



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023

Kyverno Playground: Make Policy Testing a Breeze and Enjoy the Process!

Shuting Zhao

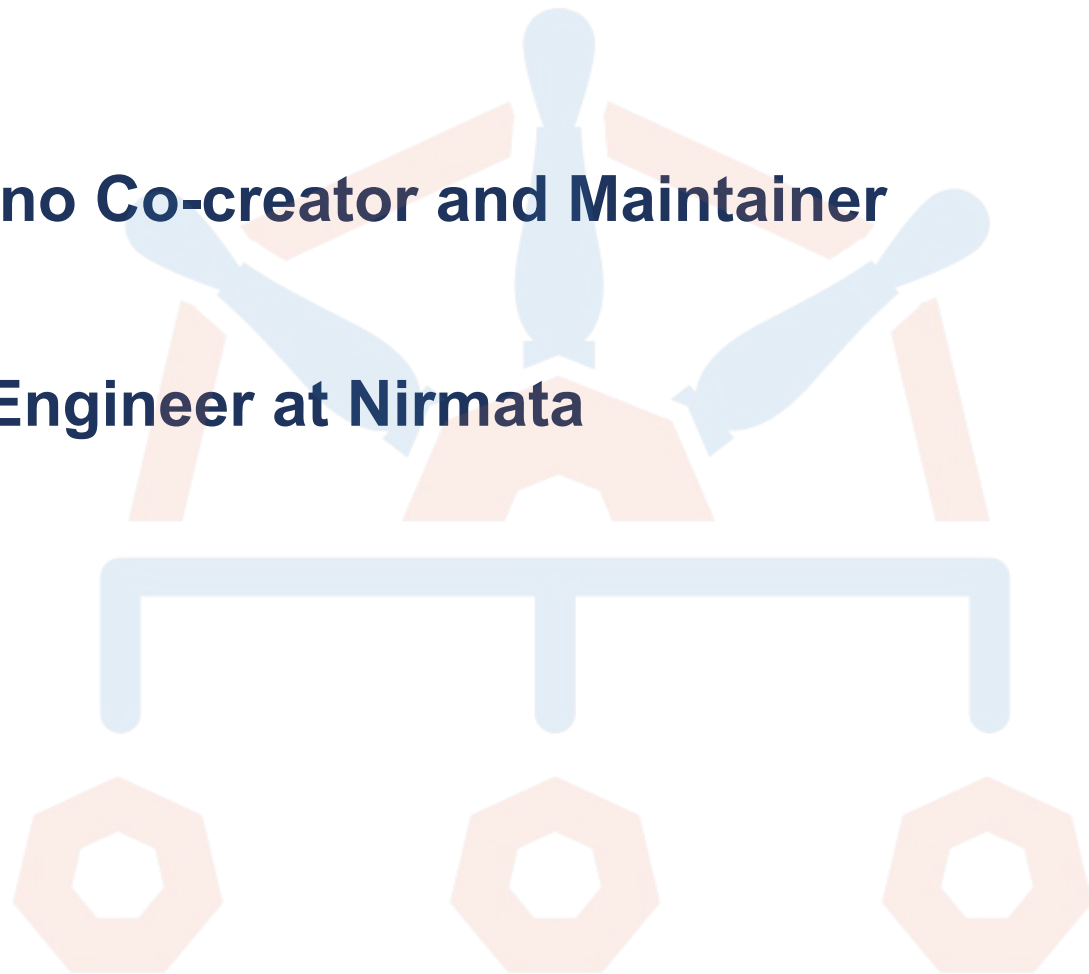
Sept 2023

About me



Shuting Zhao

- **Kyverno Co-creator and Maintainer**
- **Staff Engineer at Nirmata**



Agenda

- Why policies
- What is Kyverno
- Top use cases
- How Kyverno policies work
- Demo
- Summary
- Q & A





KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023

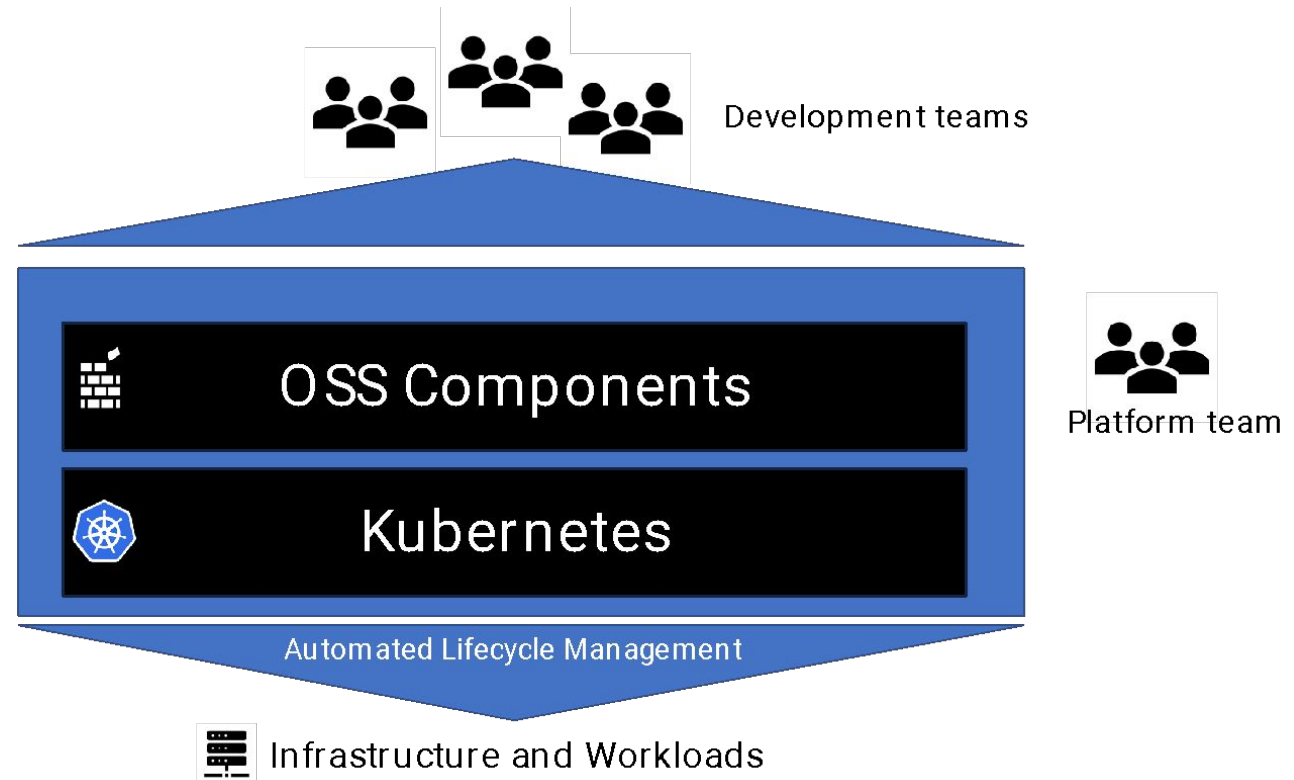
Why Policies

The rise of Kubernetes platforms

By 2026, 80% of software engineering organizations will establish platform teams as internal providers of reusable services, components and tools for application delivery.

Source: Gartner

96% of enterprises are using or evaluating Kubernetes – CNCF survey



The cost of missing Kubernetes guardrails

328

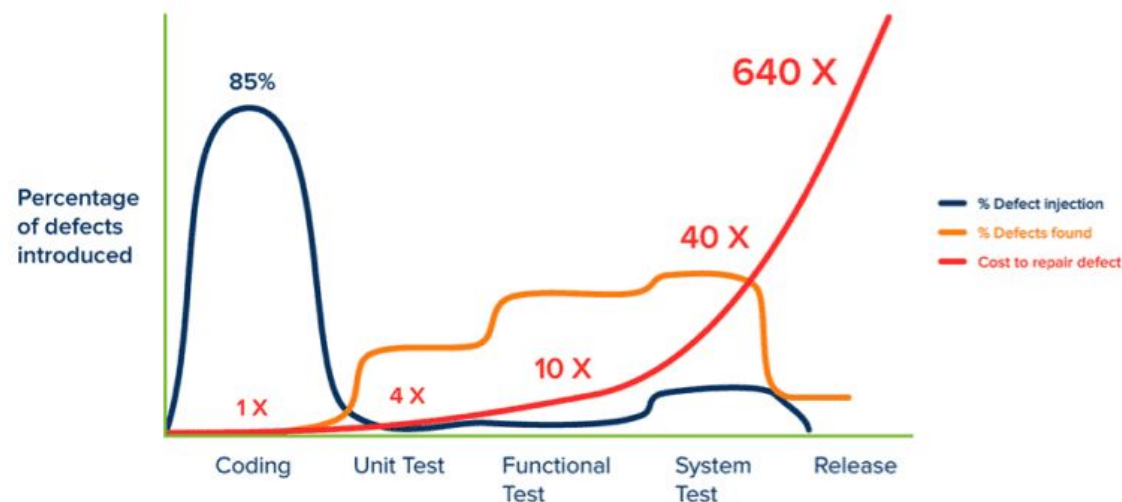
average # of misconfigurations per cluster

110

average # of workloads per cluster

3

average # of findings per workload



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality*.

Phase	Pre-deploy	Production
Cost per defect	\$25	\$15,903
Cost per cluster	\$8151	\$5,216,193

<https://www.cncf.io/blog/2022/02/02/the-cost-of-a-kubernetes-repair-in-development-vs-production/>

Policies are a contract

Developers ☐

Security ☐

Operations ☐

I Agree ☐



KubeCon



CloudNativeCon



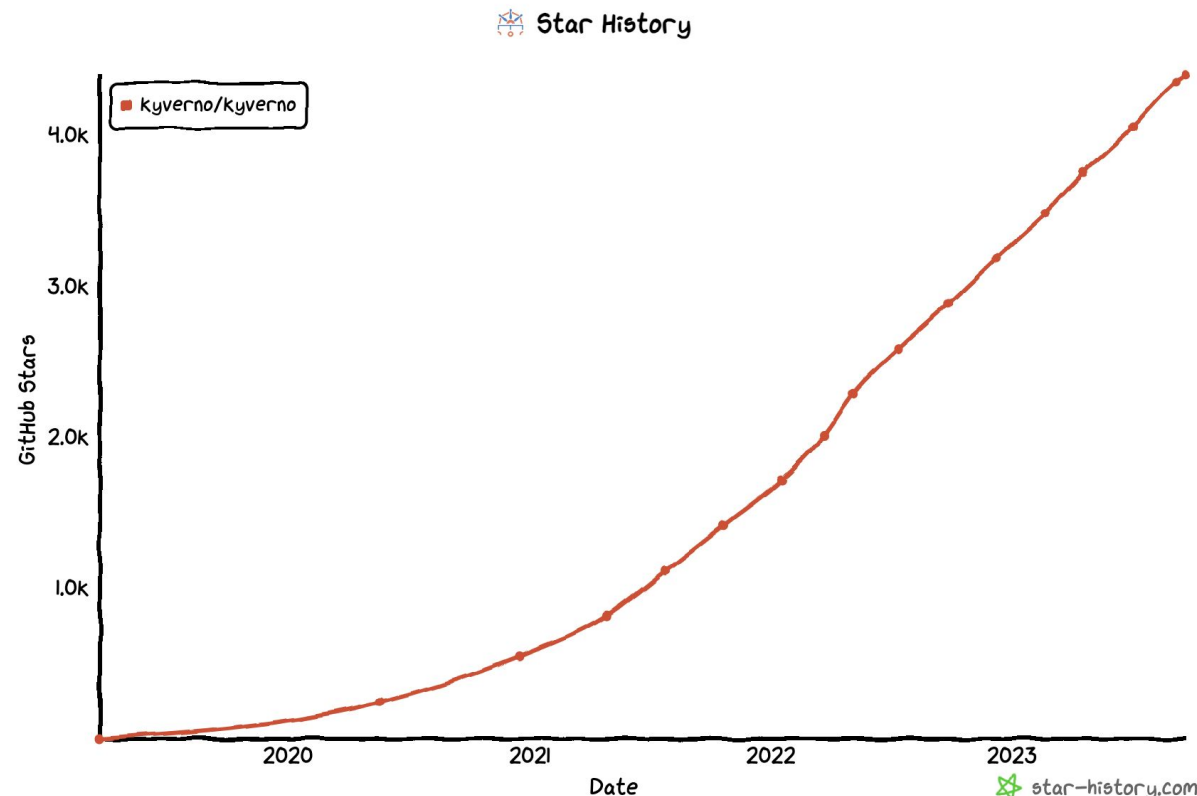
OPEN SOURCE SUMMIT

China 2023

What is Kyverno

What is Kyverno

- Kubernetes native policy engine
- CNCF incubating project
- Fast growing community
 - 2.4 Billion+ image pulls
 - 4.4K+ GitHub Stars
 - 330+ contributors
 - 2300+ Slack members
 - 290+ policies



What is Kyverno

Kyverno simplifies K8s policy management!

A K8s native policy engine:

- Make policies easy to write and manage
- Make policy results easy to process
- Validate (audit or enforce), Mutate, Generate, VerifyImages
- Support all Kubernetes types including Custom Resources
- Use Kubernetes patterns and practices
e.g. labels and selectors, annotations, events, ownerReferences, pod controllers, etc.



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023

Top Use Cases

Kyverno policy management use cases



SecOps

- Pod security
- Workload security
- Granular RBAC
- Workload isolation
- Image signing & verification
- Workload identity

DevOps

- Self-service Kubernetes environments
- Self-service infrastructure (IaC)
- Resource governance and cleanup
- Label/Annotation management
- Naming conventions
- Event driven automation
- Custom CA management
- Time-bound policies

FinOps

- Quota Management
- Pod Requests and limits
- Team and app labels
- Scaling limits
- Scheduled resources
- QoS management
- Auto-scalers



KubeCon



CloudNativeCon

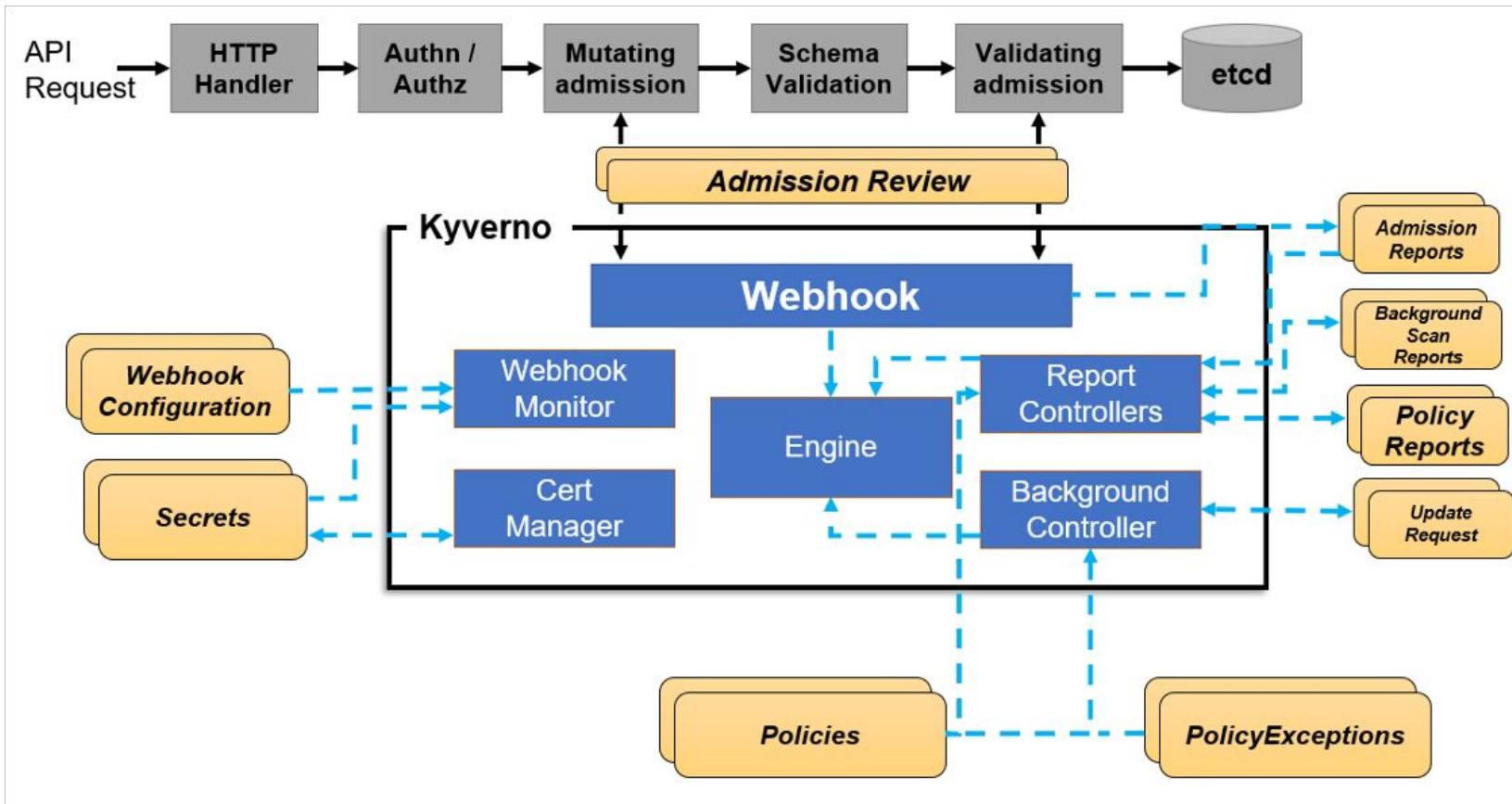


OPEN SOURCE SUMMIT

China 2023

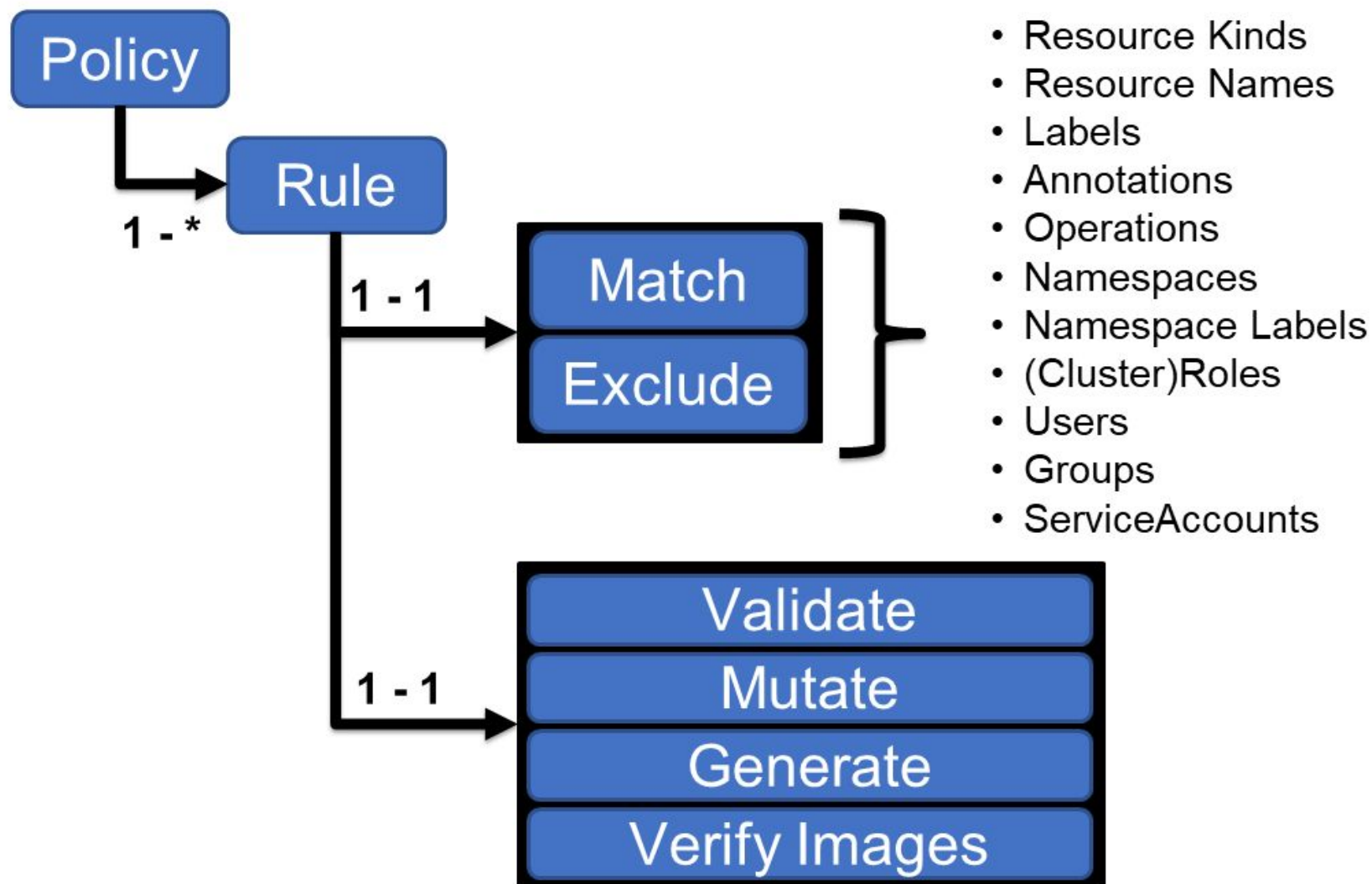
How Kyverno Policies Work

Kyverno architecture




- ✓ Admission Controller
- ✓ Background Scanner
- ✓ CLI for static analysis

A Kyverno policy



A Validate policy

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
spec:
  validationFailureAction: Enforce
  rules:
  - name: check-team
    match:
      any:
      - resources:
          kinds:
          - Pod
    validate:
      message: "label 'team' is required"
      pattern:
        metadata:
          labels:
            team: "?*"
```



```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
```


A Validate policy

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
spec:
  validationFailureAction: Enforce
  rules:
  - name: check-team
    match:
      any:
      - resources:
          kinds:
          - Pod
    validate:
      message: "label 'team' is required"
      pattern:
        metadata:
          labels:
            team: "?*"
```

```
spec:
  validationFailureAction: Enforce
  rules:
  - name: check-team
    match:
      any:
      - resources:
          kinds:
          - Pod
```

A Validate policy

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
spec:
  validationFailureAction: Enforce
  rules:
  - name: check-team
    match:
      any:
      - resources:
          kinds:
            - Pod
```

```
validate:
  message: "label 'team' is required"
  pattern:
    metadata:
      labels:
        team: "?*"
```

```
validate:
  message: "label 'team' is required"
  pattern:
    metadata:
      labels:
        team: "?*"
```



KubeCon



CloudNativeCon









OPEN SOURCE SUMMIT

China 2023

Demo

Kyverno Playground

```
⌂ Policies ⓘ     
1  apiVersion: kyverno.io/v1  
2  kind: ClusterPolicy  
3  metadata:  
4    name: require-labels  
5  spec:  
6    validationFailureAction: Enforce  
7    rules:  
8      - name: check-team  
9        match:  
10         any:  
11           - resources:  
12             kinds:  
13               - Pod  
14         validate:  
15           message: "label 'team' is required"  
16           pattern:  
17             metadata:  
18               labels:  
19                 team: "?*"  
20
```

```
⌂ Resources ⓘ     
1  apiVersion: v1  
2  kind: Pod  
3  metadata:  
4    labels:  
5      run: nginx  
6    name: nginx  
7    namespace: default  
8  spec:  
9    containers:  
10     - image: nginx  
11       name: nginx  
12     resources: {}
```

💥 Try out at <https://playground.kyverno.io/next/#/>

Demo - Validate policies

- Validate Pod labels

Policies ⓘ

```
1 apiVersion: kyverno.io/v1
2 kind: ClusterPolicy
3 metadata:
4   name: require-labels
5 spec:
```

Resources ⓘ

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   labels:
5     run: nginx
```

Results ⓘ

☐ Hide no match results

Validation Results

APIVersion	Kind	Resource	Policy	Rule	Status
v1	Pod	default/nginx	require-labels	check-team	fail ^

validation error: label 'team' is required. rule check-team failed at path /metadata/labels/team/

CLOSE

COPY POLICY TO CLIPBOARD

Demo - Mutate policies

- [Mutate Pod labels](#)

Policies ⓘ

```
1 apiVersion: kyverno.io/v1
2 kind: ClusterPolicy
3 metadata:
4   name: strategic-merge-patch
5 spec:
6   rules:
```

Resources ⓘ

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   labels:
5     run: nginx
6   name: nginx
```

Results ⓘ

☐ Hide no match results

Mutation Results

APIVersion	Kind	Resource	Policy	Rule	Status	Details
v1	Pod	dev/nginx	strategic-merge-patch	set-team-label	pass	🔗

CLOSE

COPY POLICY TO CLIPBOARD

Demo - Generate policies

- [Generate a NetworkPolicy](#)
- [Clone a Secret](#)

The screenshot shows a web interface for generating policies. It features two main panels: 'Policies' and 'Resources'. The 'Policies' panel displays a ClusterPolicy named 'default' with the following YAML:

```
1 apiVersion: kyverno.io/v1
2 kind: ClusterPolicy
3 metadata:
4   name: default
5 spec:
6   rules:
```

The 'Resources' panel displays a Namespace named 'staging' with the following YAML:

```
1 apiVersion: v1
2 kind: Namespace
3 metadata:
4   name: staging
5 spec: {}
```

A 'Results' dialog box is open, showing the following table:

APIVersion	Kind	Resource	Policy	Rule	Status	Details
v1	Namespace	staging	default	deny-all-traffic	pass	Details

The dialog also includes a 'CLOSE' button and a 'COPY POLICY TO CLIPBOARD' button.

Demo - VerifyImages policies

- Verify Image Signatures

Policies ⓘ

Resources ⓘ

Results

☐ Hide no match results

ImageVerification Results

APIVersion	Kind	Resource	Policy	Rule	Status	Details
v1	Pod	default/bad-pod	verify-images	verify-notary-signature	fail	ⓘ
v1	Pod	default/good-pod	verify-images	verify-notary-signature	pass	🔗

Validation Results

APIVersion	Kind	Resource	Policy	Rule	Status	
v1	Pod	default/bad-pod	verify-images	verify-notary-signature	fail	▼
v1	Pod	default/good-pod	verify-images	verify-notary-signature	pass	▼

CLOSE

COPY POLICY TO CLIPBOARD

BxMHU2VhdHRsZTEPMA0GA1UEChMTMjY5Y2hp
bmEyMDIzLmIiBjANBgkqhkiG9w0BAQEFAAQAQ8AMIIBCBKCAQEA
v01MmCbJ5aGItLjG2eKiU9JarvXLGtKY+5SoBtfdX0XvtQQfgid6X26/xYPkaD
sHScZ6mqFmag05cNHhcFacXDnSK+HKIC6ka3WdFW5LtvMAP60bpIMTIu5NL4yZgv
JGQq0m2WzAAYn26dfGLfGaf0UsUHJoayhcT2fLTV+Gldp612Hk9jGeLcd1WU9BL8

?

▶ START

Demo - PolicyExceptions

- Pod Exception

Policies ⓘ

Resources ⓘ

```
4 name: verify-images
5 spec:
```

```
1 apiVersion: v1
2 kind: Pod
```

Results

☐ Hide no match results

ImageVerification Results

APIVersion	Kind	Resource	Policy	Rule	Status	Details
v1	Pod	default/demo-bad-pod	verify-images	verify-notary-signature	skip	!

Validation Results

APIVersion	Kind	Resource	Policy	Rule	Status
v1	Pod	default/demo-bad-pod	verify-images	verify-notary-signature	skip

CLOSE

COPY POLICY TO CLIPBOARD

Demo - Validating Admission Policies

- Disallow hospath volume

The screenshot displays a Kubernetes dashboard with two panels: 'Policies' and 'Resources'. The 'Policies' panel shows a ClusterPolicy named 'disallow-host-path' with a validation rule that enforces the disallow of host-path volumes. The 'Resources' panel shows a Deployment named 'nginx' with a volume named 'udev' that uses the 'hostPath' type. A modal window titled 'Results' is open, showing the validation results for the 'nginx' Deployment. The modal includes a table with the following data:

APIVersion	Kind	Resource	Policy	Rule	Status
apps/v1	Deployment	nginx	disallow-host-path	host-path	fail

The modal also includes a 'CLOSE' button and a 'COPY POLICY TO CLIPBOARD' button. The background shows the YAML configuration for the 'nginx' Deployment, which includes a volume named 'udev' with a 'hostPath' type.

Join the Kyverno community

- The Kyverno docs & samples: <https://kyverno.io>
- Slack Channel: <https://slack.k8s.io/#kyverno>
- Weekly meetings: <https://groups.google.com/g/kyverno>

Bug report

Create a report to help us improve

[Get started](#)

Feature request

Suggest an idea for this project

[Get started](#)

Policy to support

Suggest a policy that you would like Kyverno to support

[Get started](#)

Q & A



@ShutingZhao2



@Shuting Zhao



@realshuting



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023