**Admission Controllers (continued)**

**Namespace Lifecycle:**

This admission controller enforces that a Namespace that is undergoing termination cannot have new objects created in it, and ensures that requests in a non-existent Namespace are rejected. This admission controller also prevents deletion of three system reserved namespaces default, kube-system, kube-public.

A Namespace deletion kicks off a sequence of operations that remove all objects (pods, services, etc.) in that namespace. In order to enforce integrity of that process, we strongly recommend running this admission controller.

**Limit Ranger:**

This admission controller will observe the incoming request and ensure that it does not violate any of the constraints enumerated in the LimitRange object in a Namespace. If you are using LimitRange objects in your Kubernetes deployment, you MUST use this admission controller to enforce those constraints. LimitRanger can also be used to apply default resource requests to Pods that don't specify any; currently, the default LimitRanger applies a 0.1 CPU requirement to all Pods in the default namespace.

**Persistent Volume Label**

Default Storage Class:This admission controller observes creation of PersistentVolumeClaim objects that do not request any specific storage class and automatically adds a default storage class to them. This way, users that do not request any special storage class do not need to care about them at all and they will get the default one.

**Default Toleration Seconds:**

This admission controller sets the default forgiveness toleration for pods to tolerate the taints notready:NoExecute and unreachable:NoExecute based on the k8s-apiserver input parameters default-not-ready-toleration-seconds and default-unreachable-toleration-seconds if the pods don't already have toleration for taints node.kubernetes.io/not-ready:NoExecute or node.kubernetes.io/unreachable:NoExecute. The default value for default-not-ready-toleration-seconds and default-unreachable-toleration-seconds is 5 minutes.

**Resource Quota:**

This admission controller will observe the incoming request and ensure that it does not violate any of the constraints enumerated in the ResourceQuota object in a Namespace. If you are using ResourceQuota objects in your Kubernetes deployment, you MUST use this admission controller to enforce quota constraints.

**Priority:**

The priority admission controller uses the priorityClassName field and populates the integer value of the priority. If the priority class is not found, the Pod is rejected.

**Mutating Admission Webhook:**

This admission controller calls any mutating webhooks which match the request. Matching webhooks are called in serial; each one may modify the object if it desires.

This admission controller (as implied by the name) only runs in the mutating phase.

If a webhook called by this has side effects (for example, decrementing quota) it must have a reconciliation system, as it is not guaranteed that subsequent webhooks or validating admission controllers will permit the request to finish.

If you disable the MutatingAdmissionWebhook, you must also disable the MutatingWebhookConfiguration object in the admissionregistration.k8s.io/v1beta1 group/version via the --runtime-config flag (both are on by default in versions >= 1.9).

**Validating Admission Webhook:**

This admission controller calls any validating webhooks which match the request. Matching webhooks are called in parallel; if any of them rejects the request, the request fails. This admission controller only runs in the validation phase; the webhooks it calls may not mutate the object, as opposed to the webhooks called by the MutatingAdmissionWebhook admission controller.

If a webhook called by this has side effects (for example, decrementing quota) it must have a reconciliation system, as it is not guaranteed that subsequent webhooks or other validating admission controllers will permit the request to finish.

If you disable the ValidatingAdmissionWebhook, you must also disable the ValidatingWebhookConfiguration object in the admissionregistration.k8s.io/v1beta1 group/version via the --runtime-config flag (both are on by default in versions 1.9 and later).

Admission controllers limit requests to create, delete, modify or connect to (proxy). They do not support read requests.

The admission control process proceeds in two phases. In the first phase, mutating admission controllers are run. In the second phase, validating admission controllers are run. Note again that some of the controllers are both.

**Why do I need them?**

Many advanced features in Kubernetes require an admission controller to be enabled in order to properly support the feature. As a result, a Kubernetes API server that is not properly configured with the right set of admission controllers is an incomplete server and will not support all the features you expect

**How do I turn on an admission controller?**

The Kubernetes API server flag enable-admission-plugins takes a comma-delimited list of admission control plugins to invoke prior to modifying objects in the cluster. For example, the following command line enables the NamespaceLifecycle and the LimitRanger admission control plugins:

kube-apiserver --enable-admission-plugins=NamespaceLifecycle,LimitRanger ...