we may encounter Concurrency safety problem, in which case multiple controllers process the same event and lead to inconsistency of the system.

we use "resourcelock" and "leaderelection" in k8s

```
// start the leader election loop
leaderelection.RunOrDie(ctx, leaderelection.LeaderElectionConfig{
    Lock: lock,
    ReleaseOnCancel: true,
    LeaseDuration:   30 * time.Second,
    RenewDeadline:   15 * time.Second,
    RetryPeriod:     5 * time.Second,
    Callbacks: leaderelection.LeaderCallbacks{
        OnStartedLeading: func(ctx context.Context) {
        },
        OnStoppedLeading: func() {
            klog.Infof("leader lost: %s", leaseLockId)
            stopCh <- struct{}{}
        },
        OnNewLeader: func(identity string) {
            if identity == leaseLockId {
                klog.Infof("we are the new leader, lockId:%s", identity)
                stopCh = make(chan struct{})
                kubeInformerFactory.Start(stopCh)
                exampleInformerFactory.Start(stopCh)
                go controller.Run(2, stopCh)
            } else {
                klog.Infof("we(%s) are not the leader: %s", leaseLockId, identit
                stopCh <- struct{}{}
            }
        },
    },
})
```