

A VPA definition can also have a resource policy that influences how VPA computes the recommended resources (e.g., by setting per container lower and upper resource boundaries).

Depending on which `.spec.updatePolicy.updateMode` is configured, the VPA involves different system components. All three VPA components—recommender, admission plugin, and updater—are decoupled, independent, and can be replaced with alternative implementations. The module with the intelligence to produce recommendations is the recommender, which is inspired by Google’s Borg system. The current implementation analyzes the actual resource usage of a container under load for a certain period (by default, eight days), produces a histogram, and chooses a high percentile value for that period. In addition to metrics, it also considers resource and specifically memory-related Pod events such as evictions and `OutOfMemory` events.

In our example we chose `.spec.updatePolicy.updateMode` equals `Off`, but there are two other options to choose from, each with a different level of potential disruption on the scaled Pods. Let’s see how different values for `updateMode` work, starting from nondisruptive to a more disruptive order:

`updateMode: Off`

The VPA *recommender* gathers Pod metrics and events and then produces recommendations. The VPA recommendations are always stored in the `status` section of the VPA resource. However, this is how far the `Off` mode goes. It analyzes and produces recommendations, but it does not apply them to the Pods. This mode is useful for getting insight on the Pod resource consumption without introducing any changes and causing disruption. That decision is left for the user to make if desired.

`updateMode: Initial`

In this mode, the VPA goes one step further. In addition to the activities performed by the recommender component, it also activates the VPA admission plugin, which applies the recommendations to newly created Pods only. For example, if a Pod is scaled manually through an HPA, updated by a Deployment, or evicted and restarted for whatever reason, the Pod’s resource request values are updated by the VPA Admission Controller.

This controller is a *mutating admission plugin* that overrides the requests of new Pods matching the VPA label selector. This mode does not restart a running Pod, but it is still partially disruptive because it changes the resource request of newly created Pods. This in turn can affect where a new Pod is scheduled. What’s more, it is possible that after applying the recommended resource requests, the Pod is scheduled to a different node, which can have unexpected consequences. Or worse, the Pod might not be scheduled to any node if there is not enough capacity on the cluster.