefaultTolerationSeconds, LimitRanger, NamespaceLifecycle, NamespaceAutoProvision, NamespaceExists, NodeRestriction, PersistentVolumeLabel, PodPreset, PodSecurityPolicy, Priority, ResourceQuota, ServiceAccount, SecurityContextDeny

New Added in v1.9

ExtendedResourceToleration

Admission Controllers: Disadvantages

- Must compiled together with Kubernetes binaries;
- Hard to customized by maintaining a fork;
- Not dynamic:
  - Only configurable with *--admission-control* flag;
  - Re-deploy (may have no authority);

Use Cases

Dynamic Admission Control

- Adding new nodes to the cluster, but don't want them get scheduled by default. (Easier than adding taints and tolerations, pod affinity/anti-affinity)
  - Want to reserve resources for dev/test for a team, for a single service;
  - Testing GPU nodes;
  - Temporary/Unstable nodes
    - Node flapping between Ready/NotReady with PLEG issues (#45419);
    - Node maintainence;
- Time-based Resources Allocation (Day and Night, Holidays, etc);
- Enforcing all container images to come from a particular registry, and prevent other images;
- TPR (Third Party Resources);
- ……

Initializers And Web Hooks

- Initializers
- Webhooks
  - HTTP callbacks to receive admission requests;
  - Not allowed to mutate the admission request in any way;
  - Must support TLS;
  - Better performance than Initializers;

What Are Initializers

- Similar to admission controller plug-ins, but different;
- A list of pending pre-initialization tasks (uninitialized), stored in every object's *METADATA* (e.g., "AddMyCorporatePolicySidecar");
- *A customized controller;*

Initializers: Status Quo

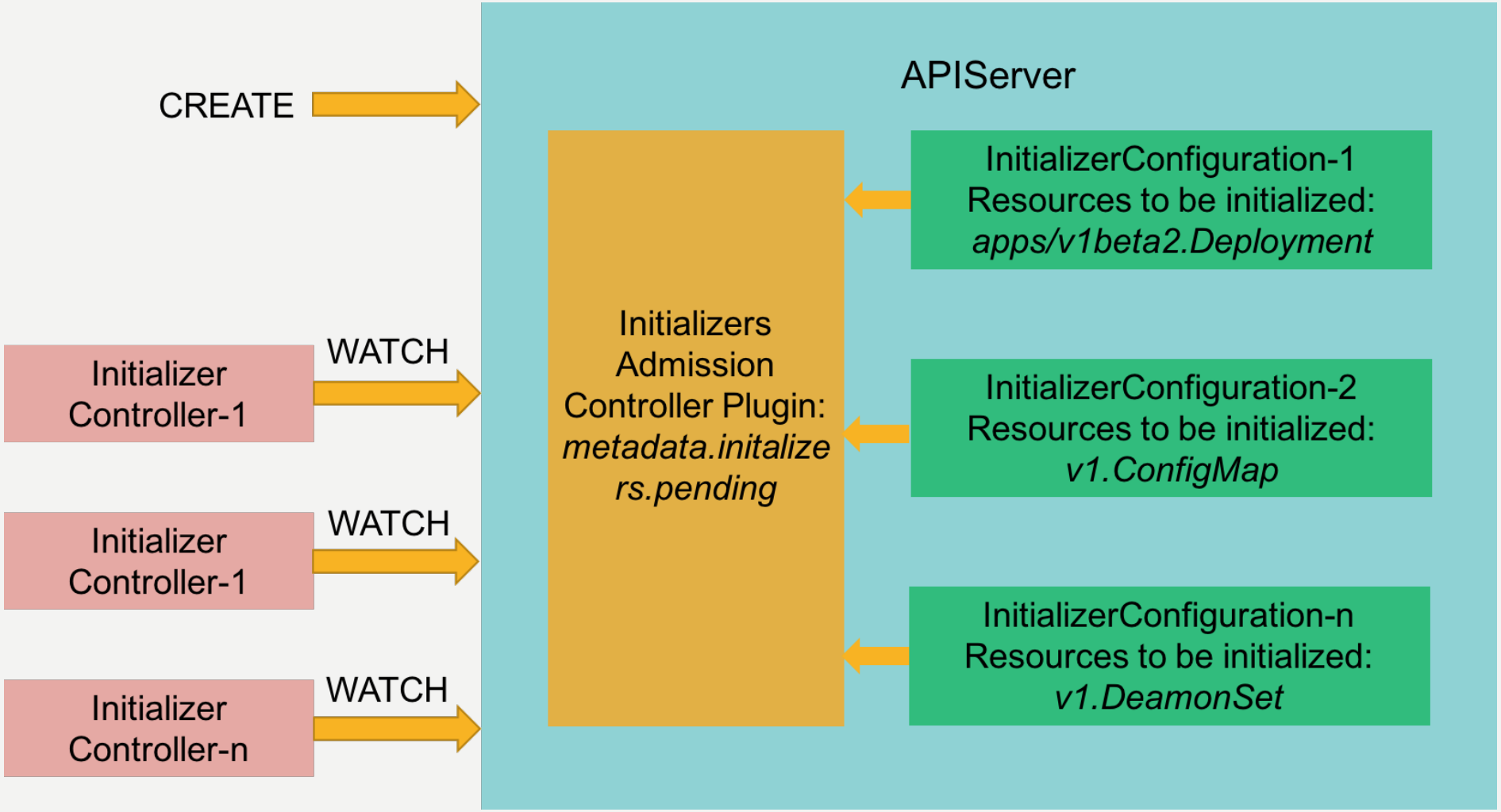
- First introduced in v1.7 (alpha). Only enabled in API.
  - requested by using the query parameter, "*?includeUninitialized=true*"
- Enabled in CLI in v1.8 (PR #50497).
  - New global flag *"--include-uninitialized*";
  - Add flag to kubectl annotate, apply, edit-last-applied, delete, describe, edit, get, label, set;
  - Ignored by default only if the names of the objects are provided.

CLI Usage

	annotate/edit-last-applied/delete/edit/label/get/set xyz	apply	describe
Default Value	false	--prune	true
--all=true	true	true	true
--selector=xyz	false	false	false
Explicit set flag			

- *--all*
  - include the uninitialized objects by default unless explicitly set *--include-uninitialized=false*
- *--selector*
  - does not include the uninitialized objects by default unless explicitly set *--include-uninitialized=true*

How Initializers Work



Enable Initializers

- Alpha feature. disabled by default.
- *--admission-control=Initializers* flag when starting kube-apiserver;
- *--runtime-config=admissionregistration.k8s.io/v1alpha1* when starting kube-apiserver;
- *--feature-gate=Initializers=true* since v1.8;

Initializers: Under The Covers

- Not scheduled directly?
- Saved on APIServer? *?includeUninitialized=true*
- Invisible for schedulers and controller managers?
- Initializers block Create requests
- Parallel? Too slow? (API support for partial orders among initializers #50714)

Initializers: Under The Covers (Cont.)

- Allow administrators to bypass initialization by setting an empty initializers list;
- *sub-resource action should be denied until the object is initialized?*
- Mirror pods are exempt from initialization;
- Pending initializers are *SORTED*!!!

PodPreset VS Initializers

- PodPreset:
  - injects certain information into pods (use label selectors) at creation time.
  - The information can include secrets, volumes, volume mounts, and environment variables;
- Intializers:
  - Dynamic and pluggable;
  - More customized; For all kinds of objects;
  - Easy to use *client-go* to handle common actions;

Developing Your Own Initializer

- List/Watch functions specifying IncludeUninitialized=true and targeting all namespaces
- Use a PATCH to perform the update;
- Set empty pending initializers to skip blocking initializer controller;
- Make sure your initializer doesn't go down and handles the objects quickly;
- Sample Gist

Some Warnings

- Never forget about the ORDER!!!
  - matches *metadata.initializers.pending[0]*
  - Ensure PVL controller is next pending initializer before labeling the PV (#56831)
- Cautious with the powerful initializers.
- Not everyone should be required to write an initializer.
- Uptime of Initializers: resources will get stuck in “uninitialized” state indefinitely

DaoCloud

IT大咖说 知识共享平台

Q&A