Understanding Kubernetes Admission Control

10m 52s left

100% **Open Cloud Environment**
Setup completed
Average setup time: 2m 26s

100% **Open Code Environment**
Setup completed

Credentials

Account ID
97118186[illegible] Copy

Username
student Copy

Password
Ca1_Yyt0[illegible] Copy

Region
US West [illegible] Copy

PEM Download PPK Download

Bastion Public Ip
34.212.26[illegible] Copy

Cluster SSH
ssh ubunt[illegible]

Introduction

Admission control is the final layer of access control in the Kubernetes API. A Kubernetes cluster can have many admission control modules and the modules used by a cluster can be configured. None of the modules in use must reject the request for a request to be allowed. For example, the ResourceQuota admission controller enforces resource quotas configured in a Namespace. Admission controllers can also modify requests before they are accepted. For example, the LimitRanger admission controller allows you to specify default cpu and memory limits for Pods that don't specify any. A full treatment of admission controllers and how they are configured is outside of the scope of this lab. A list of admission controllers is available in the official Kubernetes documentation.

You will explore the admission controllers in the cluster in this lab step.

Instructions

1. `exec` into the Kubernetes API Server Pod and display the `kube-apiserver` help to see what admission controllers are available and enabled by default:

Copy code

```
kubectl exec -n kube-system kube-apiserver-ip-10-0-0-100.us-west
```

```
      --enable-admission-plugins strings       admission plugins that should be enabled in addition to default enabled ones
(NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, Priority, DefaultTolerationSeconds, DefaultStorageC
lass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, Cert
ificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota). Comma-
delimited list of admission plugins: AlwaysAdmit, AlwaysDeny, AlwaysPullImages, CertificateApproval, CertificateSigning, Cer
tificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, DenyServiceExternalIPs, Even
tRateLimit, ExtendedResourceToleration, ImagePolicyWebhook, LimitPodHardAntiAffinityTopology, LimitRanger, MutatingAdmission
Webhook, NamespaceAutoProvision, NamespaceExists, NamespaceLifecycle, NodeRestriction, OwnerReferencesPermissionEnforcement,
 PersistentVolumeClaimResize, PersistentVolumeLabel, PodNodeSelector, PodSecurityPolicy, PodTolerationRestriction, Priority,
 ResourceQuota, RuntimeClass, SecurityContextDeny, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition, Vali
datingAdmissionWebhook. The order of plugins in this flag does not matter.
```

There are 17 admission controllers enabled by default, although they can be disabled if desired. To view any non-default admission controlle[illegible] enabled using the `--enable-admission-plugins` option, connect to a control-plane node where the API server is running.

Support

```
ssh 10.0.0.100 -oStrictHostKeyChecking=no
```

3. List all the processes on the machine and use grep to select the `kube-apiserver` process and highlight the `--enable-admission-plugins` option:

Copy code

```
ps -ef | grep kube-apiserver | grep enable-admission-plugins
```

```
--enable-admission-plugins=NodeRestriction
```

The **NodeRestriction** admission controller is enabled in addition to the default admission controllers in this cluster.

Summary

In this lab step, you learned about admission control as the final layer of access control in the Kubernetes API. You also investigated which admission controllers are enabled in the cluster.

Need help? Contact our support team

Did you like this step?

End Lab Back Submit