Admission Controller

> The Admission Controller is a beta feature. Please read the documentation carefully.

Kubecost's Admission Controller is a tool which leverages Kubecost's Predict API to display expected future costs for changes made to CPU and RAM in your workloads. This can be used to anticipate future spend before resources have been allocated.

The Admission Controller exists in Kubecost's cost-analyzer pod and uses a `ValidatingAdmissionWebhook` to send back cost data on every update you make to deployments in your Kubernetes clusters. The Predict API then determines the differences in cost, and provides that information in a convenient table in your terminal. Non-`kubectl` interactions will still be able to receive cost prediction data.

Installing the Admission Controller

Prerequisites

Before installing the Admission Controller, make sure you have `kubectl` installed.

Installation

1. This command sets up the TLS key and certificate for the Admission Controller.

```
$ git clone https://github.com/kubecost/cost-analyzer-helm-chart.git
$ cd cost-analyzer-helm-chart/cost-analyzer
$ kubectl create namespace kubecost
$ ./scripts/create-admission-controller-tls.sh kubecost
```

2. This command will update your existing Kubecost application by enabling the Admission Controller.

```
$ helm upgrade --install kubecost \
  --repo https://kubecost.github.io/cost-analyzer/ cost-analyzer \
  --namespace kubecost \
  --set kubecostAdmissionController.enabled=true \
  -f values.yaml
```

You may need to wait several minutes for the controller to activate. You can check the status of the Admission Controller with `kubectl get service -n kubecost`, when the Admission Controller has been installed in the default `kubecost` namespace. Look for `webhook-server` to confirm a successfull install.

Using the Admission Controller

When a Deployment is updated, the Kubernetes API will send requests containing Deployment information to the Kubecost pod, which will then be read for number of replicas, CPU requests, and RAM requests to calculate a monthly cost estimate. That estimate will be reported back to the client making the update. If the Kubecost pod is unable to respond to this request, the deployment will be added to the cluster without any information being sent back.

To validate, you can run `kubectl edit deployment -n kubecost`, which should open a text file containing your deployment in the namespace `kubecost`. Modify the CPU or RAM requests as desired, then save the file. In your terminal, your output will display CPU/RAM unit hours, total cost, and the cost difference (`diff` column).