

# The Manufactory of Kubernetes: The kube-controller-manager

Learn about the kube-controller-manager.

We'll cover the following



- The kube-controller-manager
  - What does the kube-controller-manager do?
  - What's in the kube-controller-manager?
- Summary

## The kube-controller-manager

In this course, we won't make any customizations of the kube-controller-manager. So, this lesson will only give us a basic understanding of what the kube-controller-manager does. At the same time, we could benefit a lot from the design of the kube-controller-manager. This is where the operator pattern originates from.

### What does the kube-controller-manager do?

As we discussed previously, the control plane is responsible for driving the actual state of the system toward the desired state. However, the kube-apiserver focuses on storing resources and providing RESTful services for all clients. The kube-scheduler assigns unassigned pods with the best-matched nodes. In the control plane, we need a component that can do the convergences. This is done by the kube-controller-manager.

The kube-controller-manager performs cluster-level functions. Primarily, it manages a group of controllers that's responsible for reconciling the state of objects—such as Deployment and Service—and performing routine tasks. For instance, a replication controller ensures that the desired number of Pod objects are running healthily in the cluster by scaling up or down when the desired number isn't met.

### What's in the kube-controller-manager?

Below is a code snippet of the kube-controller-manager:

```
1 // Codes from <https://github.com/kubernetes/kubernetes/blob/master/cmd/kube-controller-manager/app/cont
2 // NewControllerInitializers is a public map of named controller groups (you can start more than one in
3 // paired to their InitFunc. This allows for structured downstream composition and subdivision.
4 func NewControllerInitializers(loopMode ControllerLoopMode) map[string]InitFunc {
5     controllers := map[string]InitFunc{}
6     controllers["endpoint"] = startEndpointController
7     controllers["endpointslice"] = startEndpointSliceController
8     controllers["endpointslicemirroring"] = startEndpointSliceMirroringController
9     controllers["replicaset"] = startReplicaSetController
10    controllers["deployment"] = startDeploymentController
11    controllers["service"] = startServiceController
12    controllers["daemonset"] = startDaemonSetController
13    controllers["cronjob"] = startCronJobController
14    controllers["job"] = startJobController
15    controllers["pod"] = startPodController
16    controllers["statefulset"] = startStatefulSetController
17    controllers["volume"] = startVolumeController
18    controllers["persistentvolumeclaim"] = startPersistentVolumeClaimController
19    controllers["storageclass"] = startStorageClassController
20    controllers["defaultstorageclass"] = startDefaultStorageClassController
21    controllers["node"] = startNodeController
22    controllers["taint"] = startTaintController
23    controllers["poddisruptionbudget"] = startPodDisruptionBudgetController
24    controllers["horizontalpodautoscaler"] = startHorizontalPodAutoscalerController
25    controllers["verticalpodautoscaler"] = startVerticalPodAutoscalerController
26    controllers["resourcequota"] = startResourceQuotaController
27    controllers["limitrange"] = startLimitRangeController
28    controllers["priorityclass"] = startPriorityClassController
29    controllers["podpriority"] = startPodPriorityController
30    controllers["podpreset"] = startPodPresetController
31    controllers["podsecuritypolicy"] = startPodSecurityPolicyController
32    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
33    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
34    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
35    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
36    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
37    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
38    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
39    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
40    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
41    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
42    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
43    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
44    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
45    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
46    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
47    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
48    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
49    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
50    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
51    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
52    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
53    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
54    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
55    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
56    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
57    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
58    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
59    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
60    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
61    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
62    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
63    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
64    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
65    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
66    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
67    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
68    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
69    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
70    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
71    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
72    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
73    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
74    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
75    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
76    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
77    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
78    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
79    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
80    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
81    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
82    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
83    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
84    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
85    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
86    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
87    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
88    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
89    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
90    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
91    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
92    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
93    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
94    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
95    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
96    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
97    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
98    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
99    controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
100   controllers["podsecuritypolicyexceptions"] = startPodSecurityPolicyExceptionsController
```

```

9     controllers["replicationcontroller"] = startReplicationController
10    controllers["podgc"] = startPodGCController
11    controllers["resourcequota"] = startResourceQuotaController
12    controllers["namespace"] = startNamespaceController
13    controllers["serviceaccount"] = startServiceAccountController
14    controllers["garbagecollector"] = startGarbageCollectorController
15    controllers["daemonset"] = startDaemonSetController
16    controllers["job"] = startJobController
17    controllers["deployment"] = startDeploymentController
18    controllers["replicaset"] = startReplicaSetController
19    controllers["horizontalpodautoscaling"] = startHPAController
20    controllers["disruption"] = startDisruptionController
21    controllers["statefulset"] = startStatefulSetController
22    controllers["cronjob"] = startCronJobController
23    controllers["csrsigning"] = startCSRSigningController
24    controllers["csrapproving"] = startCSRApprovingController
25    controllers["csrcleaner"] = startCSRCleanerController
26    controllers["ttl"] = startTTLController
27    controllers["bootstrapsigner"] = startBootstrapSignerController
28    controllers["tokencleaner"] = startTokenCleanerController
29    controllers["nodeipam"] = startNodeIpamController
30    controllers["nodelifecycle"] = startNodeLifecycleController
31    if !os.Mode == ToolModeCloud {

```

Controller initializers in the kube-controller-manager

More than 30 controllers are running inside the kube-controller-manager. It's evident what each of these controllers does from its name. We can enable or disable some controllers as needed. This can be done with the flag `--controllers` of the kube-controller-manager.

Let's take a closer look at one controller, the ServiceController:

```

1  // Codes from <https://github.com/kubernetes/kubernetes/blob/b74d023e70d6064c7f3f77031e7d26ec38497fc9/cr
2  func startServiceController(ctx context.Context, controllerContext ControllerContext) (controller.Interface, error) {
3      serviceController, err := servicecontroller.New(
4          controllerContext.Cloud,
5          controllerContext.ClientBuilder.ClientOrDie("service-controller"),
6          controllerContext.InformerFactory.Core().V1().Services(),
7          controllerContext.InformerFactory.Core().V1().Nodes(),
8          controllerContext.ComponentConfig.KubeCloudShared.ClusterName,
9          utilfeature.DefaultFeatureGate,
10     )
11     if err != nil {
12         // This error shouldn't fail. It lives like this as a legacy.
13         klog.Errorf("Failed to start service controller: %v", err)
14         return nil, false, nil
15     }
16     go serviceController.Run(ctx, int(controllerContext.ComponentConfig.ServiceController.ConcurrentServices), int(controllerContext.ComponentConfig.ServiceController.ConcurrentNodes))
17     return nil, true, nil
18 }

```

The ServiceController in the kube-controller-manager

This controller runs as a standalone goroutine to perform actual work for deployed Services. From here, we see that the kube-controller-manager starts multiple, distinct goroutines in the background.

This is to watch the kube-apiserver for changes (including creating, updating, and deleting) to separate resources and perform operations for each change.

## Summary

Every controller uses the WATCH mechanism to get every change from the informer. Controllers also run a reconciliation loop that keeps reconciling the actual state with the desired state. Then, it will report back the newest status to the kube-apiserver. In general, controllers never talk to each other. Each controller watches the kube-apiserver to get changes purely on self-interested and responsible resources. This is a very illustrative example on how to design and write a good controller and/or operator. The golden rule for this is “Each one does things in their own way.”

[← Back](#)

The Conductor of Kubernetes: The kube-scheduler

[✓](#)

[Next →](#)

Kubernetes Worker Nodes

---