Kubernetes Native Policy Management

Kyverno.io

Pedro Ibáñez Requena
Ecosystem Field Engineering Telco 5g Team

Index

- What is Kyverno?
- How it works?
- Features
- Setup
- Demo
- References

What is Kyverno?

What is Kyverno?

Kyverno is a **policy engine designed for Kubernetes**. With Kyverno, **policies are managed as Kubernetes resources** and no new language is required to write policies.

This allows using familiar tools such as kubectl, git, and kustomize to manage policies.

Kyverno policies can validate, mutate, and generate Kubernetes resources plus ensure OCI image supply chain security. The Kyverno CLI can be used to test policies and validate resources as part of a CI/CD pipeline.

How it works?

How it works?

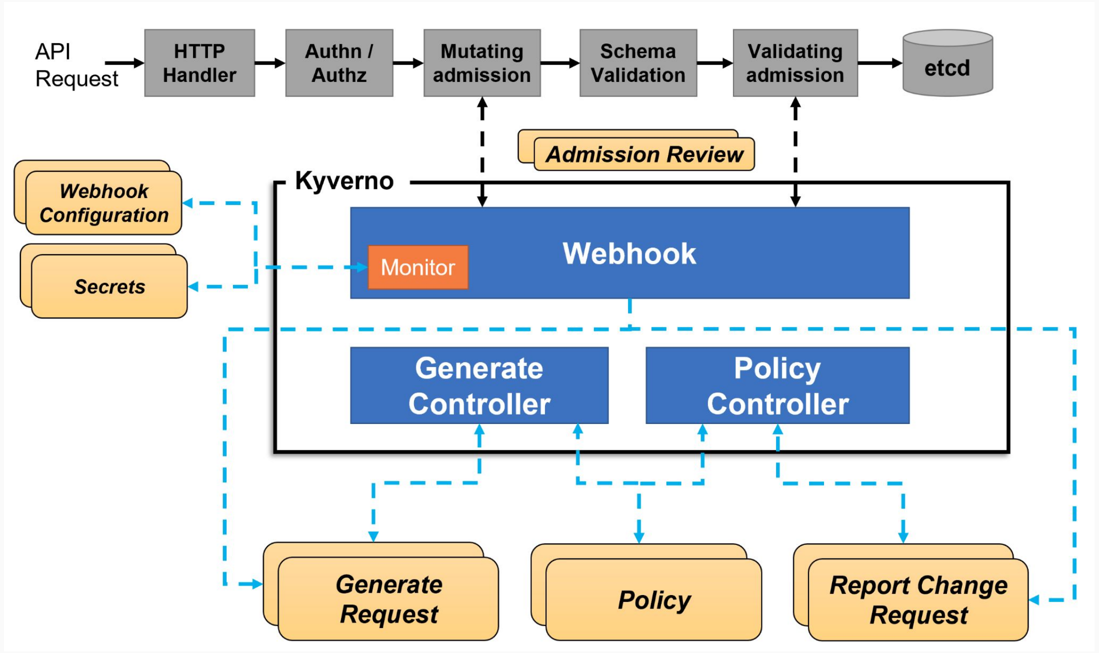
Kyverno runs as a **dynamic admission controller** in a Kubernetes cluster. Kyverno **receives validating and mutating admission webhook HTTP callbacks** from the **kube-apiserver** and **applies matching policies to return results that enforce admission policies or reject requests**.

Kyverno **policies can match resources using the resource kind, name, and label selectors**. Wildcards are supported in names.

Mutating policies can be written as **overlays** (similar to Kustomize) or as a RFC 6902 JSON Patch. Validating policies also use an **overlay** style syntax, with support for **pattern matching and conditional (if-then-else) processing**.

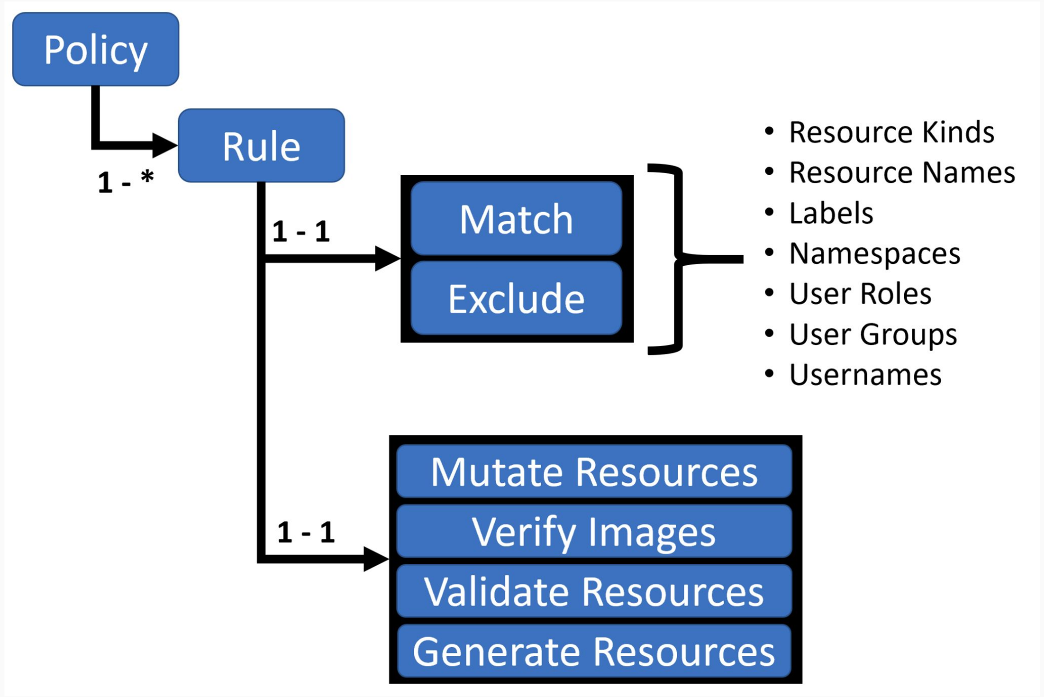
How it works?

High-level architecture



How it works?

Policies and rules



Features

Features

Kyverno (Greek for “govern”) is a policy engine designed specifically for Kubernetes. Some of its many features include:

- policies as Kubernetes resources (no new language to learn!)
- **validate, mutate, or generate any resource**
- **verify container images** for software supply chain security
- inspect image metadata
- match resources using label selectors and wildcards
- validate and mutate using overlays (like Kustomize!)
- synchronize configurations across Namespaces
- **block** non-conformant resources using admission controls, or **report** policy violations
- **test** policies and **validate** resources using the Kyverno CLI, in your CI/CD pipeline, before applying to the cluster
- manage policies as code using familiar tools like git and kustomize

Features

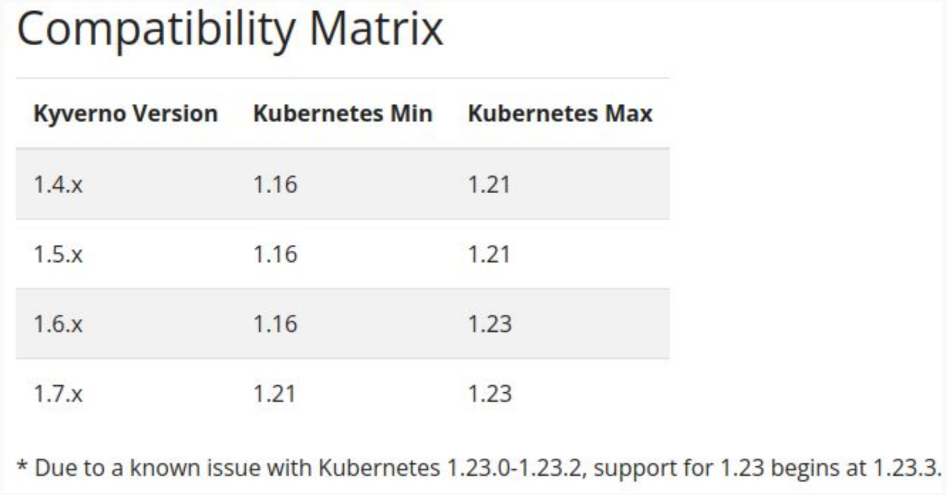
Kyverno allows cluster administrators to **manage environment specific configurations independently of workload configurations** and **enforce configuration best practices** for their clusters.

Kyverno can be used to **scan existing workloads for best practices**, or can be used to **enforce best practices by blocking or mutating API requests**.

Setup

Setup

🚫 Kyverno requires k8s above v1.14 which adds webhook timeouts



Compatibility Matrix

Kyverno Version	Kubernetes Min	Kubernetes Max
1.4.x	1.16	1.21
1.5.x	1.16	1.21
1.6.x	1.16	1.23
1.7.x	1.21	1.23

* Due to a known issue with Kubernetes 1.23.0-1.23.2, support for 1.23 begins at 1.23.3.

Setup

YAML declarations - installation of kyverno

```
$ kubectl create -f https://raw.githubusercontent.com/kyverno/kyverno/main/config/install.yaml

$ kubectl create -f
https://raw.githubusercontent.com/kyverno/kyverno/release-1.7/config/release/install.yaml

$ kubectl delete -f https://raw.githubusercontent.com/kyverno/kyverno/main/config/install.yaml
```

Setup

Install Kyverno using a Helm chart

```
# Add the Helm repository

$ helm repo add kyverno https://kyverno.github.io/kyverno/

# Scan your Helm repositories to fetch the latest available charts.

$ helm repo update

# Install the Kyverno Helm chart into a new namespace called "kyverno"

$ helm install kyverno kyverno/kyverno -n kyverno --create-namespace

$ helm uninstall kyverno kyverno/kyverno --namespace kyverno
```

Setup

Install the CLI using Krew or building the CLI from source:

```
$ git clone https://github.com/kyverno/kyverno

$ cd kyverno

$ make cli

$ cp ./cmd/cli/kubectl-kyverno/kyverno /usr/local/bin/kyverno
```

Setup

Regardless which uninstallation method is chosen, webhooks will need to be manually removed as the final step. Use the below commands to delete those webhook configurations.

```
$ kubectl delete mutatingwebhookconfigurations kyverno-policy-mutating-webhook-cfg
kyverno-resource-mutating-webhook-cfg kyverno-verify-mutating-webhook-cfg
```

```
$ kubectl delete validatingwebhookconfigurations kyverno-policy-validating-webhook-cfg
kyverno-resource-validating-webhook-cfg
```

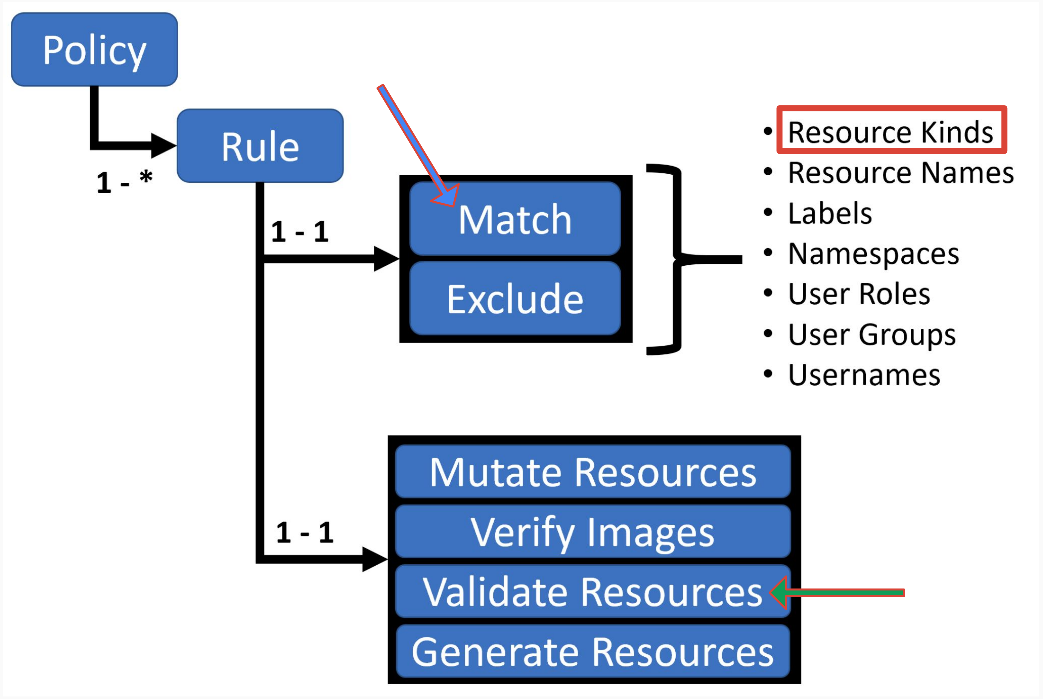
Setup - Basic example

Contains a single validation rule that requires that all Pods have a app.kubernetes.io/name label.

The policy attribute validationFailureAction is set to enforce to block API requests that are non-compliant (using the default value audit will report violations but not block requests.)

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

Setup - Basic example



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

Demo

References

References

Official web: https://kyverno.io

Source code: https://github.com/kyverno/kyverno/

Documentation: https://kyverno.io/docs/

KubeCon + CloudNativeCon North America 2021: Kyverno Office Hour:
https://www.youtube.com/watch?v=v0yh8b6lPXQ

Demo files: https://github.com/ptrnull/kyverno-demo

Thanks!