

Nine Steps to Maximizing Kubernetes Security

1



UPGRADE TO THE LATEST VERSION

Upgrades and support can become more difficult the farther behind you fall, so plan to upgrade at least once per quarter. Using a managed Kubernetes provider can make upgrades very easy.

2



ENABLE ROLE-BASED ACCESS CONTROL (RBAC)

If you have upgraded since Kubernetes 1.6, double-check your settings. You must both enable RBAC and disable legacy Attribute-Based Access Control (ABAC).

3



USE NAMESPACES TO ESTABLISH SECURITY BOUNDARIES

It's easier to apply security controls such as Network Policies when different types of workloads are deployed in separate namespaces.

4



SEPARATE SENSITIVE WORKLOADS

Run sensitive workloads on a dedicated set of machines to reduce the risk of a sensitive application being accessed through a less-secure application.

5



SECURE CLOUD METADATA ACCESS

Sensitive metadata, such as kubelet admin credentials, can sometimes be stolen or misused to escalate privileges in a cluster.

6



CREATE AND DEFINE CLUSTER NETWORK POLICIES

Network Policies allow you to control network access into and out of your containerized applications. Make sure that you have a networking provider that supports this resource.

7



RUN A CLUSTER-WIDE POD SECURITY POLICY

A Pod Security Policy sets defaults for how workloads are allowed to run in your cluster. Define a policy and enable the Pod Security Policy admission controller.

8



HARDEN NODE SECURITY

- Ensure the host is secure and configured correctly.
- Control network access to sensitive ports.
- Minimize administrative access to Kubernetes nodes.

9



TURN ON AUDIT LOGGING

Make sure you have audit logs enabled and are monitoring them for anomalous or unwanted API calls, especially any authorization failures.

GET DOWNLOADABLE CODE SNIPPETS AND MORE DETAILS IN **THE FULL WHITE PAPER**

