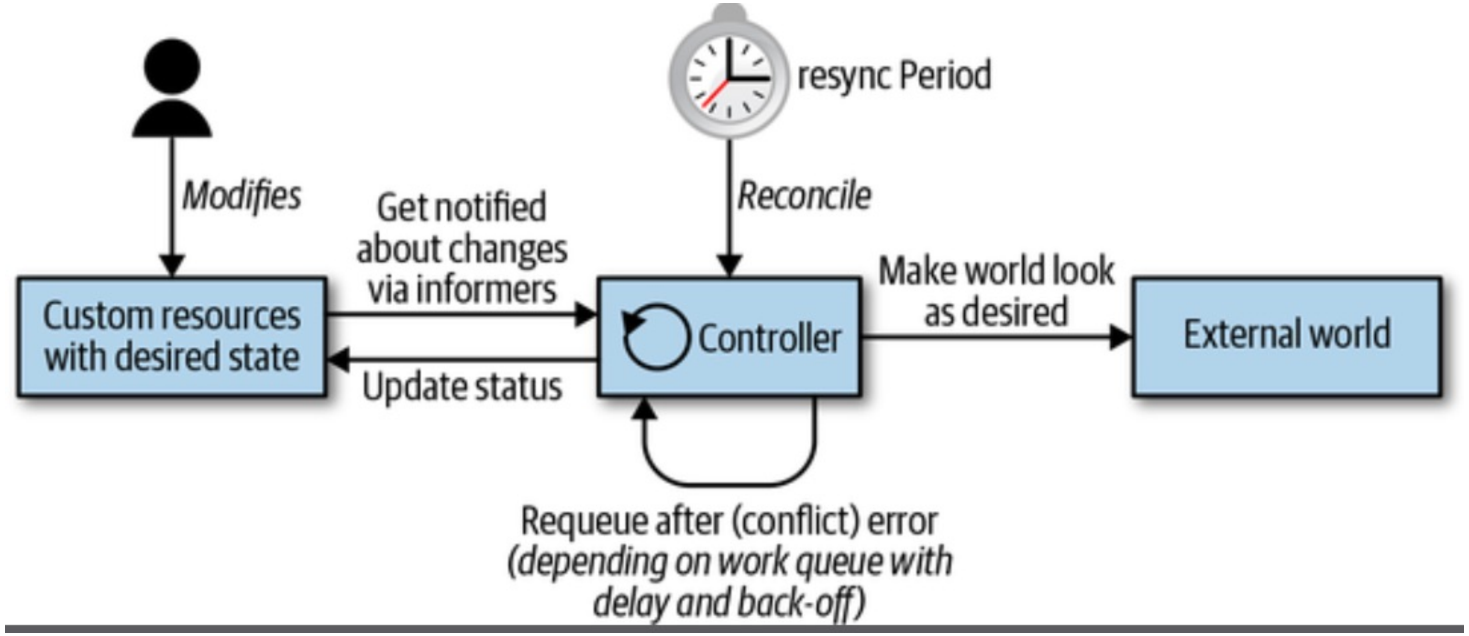
基于client-go的K8s 控制器

神奇代码在哪里k8s篇第4期

控制器

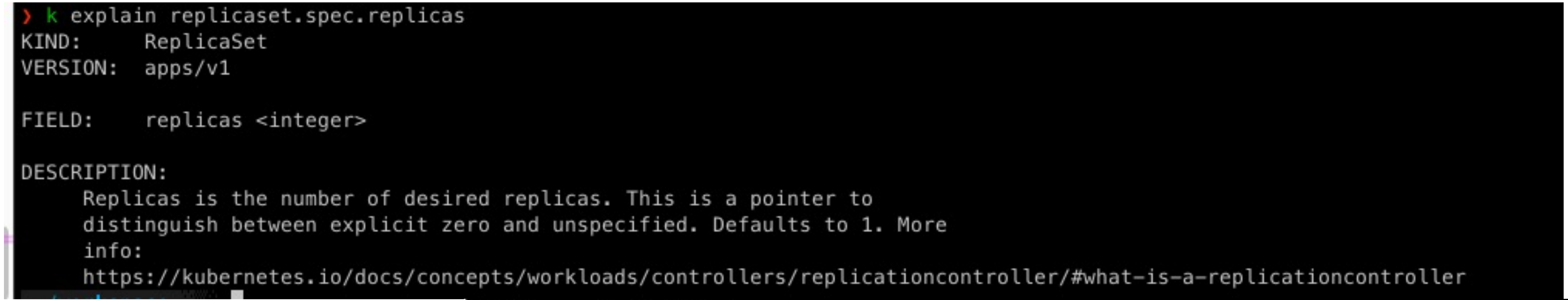
- Controller (控制器) 至少追踪一种类型的 Kubernetes 资源。这些 对象 有一个代表期望状态的 spec 字段。该资源的控制器负责确保其**当前状态**接近**期望状态**。 [1]


resync Period
Modifies
Reconcile
Get notified about changes via informers
Make world look as desired
Custom resources with desired state
Controller
External world
Update status
Requeue after (conflict) error (depending on work queue with delay and back-off)

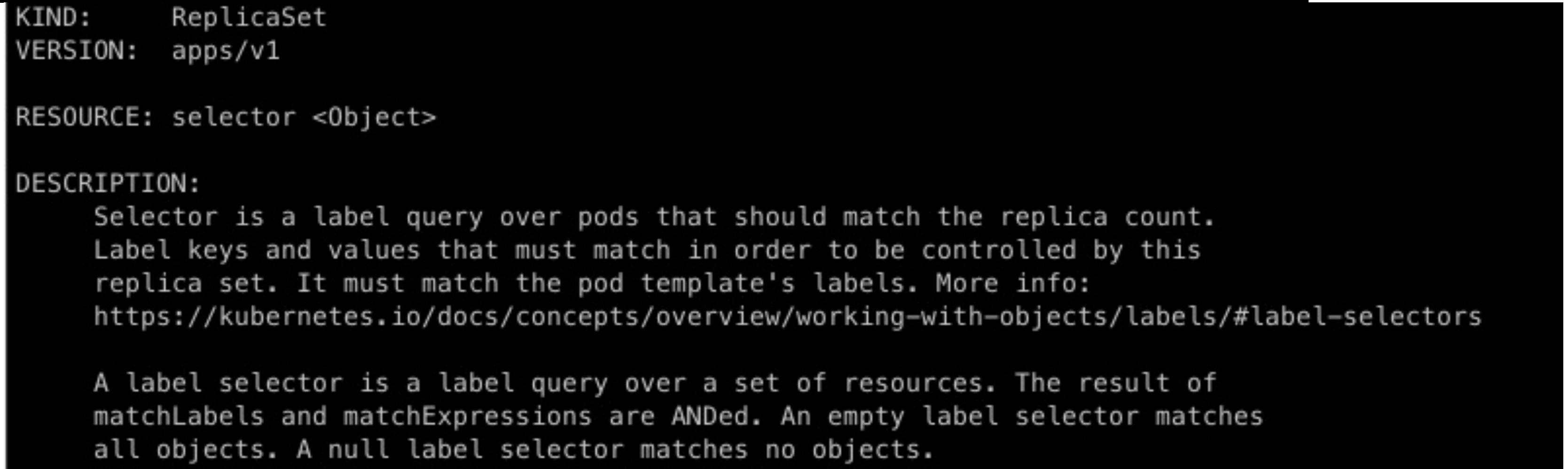
[2]

控制器的典型例子 – replicaset controller

- ReplicaSet 控制器负责保证有 desired number of pods 来match所规定的selector
  - 如果match selector的pod少于所预期，则增加pod
  - 反之，减少pod



```
❯ k explain replicaset.spec.replicas
KIND:     ReplicaSet
VERSION:  apps/v1

FIELD:    replicas <integer>

DESCRIPTION:
     Replicas is the number of desired replicas. This is a pointer to
     distinguish between explicit zero and unspecified. Defaults to 1. More
     info:
     https://kubernetes.io/docs/concepts/workloads/controllers/replicationcontroller/#what-is-a-replicationcontroller
```

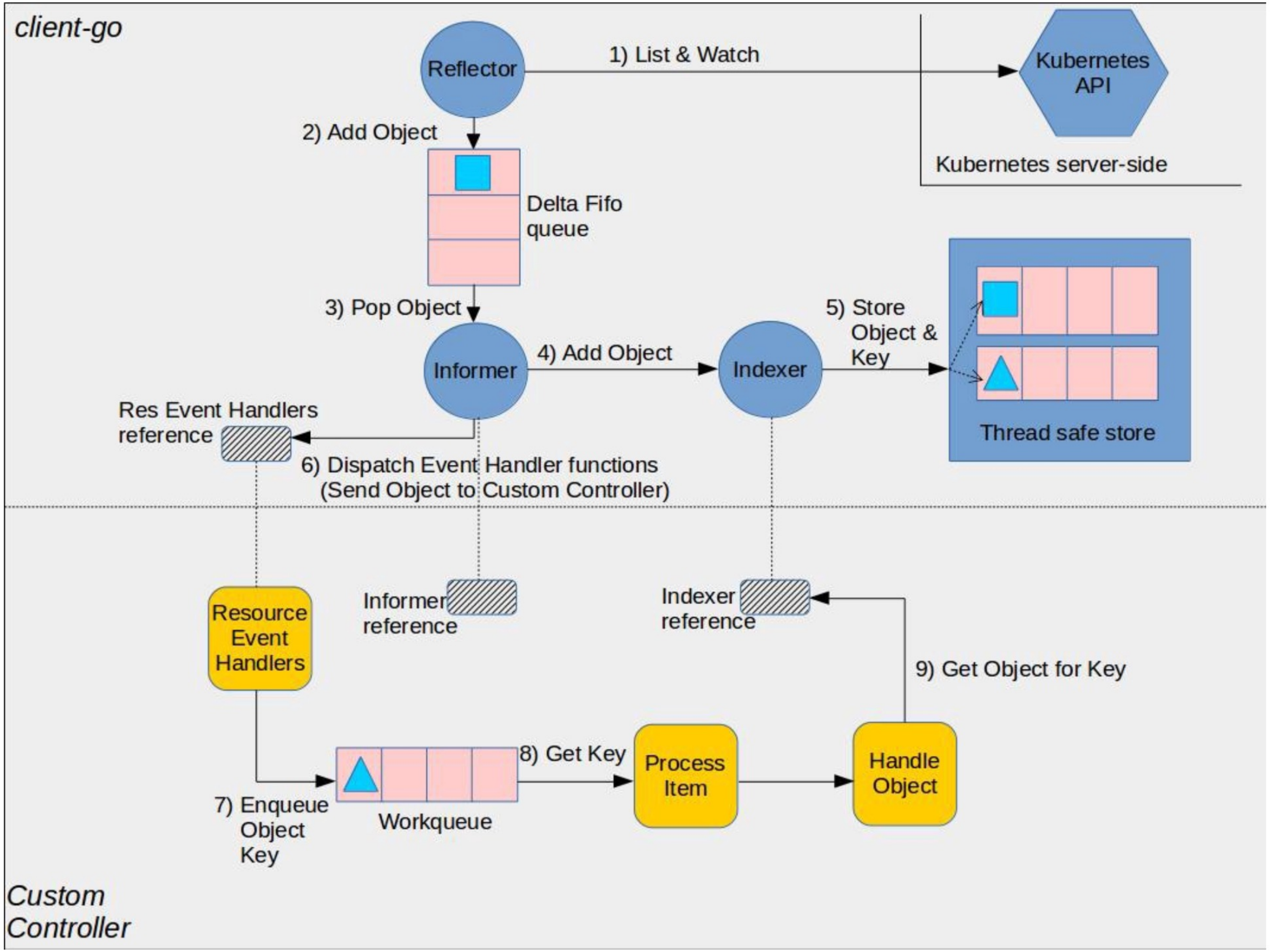


```
KIND:     ReplicaSet
VERSION:  apps/v1

RESOURCE: selector <Object>

DESCRIPTION:
     Selector is a label query over pods that should match the replica count.
     Label keys and values that must match in order to be controlled by this
     replica set. It must match the pod template's labels. More info:
     https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors

     A label selector is a label query over a set of resources. The result of
     matchLabels and matchExpressions are ANDed. An empty label selector matches
     all objects. A null label selector matches no objects.
```

控制器的源码 – replicaset controller

- ReplicaSet 控制器监听 replicasets 和 pods, 把他们的所对应的key放在队列里面去
  - https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L156
- 处理 replicaset的更新事件，首先使用key来获取所更新的object, 也就是我们的replicaset
  - https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L674
- 获取当前replicaset的namespace下面所有的active pods
  - https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L703
- 比较预期的replica num和当前pods的数目，做出相应的行为
  - https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L559
- 计算并更新当前replicaset的状态
  - https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L716

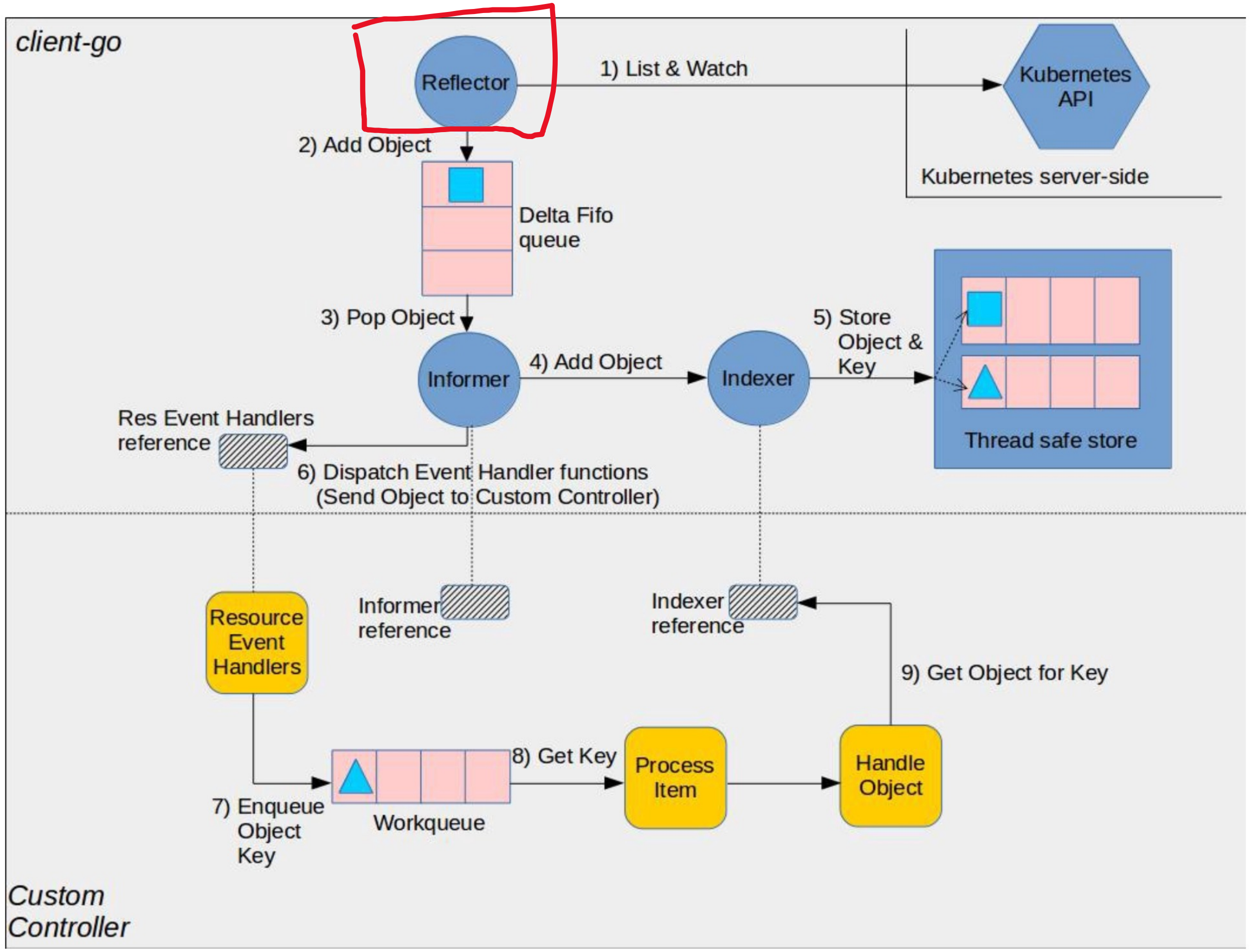
大体架构


client-go
Reflector
1) List & Watch
Kubernetes API
Kubernetes server-side
2) Add Object
Delta Fifo queue
3) Pop Object
Informer
4) Add Object
Indexer
5) Store Object & Key
Thread safe store
Res Event Handlers reference
6) Dispatch Event Handler functions (Send Object to Custom Controller)
Resource Event Handlers
Informer reference
Indexer reference
9) Get Object for Key
7) Enqueue Object Key
Workqueue
8) Get Key
Process Item
Handle Object
Custom Controller

[3]

Reflector

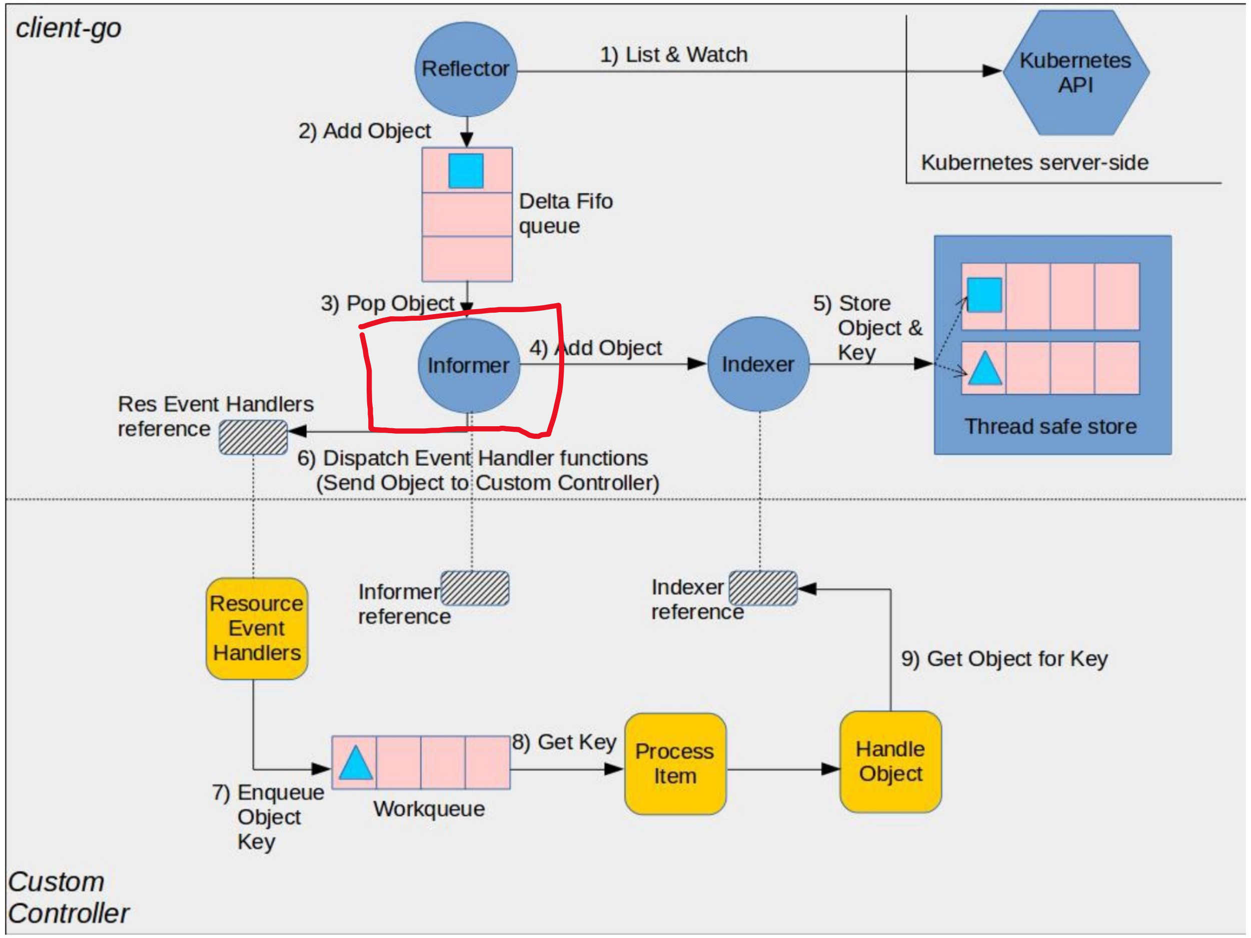
Reflector, inputs data into the local Delta FIFO queue after calling the list/watch method.[4]



[3]

Informer

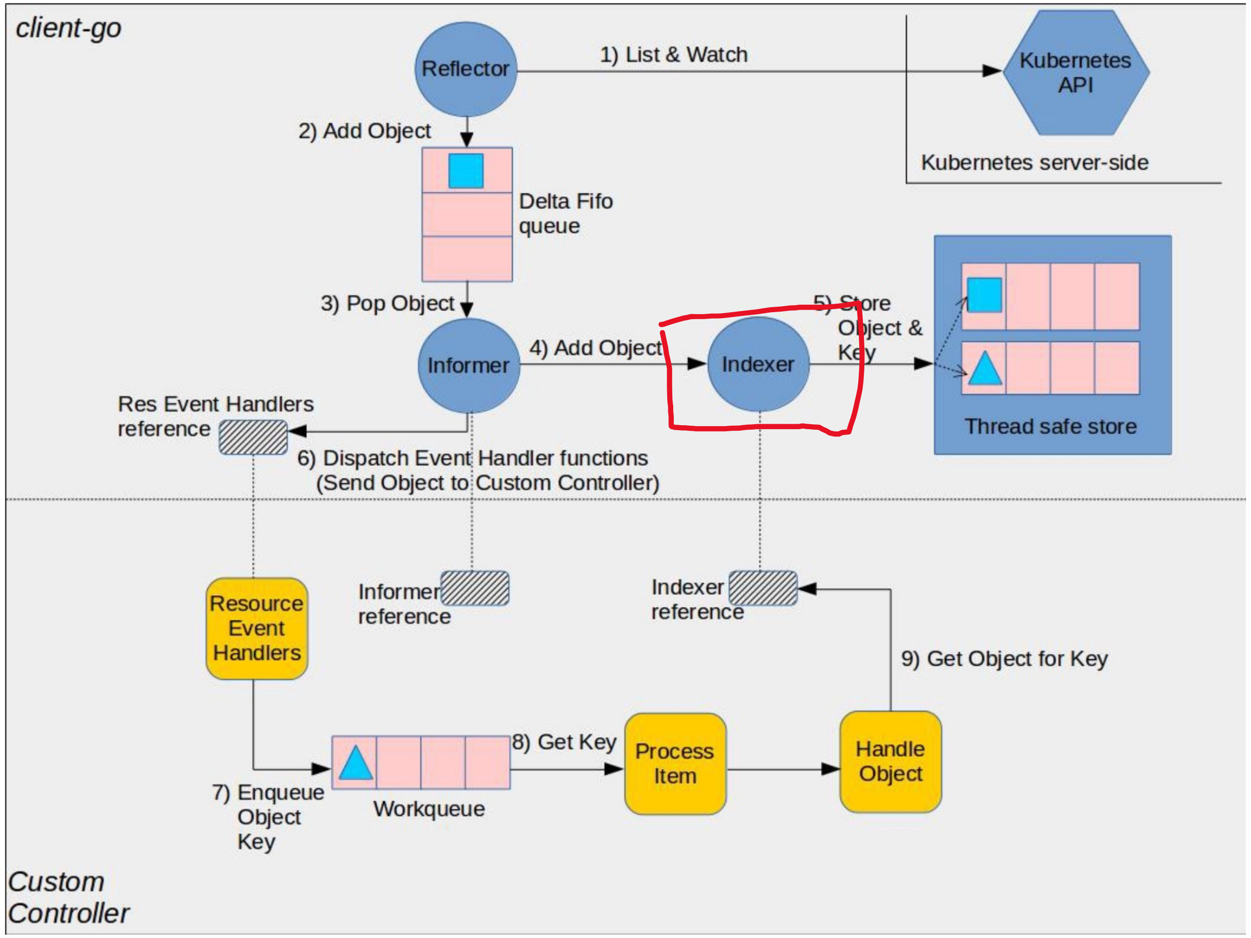
Informer, reads data, adds data to Indexer, and distributes data to specific eventhandler according to the data type(ADD, DELETE, UPDATE). [4]



[3]

Indexer

Indexer, maintains a map to cache data.[4]

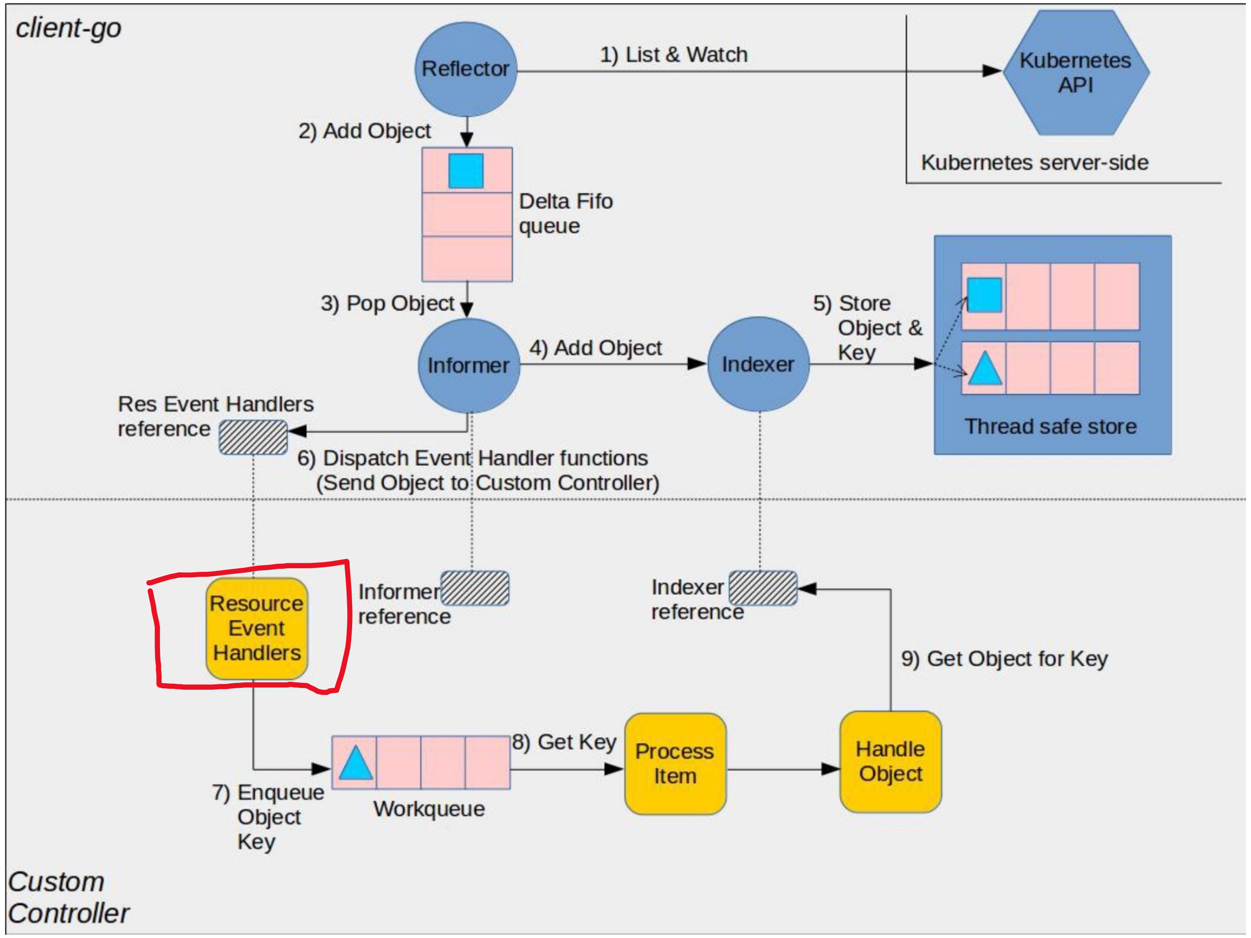


[3]

Controller - Enqueue

Get Object key from the informer reference.

Enqueue it to the workqueue.

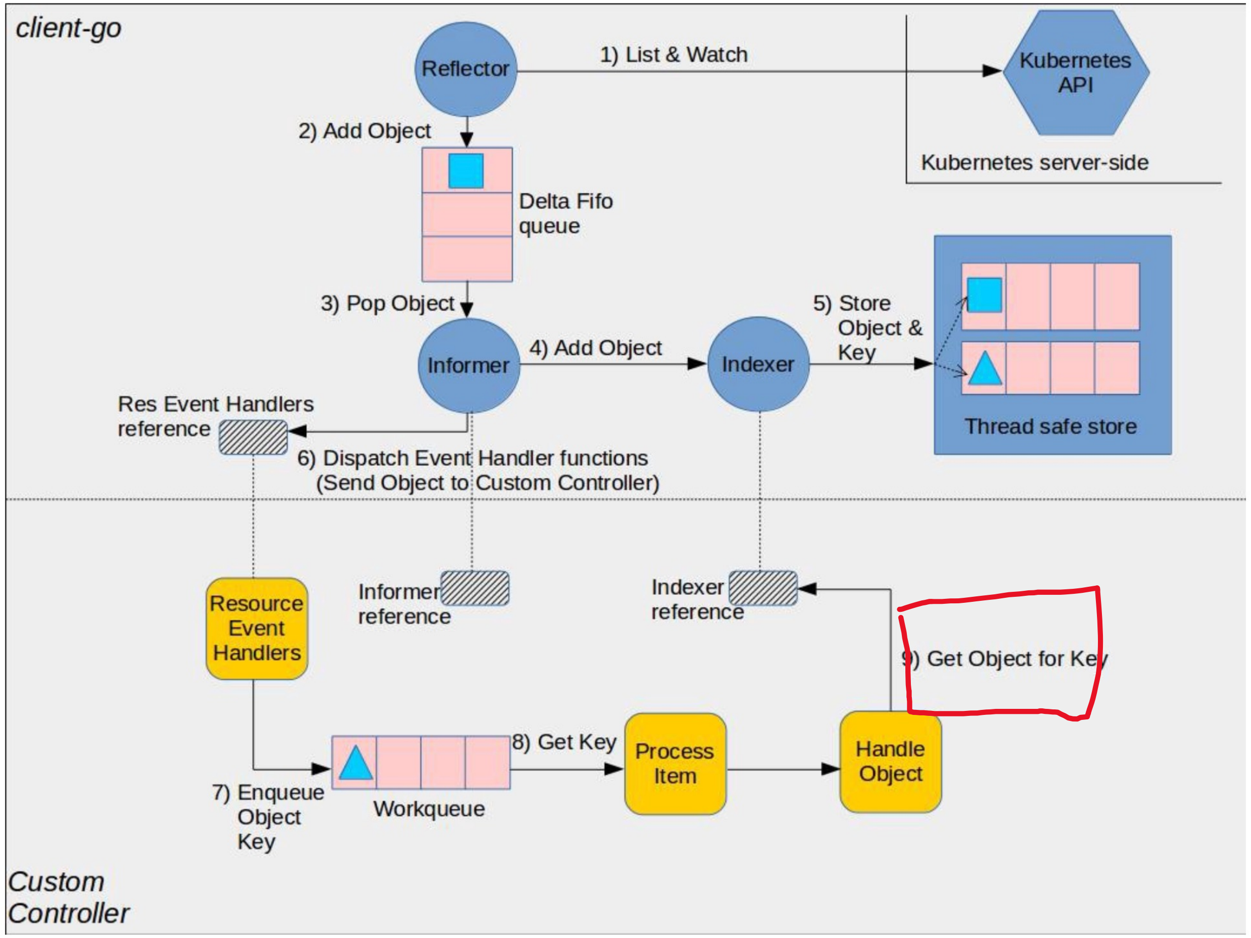


[3]

Controller- Dequeue

Worker goroutines will deque and retrieve the key.

Use key to get back the object using indexer reference.



[3]

具体要怎么用client-go写一个k8s控制器呢？

一个简单的sample controller

- Code walkthrough
  - https://github.com/kubernetes/sample-controller

怎么优雅地使用client-go?

用client-go写控制器最佳做法 (1/4)

- 启动worker goroutine之前， 先确保cache is synced
  - Replicaset controller 例子
    https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L208
  - Sample controller 例子
    - https://github.com/HanFa/sample-controller/blob/735d68047545e2fd1fd2bf3c04526f2cf9614e7c/controller.go#L160

用client-go写控制器最佳做法 (2/4)

- 从cache，或者说informer里拿到的object, 如果要写入，一定要先进行deepcopy，避免cache里面的object被污染
  - Replicaset controller 例子 https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L712
  - Sample controller 例子
    - https://github.com/kubernetes/sample-controller/blob/f42769d51c5abf1dcc8f887fe1086b14bff6a573/controller.go#L325

用client-go写控制器最佳做法 (3/4)

- Informer的周期性resync机制会导致重复冗余的update事件，用ResourceVersion进行early return
  - Replicaset controller 例子 https://github.com/kubernetes/kubernetes/blob/b7b824bb9e01edfb39411b8c139eab592af0d667/pkg/controller/replicaset/replica_set.go#L420
  - Sample controller 例子
    - https://github.com/kubernetes/sample-controller/blob/f42769d51c5abf1dcc8f887fe1086b14bff6a573/controller.go#L134

用client-go写控制器最佳做法 (4/4)

- 使用shareinformer来让多个informer共享一个cache
- 使用NewSharedInformerFactory来新建informer的reference给控制器使用
- https://github.com/HanFa/sample-controller/blob/b3876539afadc37d7b8fc6bd3a2cfd61fb39df4a/main.go#L62

感谢收看，欢迎关注加一键三连~

弯曲的一条虫

弯曲的一条虫，爱撸代码的一条虫 VMware@Palo Alto

B站空间：https://space.bilibili.com/1557732

如果你有纠正或者是建议，欢迎在https://github.com/HanFa/learn-k8s/issues提新的issue。

References

- [1] 控制器， https://kubernetes.io/zh/docs/concepts/architecture/controller/
- [2] Programming Kubernetes.
- [3] Client-go controller interaction. https://github.com/kubernetes/sample-controller/blob/master/docs/images/client-go-controller-interaction.jpeg
- [4] Explore client-go Informer Patterns. https://medium.com/codex/explore-client-go-informer-patterns-4415bb5f1fbd