whois

Mateusz Szostok

Senior Software Engineer
at SAP Labs Poland

Kyma Maintainer

Service Catalog SIG Chair

SIG-Service Catalog Deep-Dive

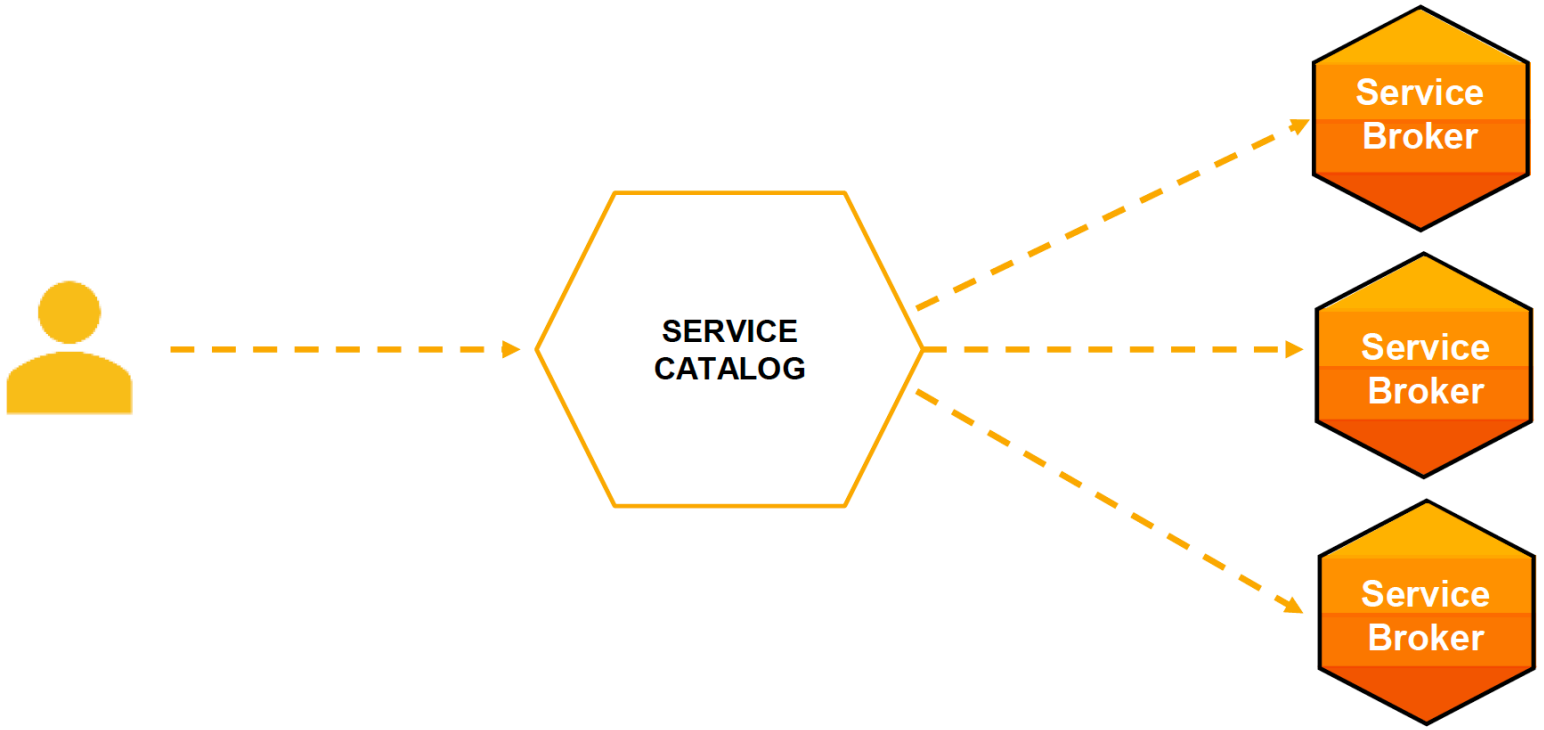
Mateusz Szostok, SAP

You will learn

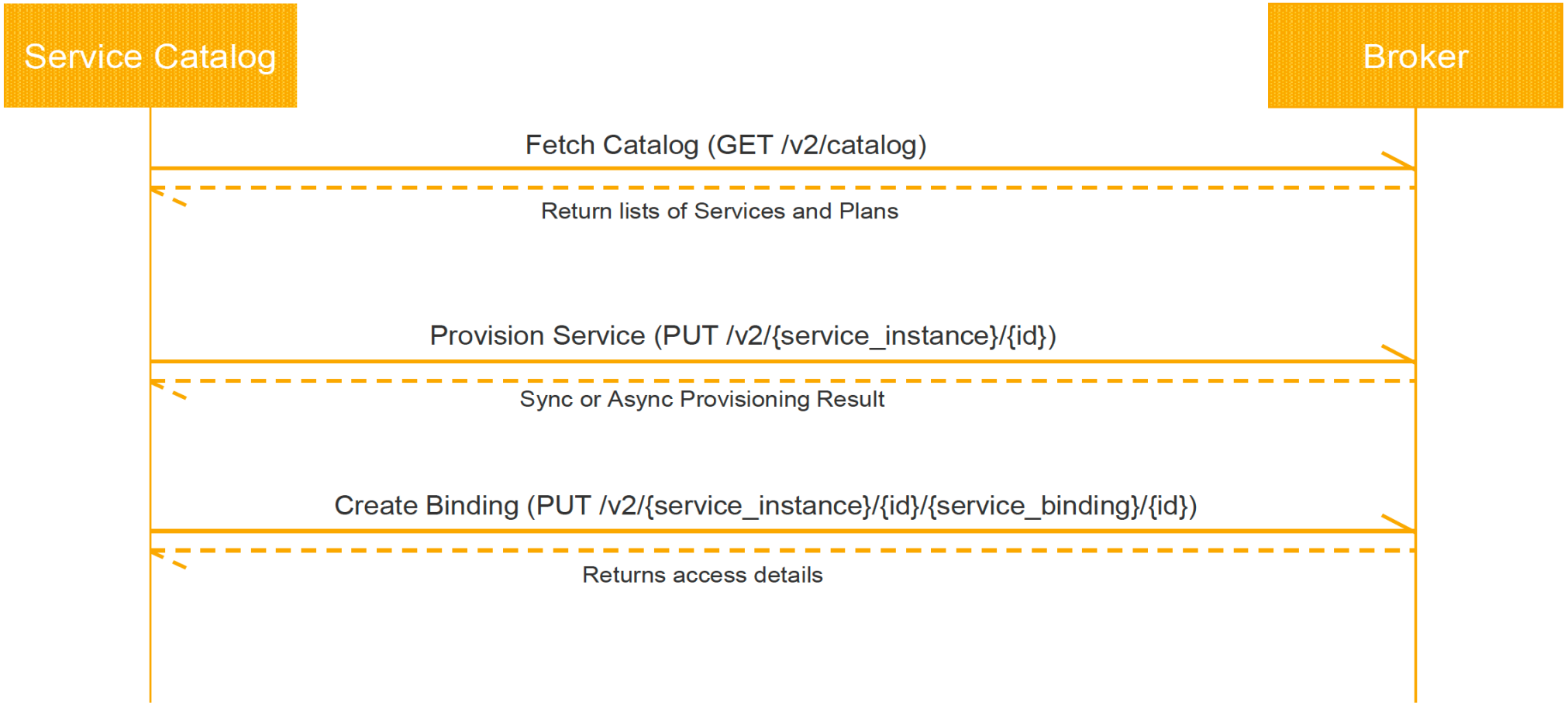
- Open Service Broker API Specification
- Status of CRD implementation
- New release process
- New subprojects that we own
- Cascade deletion proposal
- Future plans

Open Service Broker API

The Open Service Broker API defines the interaction between the platform and a broker



Open Service Broker API



…Unbind, Deprovision Service

Open Service Broker API

Service Catalog

- **ClusterServiceBroker** (and namespaced **ServiceBroker**)

  A server running somewhere that offers various services, e.g. MySQL Broker

- **ClusterServiceClass** (and namespaced **ServiceClass**)

  A category of services offered by a Broker, e.g. MySQL Databases

- **ClusterServicePlan** (and namespaced **ServicePlan**)

  A specific type of a Service that a Broker offers, e.g. 100 MB MySQL Databases

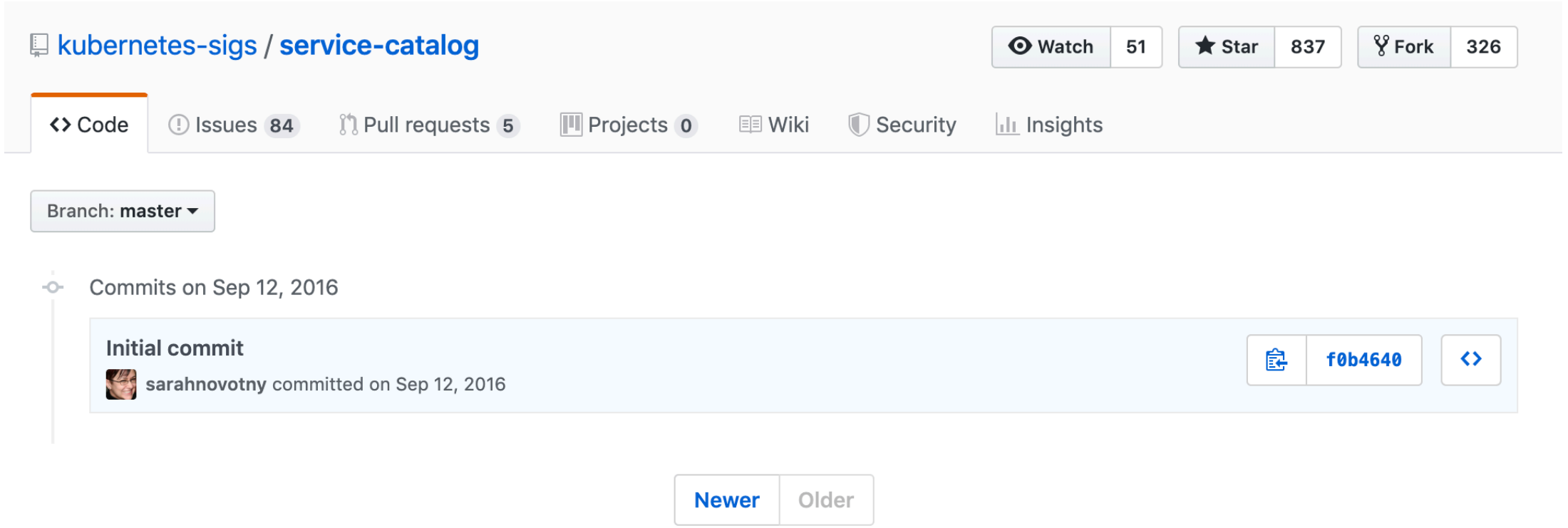
- **ServiceInstance**

  A single instantiation of a Service/Plan, e.g. Matt's 100 MB MySQL Database

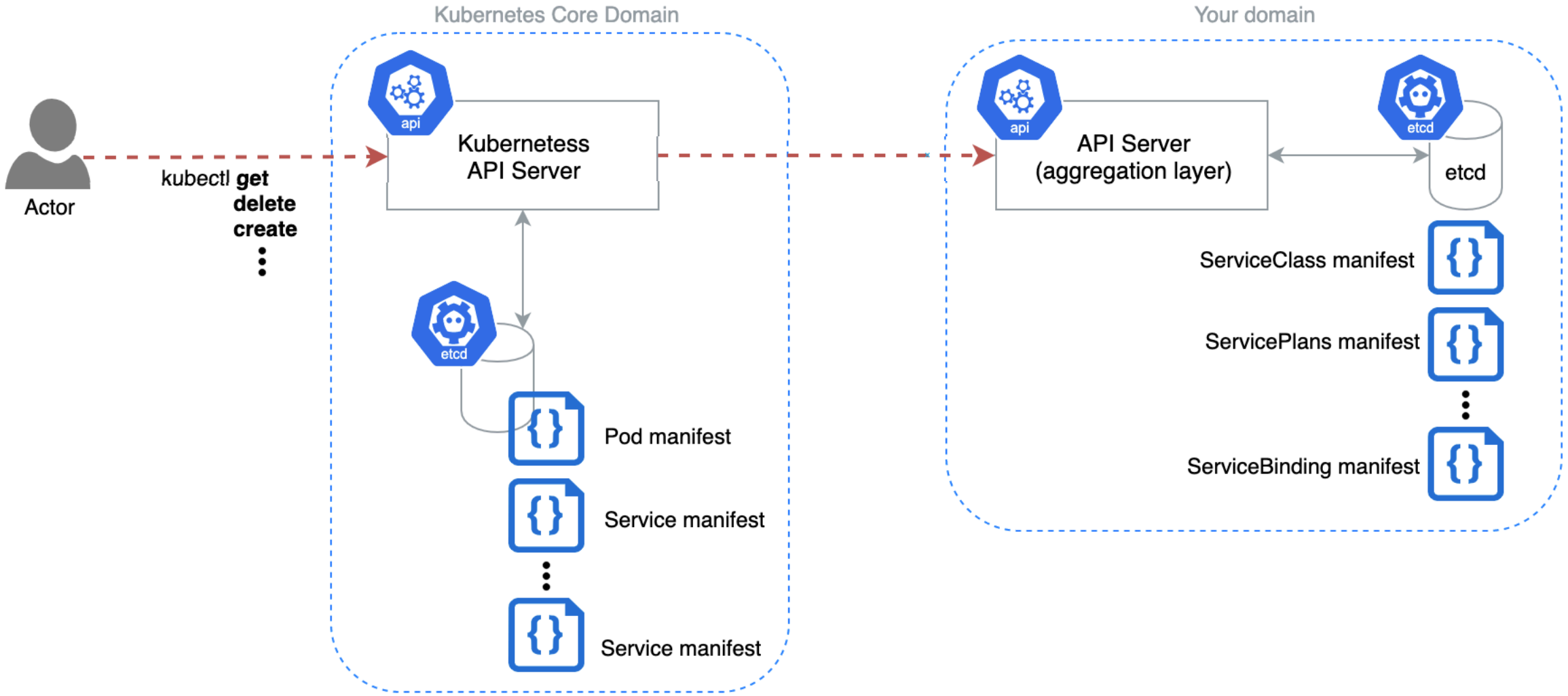
- **ServiceBinding**

  A unique set of creds to access a specific Instance, e.g. username/password for Matt's 100 MB MySQL Database

Long long time ago



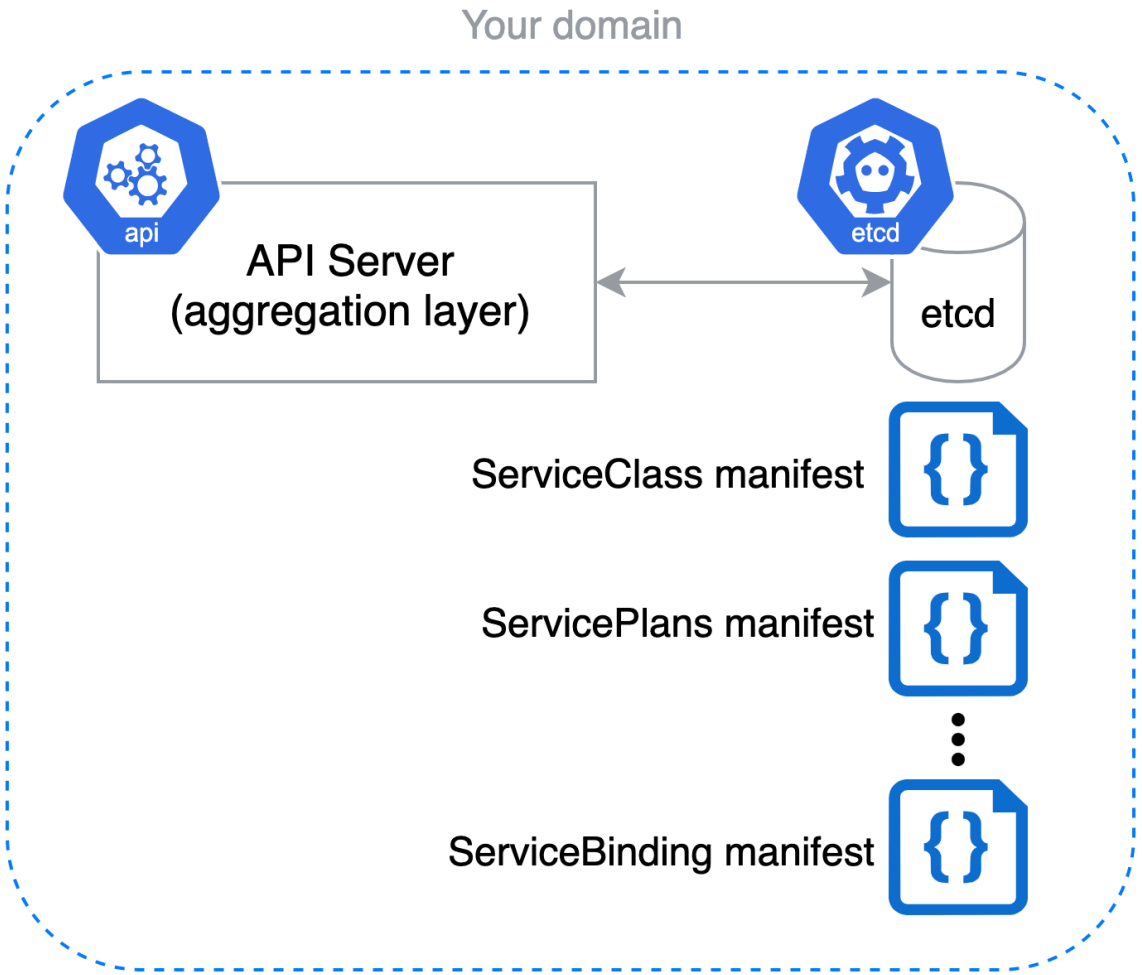
API Server Architecture



How good architecture goes bad

a.k.a why it's worth switching to CRDs in your project

reason 0


Your domain
api
API Server
(aggregation layer)
etcd
etcd
ServiceClass manifest
ServicePlans manifest
ServiceBinding manifest

reason 0

- registry
  - servicecatalog
    - binding
      - storage.go
      - storage_test.go
      - strategy.go
      - strategy_test.go
    - clusterservicebroker
      - storage.go
      - storage_test.go
      - strategy.go
      - strategy_test.go
    - clusterserviceclass
      - storage.go
      - storage_test.go
      - strategy.go
    - clusterserviceplan
      - storage.go
      - strategy.go
    - instance
      - storage.go
      - storage_test.go
      - strategy.go
      - strategy_test.go
    - rest
    - server
    - servicebroker
      - storage.go
      - storage_test.go
      - strategy.go
      - strategy_test.go
    - serviceclass
      - storage.go
      - storage_test.go
      - strategy.go
    - serviceplan
      - storage.go
      - strategy.go

236 lines (202 sloc) | 8.24 KB — Raw | Blame | History

```
/*
```

236 lines (202 sloc) | 8.24 KB — Raw | Blame | History

```
/*
Copyright 2017 The Kubernetes Authors.
```

93 lines (73 sloc) | 3.3 KB — Raw | Blame | History

```
/*
Copyright 2017 The Kubernetes Authors.
```

135 lines (118 sloc) | 4.52 KB — Raw | Blame | History

```
/*
Copyright 2017 The Kubernetes Authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
*/

package apiserver

import (
        "k8s.io/apiserver/pkg/registry/generic"
        "k8s.io/apiserver/pkg/registry/generic/registry"
        "k8s.io/apiserver/pkg/registry/rest"
        genericapiserver "k8s.io/apiserver/pkg/server"
        "k8s.io/apiserver/pkg/server/storage"
        "k8s.io/klog"
)

// EtcdConfig contains a generic API server Config along with config specific to
// the service catalog API server.
type etcdConfig struct {
        genericConfig *genericapiserver.RecommendedConfig
        extraConfig   *extraConfig
}

// extraConfig contains all additional configuration parameters for etcdConfig
type extraConfig struct {
        // BABYNETES: cargo culted from master.go
        deleteCollectionWorkers int
        storageFactory          storage.StorageFactory
}

// NewEtcdConfig returns a new server config to describe an etcd-backed API server
func NewEtcdConfig(
        genCfg *genericapiserver.RecommendedConfig,
        deleteCollWorkers int,
        factory storage.StorageFactory,
) Config {
        return &etcdConfig{
                genericConfig: genCfg,
                extraConfig: &extraConfig{
                        deleteCollectionWorkers: deleteCollWorkers,
                        storageFactory:          factory,
                },
        }
}

// Complete fills in any fields not set that are required to have valid data
// and can be derived from other fields.
func (c *etcdConfig) Complete() CompletedConfig {
```

reason 1

A lot of code

Problem: Owning a lot of code which is not directly connected with your business logic.

reason 1

```
118  func (clusterServiceBrokerRESTStrategy) PrepareForUpdate(ctx context.Context, new, old runtime.Object) {
119          newClusterServiceBroker, ok := new.(*sc.ClusterServiceBroker)
120          if !ok {
121                  klog.Fatal("received a non-clusterservicebroker object to update to")
122          }
123          oldClusterServiceBroker, ok := old.(*sc.ClusterServiceBroker)
124          if !ok {
125                  klog.Fatal("received a non-clusterservicebroker object to update from")
126          }
127
128          newClusterServiceBroker.Status = oldClusterServiceBroker.Status
129
130          // Ignore the RelistRequests field when it is the default value
131          if newClusterServiceBroker.Spec.RelistRequests == 0 {
132                  newClusterServiceBroker.Spec.RelistRequests = oldClusterServiceBroker.Spec.RelistRequests
133          }
134
135          // Spec updates bump the generation so that we can distinguish between
136          // spec changes and other changes to the object.
137          if !apiequality.Semantic.DeepEqual(oldClusterServiceBroker.Spec, newClusterServiceBroker.Spec) {
138                  newClusterServiceBroker.Generation = oldClusterServiceBroker.Generation + 1
139          }
140  }
```

reason 1

Duplicating core logic

Problem: Easy to introduce bugs
Problem: Hard to follow best practices

reason 2

Bump kube 1.15 #2683

Edit

Merged **k8s-ci-robot** merged 1 commit into `kubernetes-sigs:master` from `jberkhahn:bump_kube_1.15` 23 days ago

Conversation 18 | Commits 1 | Checks 0 | Files changed 362 | +17,420 −5,871

jberkhahn commented on Aug 3 — Member

No description provided.

Bump kube 1.15 is **closed** — + add dependency

k8s-ci-robot added the `cncf-cla: yes` label on Aug 3

Pipelines

S service-catalog
Closed

Reviewers

MHBauer

mszostok

reason 2

4 pkg/registry/servicecatalog/server/options.go

Viewed

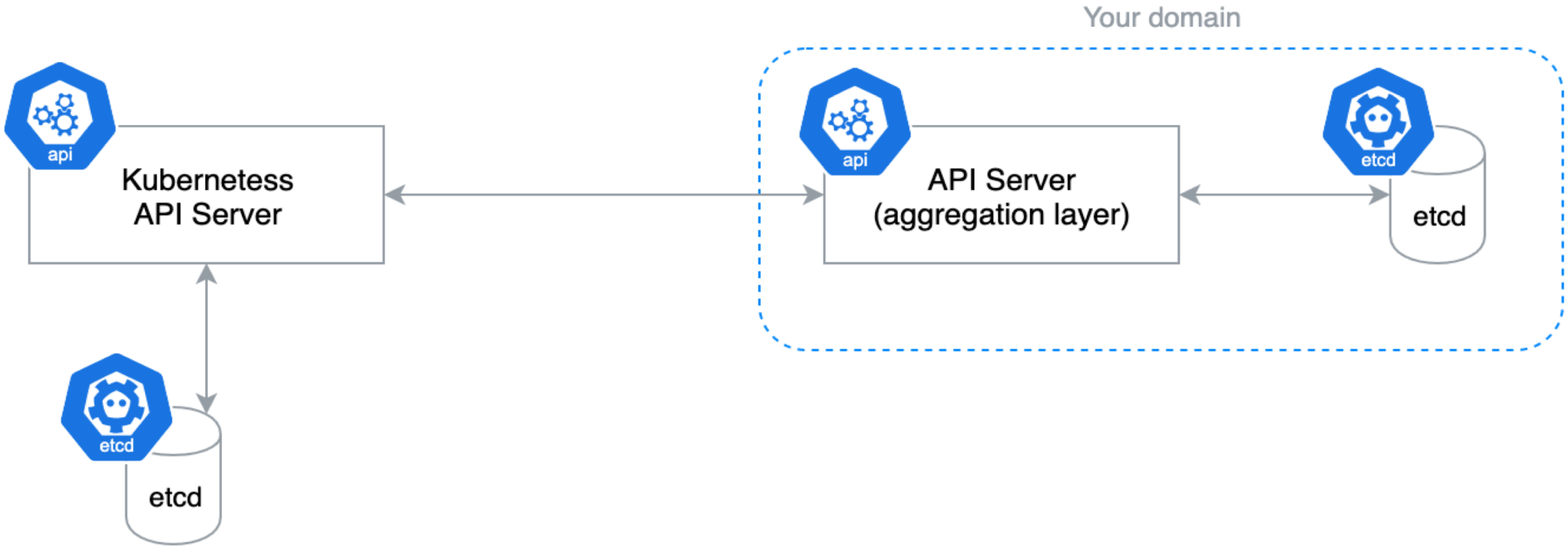
```
@@ -72,19 +72,19 @@ func (o Options) KeyFunc(namespaced bool) func(context.Context, string) (string,
72 72
73 73    // GetStorage returns the storage from the given parameters
74 74    func (o Options) GetStorage(
75    -           objectType runtime.Object,
76 75              resourcePrefix string,
77 76              scopeStrategy rest.NamespaceScopedStrategy,
   77 +           newFunc func() runtime.Object,
78 78              newListFunc func() runtime.Object,
79 79              getAttrsFunc storage.AttrFunc,
80 80              trigger storage.TriggerPublisherFunc,
81 81    ) (registry.DryRunnableStorage, factory.DestroyFunc) {
82 82              etcdRESTOpts := o.EtcdOptions.RESTOptions
83 83              storageInterface, dFunc := etcdRESTOpts.Decorator(
84 84                      etcdRESTOpts.StorageConfig,
85    -                     objectType,
86 85                      resourcePrefix,
87 86                      nil, /* keyFunc for decorator -- looks to be unused everywhere */
   87 +                   newFunc,
88 88                      newListFunc,
89 89                      getAttrsFunc,
90 90                      trigger,
```

reason 2

Be in sync with Kubernetes version

Problem: a lot of manual changes

reason 3


Your domain
api
Kubernetess
API Server
etcd
etcd
api
API Server
(aggregation layer)
etcd
etcd

reason 3

Requires own etcd storage

Problem: Our clients need to maintain an additional database for managed Kubernetes installation.

New kid on the block

New kid on the block

Stefan Schimanski
@the_sttts

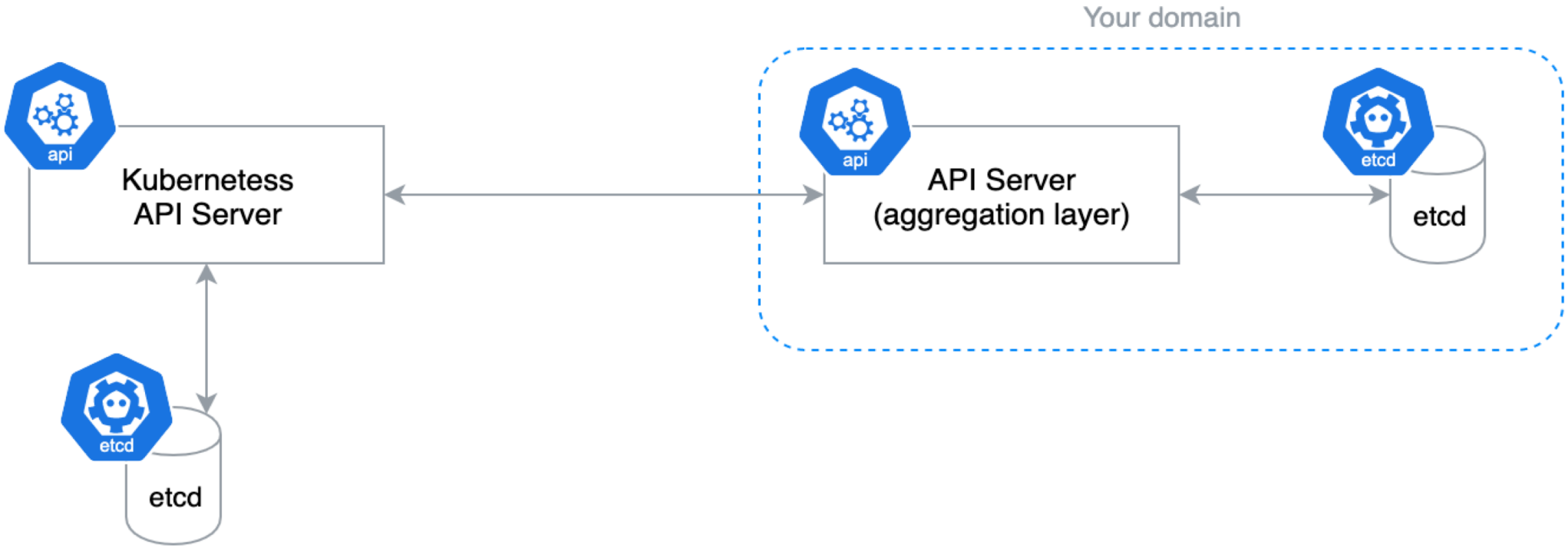
github.com/kubernetes/kub... merged 😍 CRDs are officially GA now with 1.16. Awesome work from everybody involved!

Bump the CRD feature gates to GA by roycaihw · Pull Request #81965 ...
What type of PR is this? /kind cleanup What this PR does / why we need it: Promote the following CRD feature gates to GA and lock the default ...
github.com

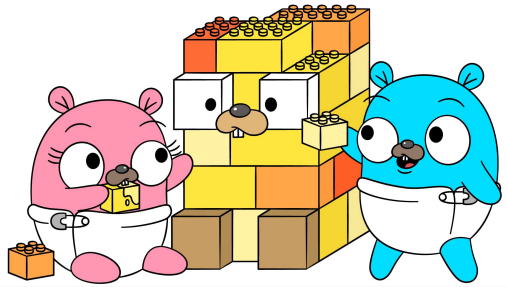
11:15 AM · Aug 29, 2019 · Tweetbot for Mac

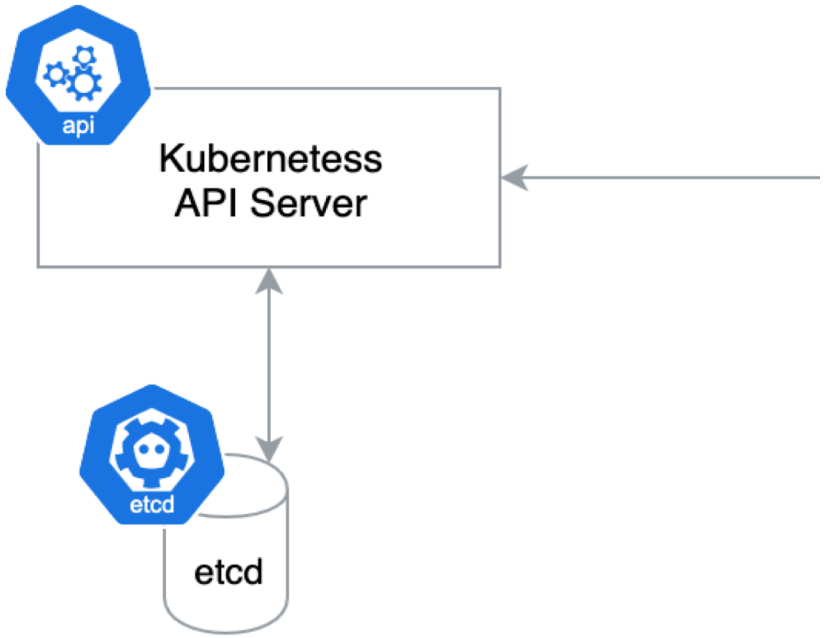
92 Retweets **259** Likes

How hard could CRUD be?



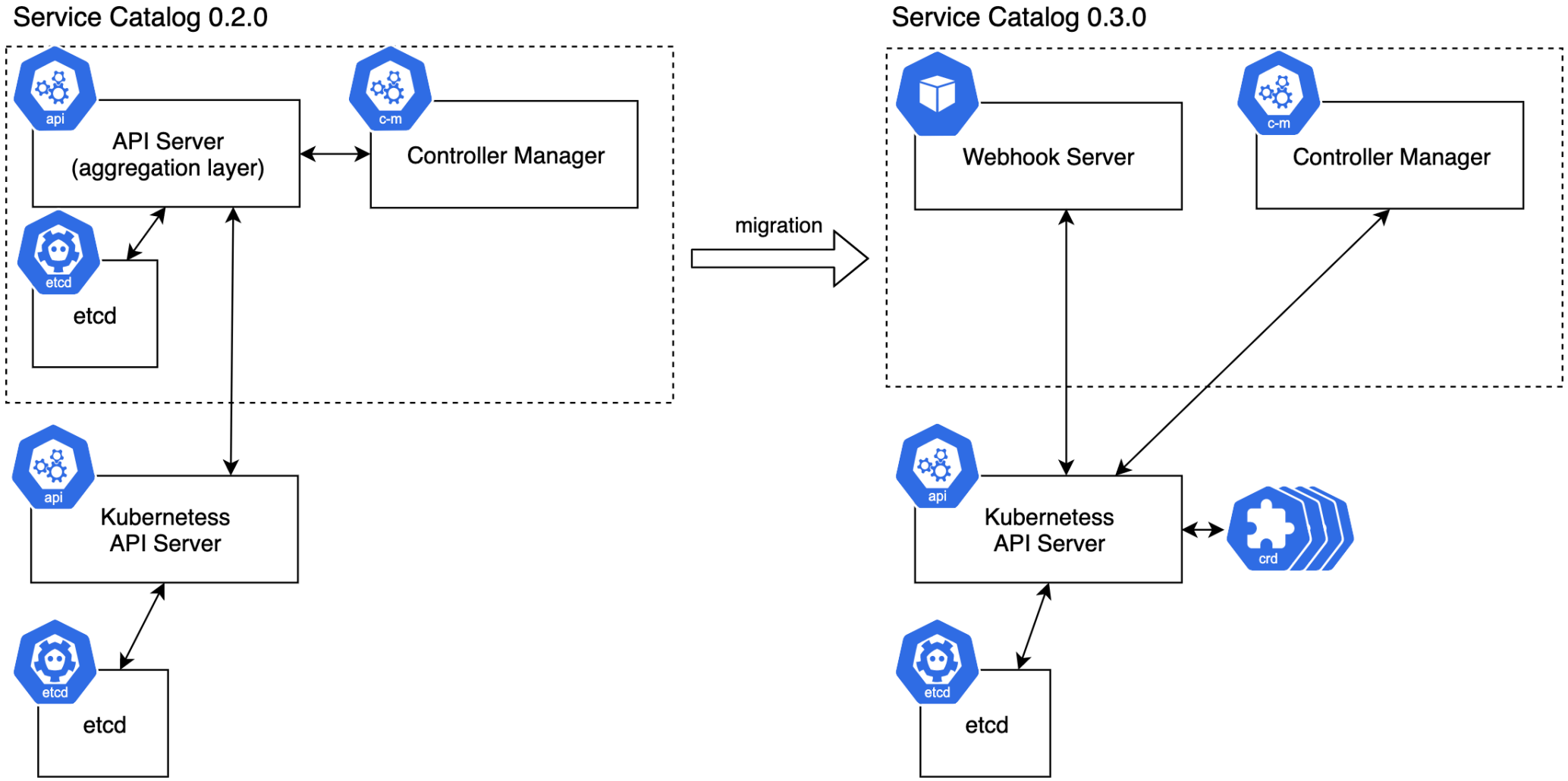
How hard could CRUD be?






```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  # name must match the spec fields below, and be in the form: <plural>.<group>
  name: crontabs.stable.example.com
spec:
  # group name to use for REST API: /apis/<group>/<version>
  group: stable.example.com
  # list of versions supported by this CustomResourceDefinition
  versions:
    - name: v1
      # Each version can be enabled/disabled by Served flag.
      served: true
      # One and only one version must be marked as the storage version.
      storage: true
  # either Namespaced or Cluster
  scope: Namespaced
  names:
    # plural name to be used in the URL: /apis/<group>/<version>/<plural>
    plural: crontabs
    # singular name to be used as an alias on the CLI and for display
    singular: crontab
    # kind is normally the CamelCased singular type. Your resource manifests use this.
    kind: CronTab
    # shortNames allow shorter string to match your resource on the CLI
    shortNames:
    - ct
```

New architecture



How did we move our features

Migrate features

Subresource
A custom sub-resource allows you to define fine-grained actions on its Kind. For example, user writes **spec** section, controller writes **status** section.

TableConvertor
API Server uses TableConvertor for printing custom columns on `kubectl get` command.

PrepareForCreate/Update func
Functions to adjust newly created resources or updates existing ones.

Validation
API Server uses plugins which are gathered and registered as validators.

FieldSelector
Restricts the list of returned objects by their fields.

FieldSelector

```
89  // RetrieveInstancesByPlan retrieves all instances of a plan.
90  func (sdk *SDK) RetrieveInstancesByPlan(plan Plan) ([]v1beta1.ServiceInstance, error) {
91          planOpts := v1.ListOptions{
92                  FieldSelector: fields.OneTermEqualSelector(FieldServicePlanRef, plan.GetName()).String(),
93          }
94          instances, err := sdk.ServiceCatalog().ServiceInstances("").List(planOpts)
95          if err != nil {
96                  return nil, fmt.Errorf("unable to list instances (%s)", err)
97          }
98
99          return instances.Items, nil
100 }
```

CRDs - Field Selector

Open **Generic field selectors** #1362

lavalamp opened this issue on Sep 18, 2014 · 26 comments

liggitt commented on Jan 23, 2018 — Member

> I found that it didn't support pass metadata.uid parameter to get the pod in the api. Is this design unreasonable?

Using a field selector to look up a single object in a list is *extremely* inefficient today. We'd likely need a way to build indexes before supporting that query pattern across all objects (especially across namespaces)

Label me!

```
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ClusterServiceClass
metadata:
  name: sample-cluster-class
  labels:
    servicecatalog.k8s.io/spec.clusterServiceBrokerName: 6ca41896e...// <- sha224
    servicecatalog.k8s.io/spec.externalID: f4947a36695f7a77...       // <- sha224
    servicecatalog.k8s.io/spec.externalName: 730d36fa34c1b3308...    // <- sha224
spec:
  bindable: true
  bindingRetrievable: false
  clusterServiceBrokerName: ups-broker
  description: A user provided service
  externalID: 4f6e6cf6-ffdd
  externalName: user-provided-service
  planUpdatable: true
```

Label me!

„(..) Valid label values must be **63** characters or less and must be empty or **begin and end with an alphanumeric** character **([a-z0-9A-Z]) with dashes (-), underscores (_), dots (.),** and alphanumerics between.”

Label me!

```
func (h *CreateUpdateHandler) syncLabels(obj *sc.ClusterServicePlan) {
        if obj.Labels == nil {
                obj.Labels = make(map[string]string)
        }

        obj.Labels[sc.GroupName+"/"+sc.FilterSpecExternalID] = util.GenerateSHA(obj.Spec.ExternalID)
        obj.Labels[sc.GroupName+"/"+sc.FilterSpecExternalName] = util.GenerateSHA(obj.Spec.ExternalName)
        obj.Labels[sc.GroupName+"/"+sc.FilterSpecClusterServiceClassRefName] = util.GenerateSHA(obj.Spec.ClusterServiceClassRef.Name)
        obj.Labels[sc.GroupName+"/"+sc.FilterSpecClusterServiceBrokerName] = util.GenerateSHA(obj.Spec.ClusterServiceBrokerName)
}
```

Label me!

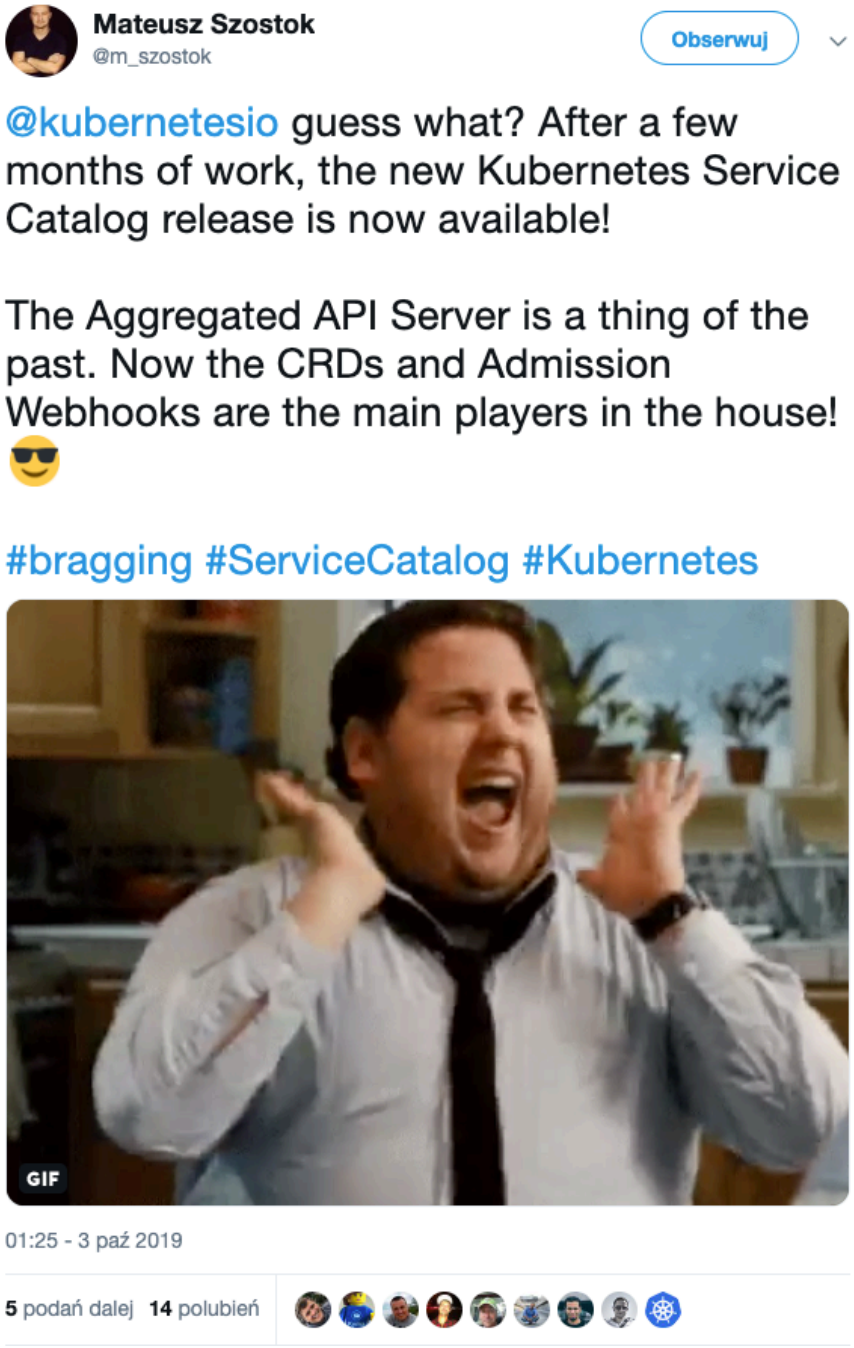
```
89  // RetrieveInstancesByPlan retrieves all instances of a plan.
90  func (sdk *SDK) RetrieveInstancesByPlan(plan Plan) ([]v1beta1.ServiceInstance, error) {
91          planOpts := v1.ListOptions{
92                  FieldSelector: fields.OneTermEqualSelector(FieldServicePlanRef, plan.GetName()).String(),
93          }
94          instances, err := sdk.ServiceCatalog().ServiceInstances("").List(planOpts)
95          if err != nil {
96                  return nil, fmt.Errorf("unable to list instances (%s)", err)
97          }
98
99          return instances.Items, nil
100 }
```

Label me!

```
83  // RetrieveInstancesByPlan retrieves all instances of a plan.
84  func (sdk *SDK) RetrieveInstancesByPlan(plan Plan) ([]v1beta1.ServiceInstance, error) {
85          planOpts := v1.ListOptions{
86                  LabelSelector: labels.SelectorFromSet(labels.Set{
87                          v1beta1.GroupName + "/" + v1beta1.FilterSpecClusterServicePlanRefName: plan.GetName(),
88                  }).String(),
89          }
90          instances, err := sdk.ServiceCatalog().ServiceInstances("").List(planOpts)
91          if err != nil {
92                  return nil, fmt.Errorf("unable to list instances (%s)", err)
93          }
94
95          return instances.Items, nil
96  }
```

Video about migrating other features

New architecture in place!



New architecture in place!

Pre-release

v0.3.0-beta.2

f950e68

Verified

v0.3.0-beta.2

mszostok released this 9 days ago · 2 commits to master since this release

Release notes:

Service Catalog v0.3.0-beta.2 is now available!

It is to help gather feedback from the community as well as give users a chance to test Service Catalog in staging environments before v0.3.0 is officially released.

To install the Service Catalog from this release, run:

```
helm install svc-cat/catalog \
    --name catalog --namespace catalog --version 0.3.0-beta.2
```

In this third beta release for the Service Catalog 0.3.0 milestone, we have worked on the following:

- Fix the problem when the (Cluster)ServiceBroker name longer than 63 characters was not accepted.
- Fix the problem when the (Cluster)ServicePlan name with "_" at the beginning was not accepted.
- Update Deployments in Helm chart to support Kubernetes 1.16.x. More info here.
- Add retries to migration process in the case of restoring Service Catalog objects.

Check beta.0 and beta.1 for more release notes.

We'd really appreciate any feedback on the upgrade procedure and any issues or tips you may run into.

Changes since beta.1

Node selector is a dict (#2733)
Changing Class and Plans labels to sha224 (#2741)
fix golint errors (#2737)
Fix ServiceInstance restore process in migration job (#2735)
cleanup codebase and run gofmt (#2729)
Improve data migration process (restore crds) (#2730)
Improve travis build - skip redundant jobs (#2731)
Change Deployment api to apps/v1 (#2728)
Dump cluster info in case of failed migration and e2e tests (#2723)

SVCat Binaries

macOS: https://download.svcat.sh/cli/v0.3.0-beta.2/darwin/amd64/svcat
Windows: https://download.svcat.sh/cli/v0.3.0-beta.2/windows/amd64/svcat.exe
Linux: https://download.svcat.sh/cli/v0.3.0-beta.2/linux/amd64/svcat

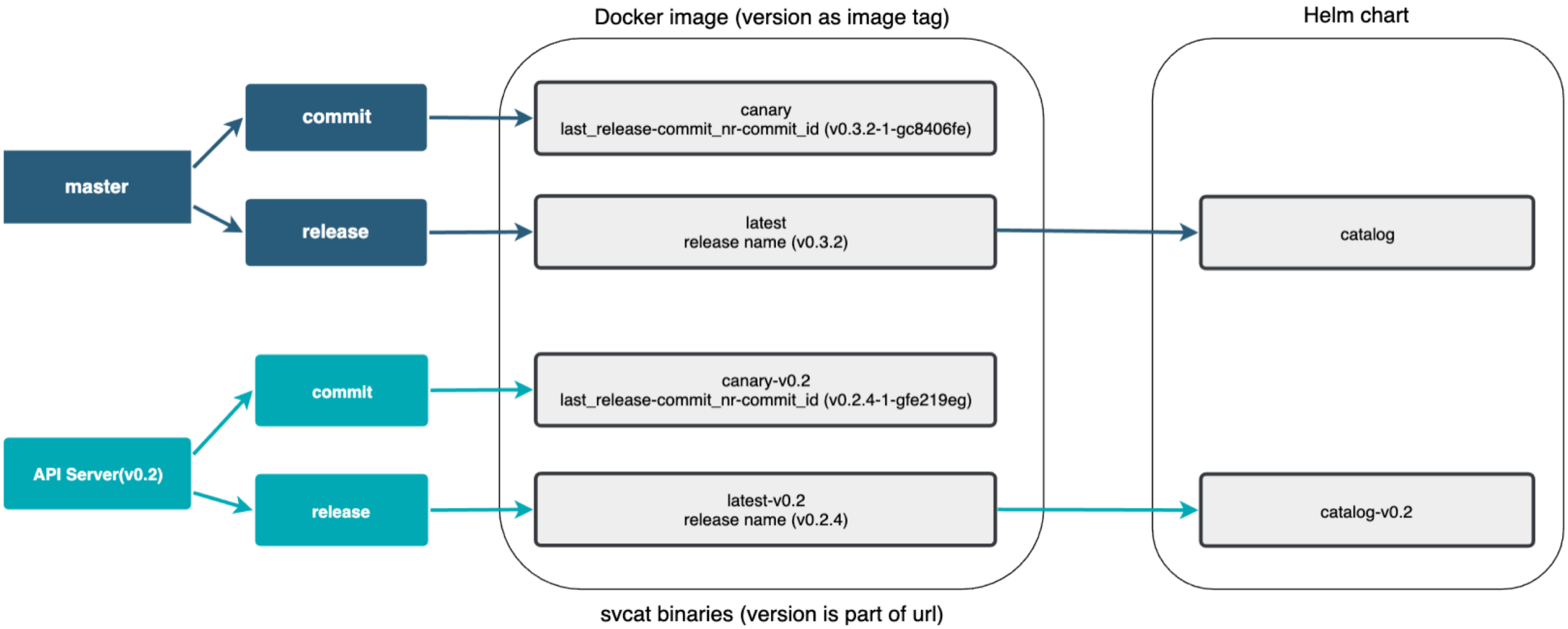
How do these plans affect you

How these plans affect you

The old API Server-based implementation is available on the v0.2 branch.

We support this implementation by providing bug fixes until July 2020.

How these plans affect you



How these plans affect you

master (v0.3 - CRDs)

```
helm install svc-cat/catalog \
   --name catalog --namespace catalog
```

```
helm upgrade catalog svc-cat/catalog
```

v0.2 (api-server)

```
helm install svc-cat/catalog-v0.2 \
   --name catalog --namespace catalog
```

```
helm upgrade catalog svc-cat/catalog-v0.2
```

How these plans affect you

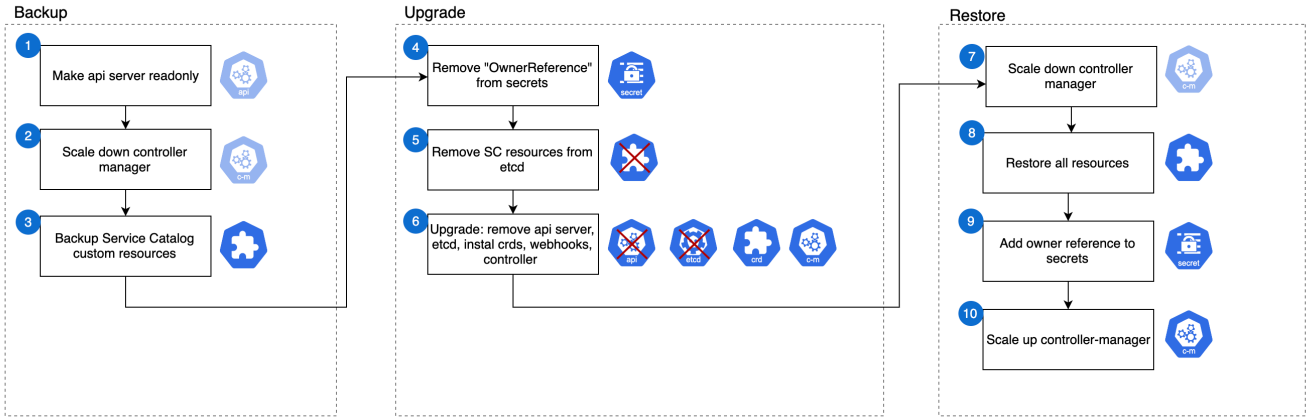
Service Catalog

Documentation GitHub Community

Documentation

Upgrade Service Catalog as a Helm release

The Service Catalog Helm release can be upgraded using the `helm upgrade` command, which runs all necessary actions.



The upgrade to CRDs consists of the following steps:

1. Make API Server read-only. Before any backup, we should block any resource changes to be sure the backup makes a snapshot. We need to avoid any changes when the migration tool is backing up resources.
2. Check if Apiserver deployment with a given name exist. **If deployment was not found we skip the migration**.
3. Scale down the Controller Manager to avoid resources processing, such as Secret deletion.
4. Backup Service Catalog custom resources to files in a Persistent Volume.
5. Remove `OwnerReference` fields in all Secrets pointed by any ServiceBinding. This is needed to avoid Secret deletion.
6. Remove all Service Catalog resources. This must be done if the Service Catalog uses the main Kubernetes etcd instance.
7. Upgrade the Service Catalog: remove API Server, install CRDs, Webhook Server and roll up the Controller Manager.
8. Scale down the Controller Manager to avoid any resource processing while applying resources.
9. Restore all resources. The migration tool sets all necessary fields added in the Service Catalog 0.3.0. Creating resources triggers all logic implemented in webhooks so we can be sure all

How these plans affect you

⚠️ Run the sanity check script before the migration

New subprojects in SIG-Service Catalog

Subproject Minibroker

Minibroker is an implementation of the Open Service Broker API suited for local development and testing.

We use it in the official Service Catalog walkthrough documentation.

Repository: https://github.com/kubernetes-sigs/minibroker

Previously it was a personal project maintained by @carolynvs. She has graciously decided to donate it to Kubernetes.

Subproject go-open-service-broker-client

Go Open Service Broker Client is a Go client for communicating with service brokers which implements the Open Service Broker API.

We use it in the Service Catalog controllers.

Repository: https://github.com/kubernetes-sigs/go-open-service-broker-client

Previously it was a project created by a Service Catalog contributor, @pmorie. He has graciously decided to donate it to Kubernetes.

What else is new

The repository was moved from **kubernetes-incubator** to **kubernetes-sigs.**

Update your forked project location! ☺

Design Issue - Deleting the Service Instance

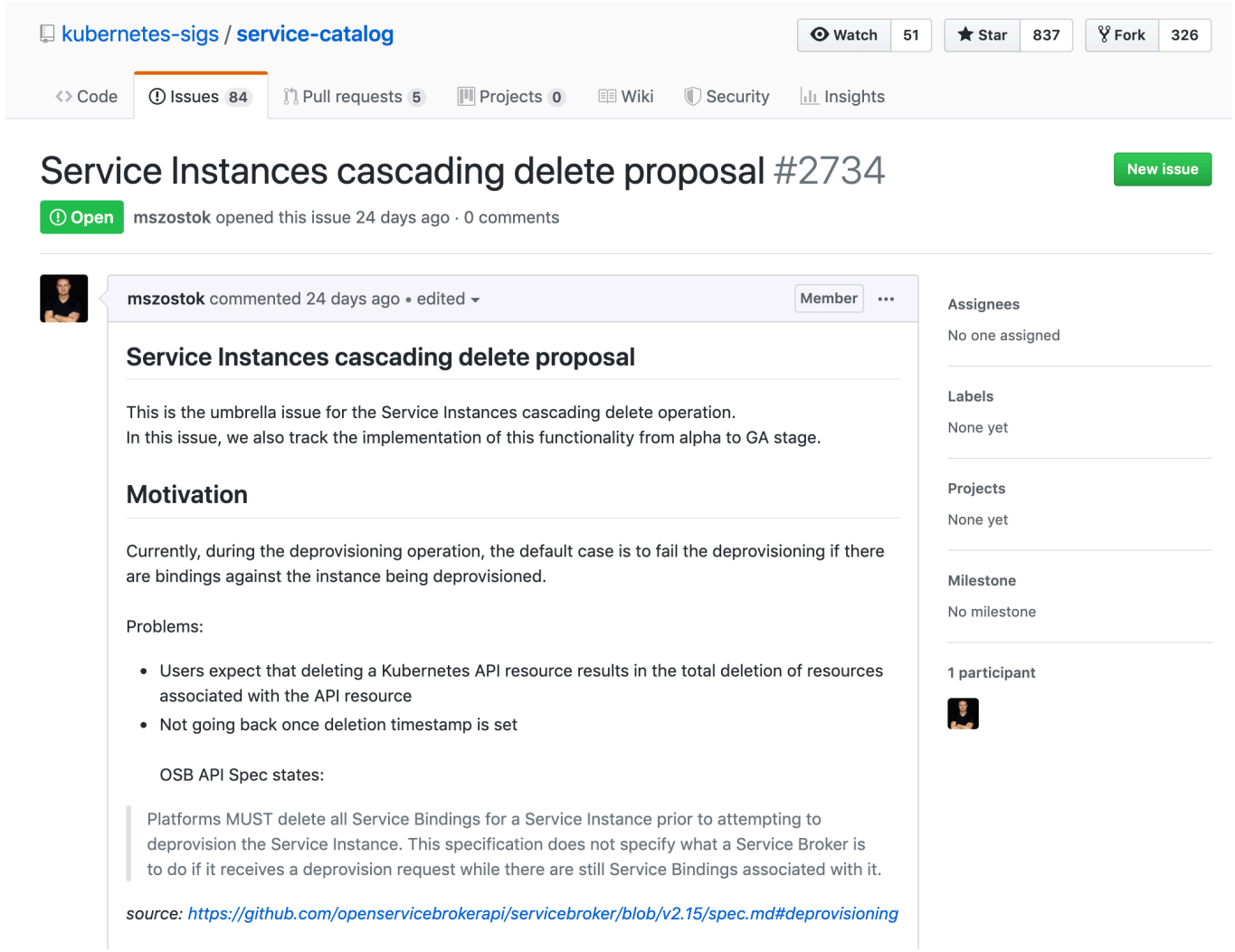
Current approach:

Fail the deprovision operation if the given ServiceInstance still has associated ServiceBindings.

Problem:

The ServiceInstance is **marked for deletion** and this status cannot be reverted. As a result, you cannot update that ServiceInstance and you cannot create a new ServiceBinding for it.

Cascade Deletion Proposal



kubernetes-sigs / service-catalog

Watch 51 | Star 837 | Fork 326

Code | Issues 84 | Pull requests 5 | Projects 0 | Wiki | Security | Insights

Service Instances cascading delete proposal #2734

New issue

Open mszostok opened this issue 24 days ago · 0 comments

mszostok commented 24 days ago • edited — Member

Service Instances cascading delete proposal

This is the umbrella issue for the Service Instances cascading delete operation.
In this issue, we also track the implementation of this functionality from alpha to GA stage.

Motivation

Currently, during the deprovisioning operation, the default case is to fail the deprovisioning if there are bindings against the instance being deprovisioned.

Problems:

- Users expect that deleting a Kubernetes API resource results in the total deletion of resources associated with the API resource
- Not going back once deletion timestamp is set

  OSB API Spec states:

> Platforms MUST delete all Service Bindings for a Service Instance prior to attempting to deprovision the Service Instance. This specification does not specify what a Service Broker is to do if it receives a deprovision request while there are still Service Bindings associated with it.

source: https://github.com/openservicebrokerapi/servicebroker/blob/v2.15/spec.md#deprovisioning

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

1 participant

Alternative?

Implement a validation admission webhook which will block the DELETION request if a given ServiceInstance has associated ServiceBindings.

“We have a strategic plan
it's called doing things”

Next plans – milestone 0.3.0

- Documentation enhancements (compliance with official guidelines, doc structure clean-up , etc.)
- Compliance with the new Open Service Broker API 2.15
- CI pipelines clean-up (probably get rid of Travis and use only Prow)
- Migration of Service Catalog resources under the SIG control
  - https://svcat.io → https://svcat.sigs.k8s.io
  - https://download.svcat.sh → https://download.svcat.sigs.k8s.io *(final URL may differ)*
- Decision on the future of the PodPresets functionality

General Info

Become a Service Catalog contributor! ☺

Regular SIG Meeting: Mondays at 9:00AM PT (Pacific Time) (weekly).
Convert to your timezone.

General Info

- **Chairs**
  - Jonathan Berkhahn - jaberkha@us.ibm.com - @jberkhahn
  - Mateusz Szostok - mateusz.szostok@sap.com - @mszostok
- **Home page:** svc-cat.io
- **GitHub repo:** github.com/kubernetes-sigs/service-catalog
- **Slack channel:** kubernetes.slack.com/messages/sig-service-catalog
- **Mail List:** groups.google.com/forum/#!forum/kubernetes-sig-service-catalog
- **OSB API Spec:** openservicebrokerapi.org

Contact information

Mateusz Szostok

Senior Software Engineer
at SAP Labs Poland

Slack slack.k8s.io

Slack slack.kyma-project.io

Github github.com/mszostok

Twitter twitter.com/m_szostok

Linkedin linkedin.com/in/mszostok