Security

Concepts for keeping your cloud-native workload secure.

This section of the Kubernetes documentation aims to help you learn to run workloads more securely, and about the essential aspects of keeping a Kubernetes cluster secure.

Kubernetes is based on a cloud-native architecture, and draws on advice from the CNCF about good practice for cloud native information security.

Read Cloud Native Security and Kubernetes for the broader context about how to secure your cluster and the applications that you're running on it.

Kubernetes security mechanisms

Kubernetes includes several APIs and security controls, as well as ways to define policies that can form part of how you manage information security.

Control plane protection

A key security mechanism for any Kubernetes cluster is to control access to the Kubernetes API.

Kubernetes expects you to configure and use TLS to provide data encryption in transit within the control plane, and between the control plane and its clients. You can also enable encryption at rest for the data stored within Kubernetes control plane; this is separate from using encryption at rest for your own workloads' data, which might also be a good idea.

Secrets

The Secret API provides basic protection for configuration values that require confidentiality.

Workload protection

Enforce Pod security standards to ensure that Pods and their containers are isolated appropriately. You can also use RuntimeClasses to define custom isolation if you need it.

Network policies let you control network traffic between Pods, or between Pods and the network outside your cluster.

You can deploy security controls from the wider ecosystem to implement preventative or detective controls around Pods, their containers, and the images that run in them.

Admission control

Admission controllers are plugins that intercept Kubernetes API requests and can validate or mutate the requests based on specific fields in the request. Thoughtfully designing these controllers helps to avoid unintended disruptions as Kubernetes APIs change across version updates. For design considerations, see Admission Webhook Good Practices.

Auditing

Kubernetes audit logging provides a security-relevant, chronological set of records documenting the sequence of actions in a cluster. The cluster audits the activities generated by users, by applications that use the Kubernetes API, and by the control plane itself.

Cloud provider security

> **Note:** Items on this page refer to vendors external to Kubernetes. The Kubernetes project authors aren't responsible for those third-party products or projects. To add a vendor, product or project to this list, read the content guide before submitting a change. More information.

If you are running a Kubernetes cluster on your own hardware or a different cloud provider, consult your documentation for security best practices. Here are links to some of the popular cloud providers' security documentation:

IaaS Provider	Link
Alibaba Cloud	https://www.alibabacloud.com/trust-center
Amazon Web Services	https://aws.amazon.com/security
Google Cloud Platform	https://cloud.google.com/security
Huawei Cloud	https://www.huaweicloud.com/intl/en-us/securecenter/overallsafety
IBM Cloud	https://www.ibm.com/cloud/security
Microsoft Azure	https://docs.microsoft.com/en-us/azure/security/azure-security
Oracle Cloud Infrastructure	https://www.oracle.com/security
Tencent Cloud	https://www.tencentcloud.com/solutions/data-security-and-information-protection
VMware vSphere	https://www.vmware.com/solutions/security/hardening-guides

Policies

You can define security policies using Kubernetes-native mechanisms, such as NetworkPolicy (declarative control over network packet filtering) or ValidatingAdmissionPolicy (declarative restrictions on what changes someone can make using the Kubernetes API).

However, you can also rely on policy implementations from the wider ecosystem around Kubernetes. Kubernetes provides extension mechanisms to let those ecosystem projects implement their own policy controls on source code review, container image approval, API access controls, networking, and more.

For more information about policy mechanisms and Kubernetes, read Policies.

What's next

Learn about related Kubernetes security topics:

- Securing your cluster
- Known vulnerabilities in Kubernetes (and links to further information)
- Data encryption in transit for the control plane
- Data encryption at rest
- Controlling Access to the Kubernetes API
- Network policies for Pods
- Secrets in Kubernetes
- Pod security standards
- RuntimeClasses

Learn the context:

- Cloud Native Security and Kubernetes

Get certified:

- Certified Kubernetes Security Specialist certification and official training course.

Read more in this section:

- Pod Security Standards

- Pod Security Admission
- Service Accounts
- Pod Security Policies
- Security For Linux Nodes
- Security For Windows Nodes
- Controlling Access to the Kubernetes API
- Role Based Access Control Good Practices
- Good practices for Kubernetes Secrets
- Multi-tenancy
- Hardening Guide - Authentication Mechanisms
- Hardening Guide - Scheduler Configuration
- Kubernetes API Server Bypass Risks
- Linux kernel security constraints for Pods and containers
- Security Checklist
- Application Security Checklist

> Items on this page refer to third party products or projects that provide functionality required by Kubernetes. The Kubernetes project authors aren't responsible for those third-party products or projects. See the CNCF website guidelines for more details.
>
> You should read the content guide before proposing a change that adds an extra third-party link.

Feedback

Was this page helpful?

Yes No

Last modified February 03, 2025 at 5:28 PM PST: Add a new page for mutating webhook good practices. (bf971d28d3)