



Automate Kubernetes security
with policy as code

Agenda

Au programme

- Qui suis-je ?
- La gouvernance, pour quoi faire ?
- Kyverno en bref
- Un peu de pratique
- De la validation mais pas que...
- L'écosystème kyverno
- Mot de la fin

whoami

Charles-Edouard Brétéché

Kyverno contributor since Feb 2022 (kyverno 1.7)

Kyverno maintainer since Apr 2022

Staff Engineer @ Nirmata (creator of kyverno)

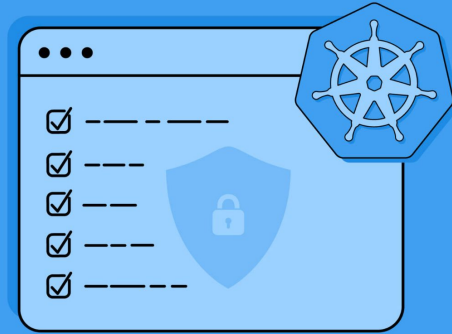
Working full time on kyverno and OSS projects

- <https://www.linkedin.com/in/eddycharly>
- <https://github.com/eddycharly>
- <https://sessionize.com/charles-edouard-breteche>



La gouvernance - pour quoi faire ?

La sécurité, une préoccupation majeure



Détecter les vulnérabilités

Prévenir les fuites de données

Sécuriser la chaîne de build

Le contrôle des coûts



Gestion de quotas de ressources

Contrôle des load balancers

Gestion des ressources cloud

L'automatisation

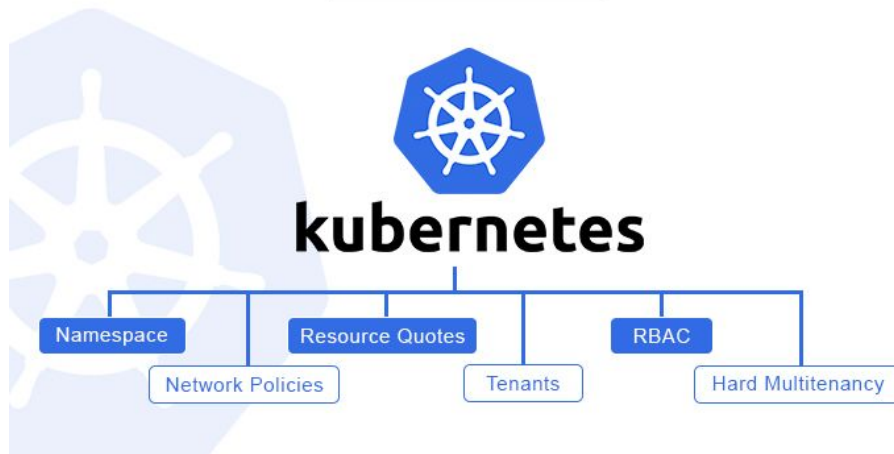


Stratégies de nettoyage

Template de création automatisée

Injection d'éléments

Multi tenancy



Namespaces as a service

Network policies

Roles et RoleBindings

Un élément de platform engineering



Réduire les risques liés au déploiement continu

Permettre la collaboration des équipes

Rendre visibles et accessibles les politiques aux yeux de tous

Kyverno en bref

Kyverno en quelque mots

Open-source

Kubernetes native

Simple d'utilisation

Déclaratif



GitOps friendly

Les types de polices

Validation

Image
verification



Generation

Mutation

Cleanup

Une policy ça ressemble a quoi ?

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  // cluster policies have no namespace
  name: require-labels
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: require-team
      match:
        any:
          - resources:
              kinds:
                - Pod
      validate:
        message: 'The label `team` is required.'
        pattern:
          metadata:
            labels:
              team: '?*'
```

```
apiVersion: kyverno.io/v1
kind: Policy
metadata:
  namespace: foo
  name: require-labels
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: require-team
      match:
        any:
          - resources:
              kinds:
                - Pod
      validate:
        message: 'The label `team` is required.'
        pattern:
          metadata:
            labels:
              team: '?*'
```

Un catalogue a disposition

Plus de 300 politiques au catalogue <https://kyverno.io/policies>



Policies

Pod Security

Gatekeeper Migration

Policy Type

- ☐ Generate
- ☐ Mutate
- ☐ Validate
- ☐ VerifyImages
- ☐ Cleanup

Policy Category

- ☐ AWS
- ☐ Argo
- ☐ Best Practices
- ☐ CAST AI
- ☐ Cert-Manager
- ☐ Consul
- ☐ EKS Best Practices
- ☐ ExternalSecretOperator
- ☐ Flux
- ☐ Istio
- ☐ Karpenter

Most validate policies here are set to **Audit** mode by default. To block resources immediately, set to **Enforce**.

For information on the Kyverno annotations listed in these policies, see [this page](#).

313 Policies Found

Add AppArmor Annotations

In the earlier Pod Security Policy controller, it was possible to define a setting which would enable AppArmor for all the containers within a Pod so they may be assigned the desired profile. Assigning an AppArmor profile, accomplished via an annotation, is useful in that it allows secure defaults to be defined and may also result in passing other validation rules such as those in the Pod Security Standards. This policy mutates Pods to add an annotation for every container to enable AppArmor at the runtime/default level.

Add Capabilities

In the earlier Pod Security Policy controller, it was possible to configure a policy to add capabilities to containers within a Pod. This made it easier to assign some basic defaults rather than blocking Pods or to simply provide capabilities for certain workloads if not specified. This policy mutates Pods to add the capabilities SETFCAP and SETUID so long as they are not listed as dropped capabilities first.

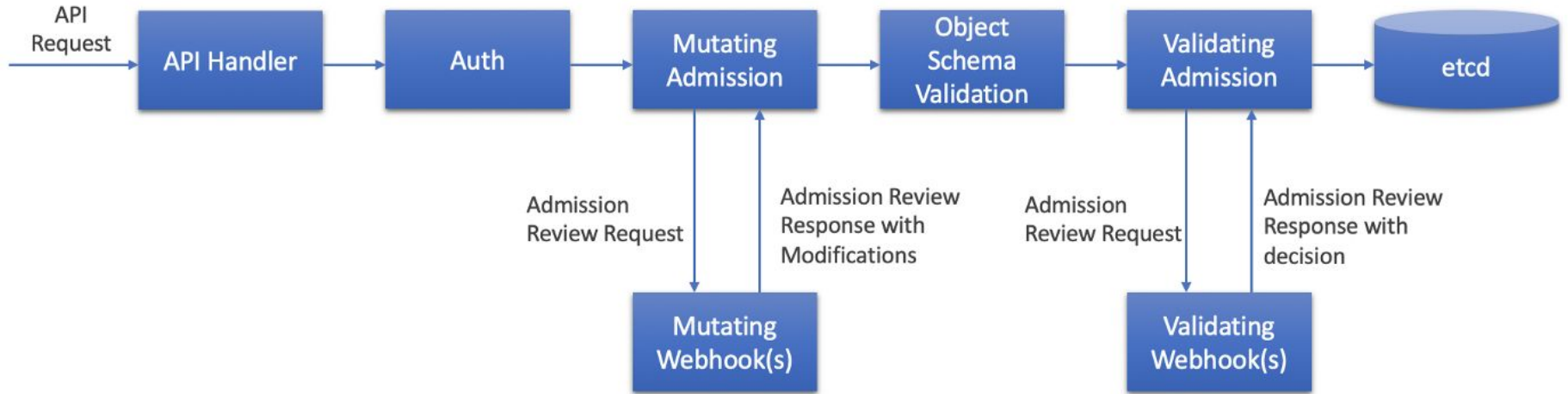
Add CAST AI Removal Disabled

CAST AI will not downscale a node that includes a pod with the autoscaling.cast.ai/removal-disabled="true" label on it, this protects sensitive workloads from being evicted and can be attributed to any pod to protect against unwanted downscaling. This policy will mutate jobs and cronjobs to add the removal-disabled label to protect against eviction.

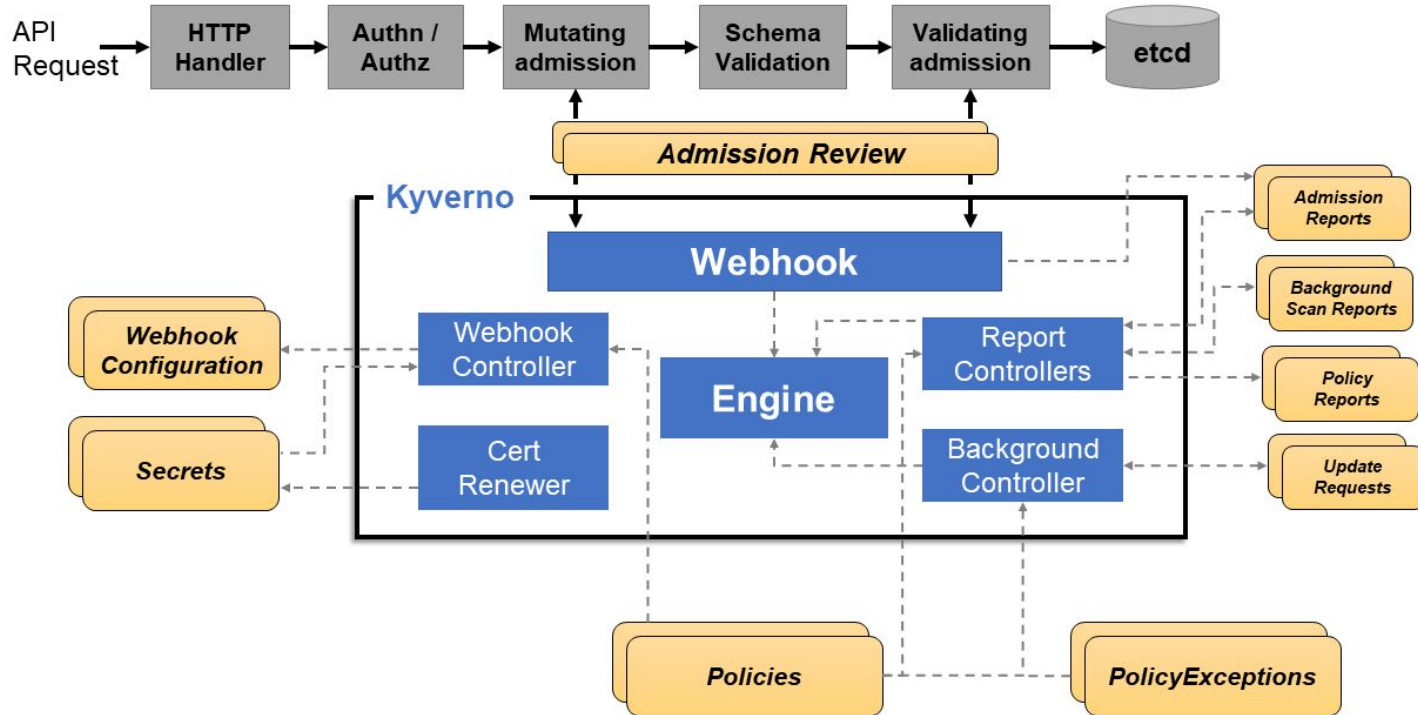
Add Certificates as a Volume

In some cases you would need to trust custom CA certificates for all the containers of a Pod. It makes sense to be in a

Cycle de vie d'une requête Kubernetes



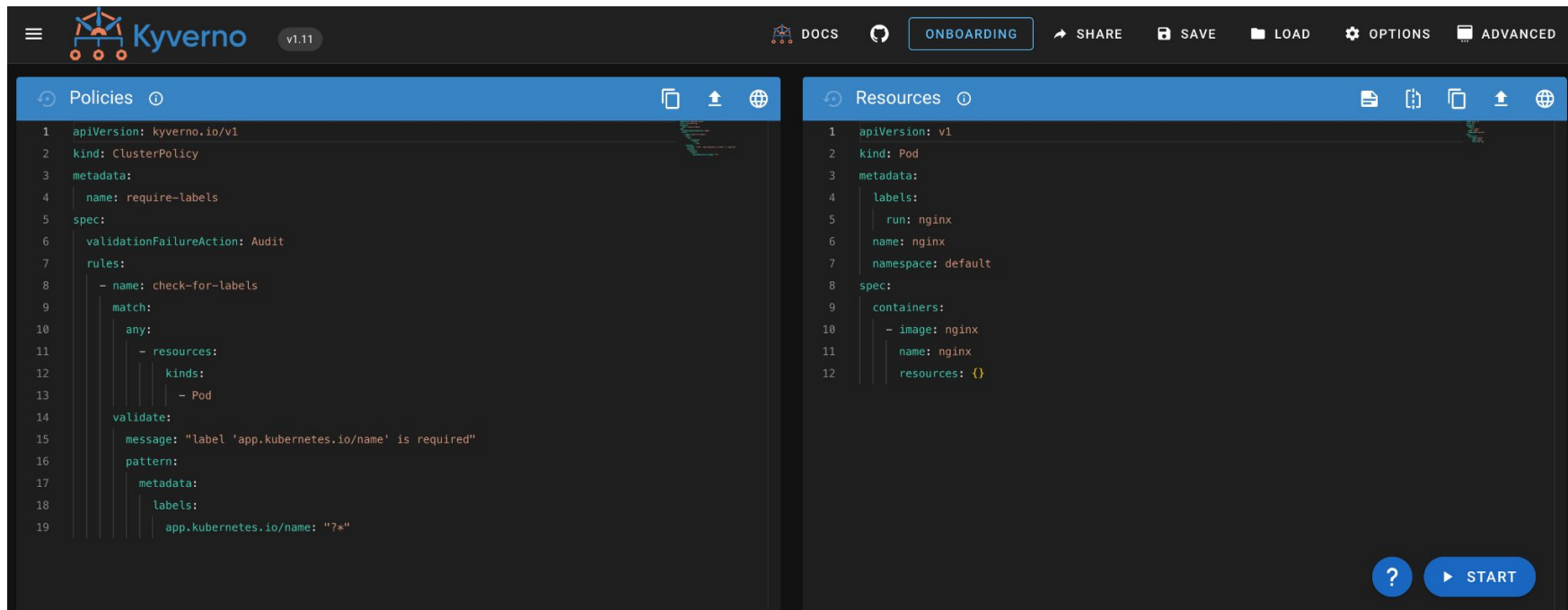
Vision globale



Un peu de pratique

De quoi a-t-on besoin ? (OPTION 1)

De rien, juste un navigateur <https://playground.kyverno.io>



The screenshot displays the Kyverno playground interface, which is a web-based tool for testing and applying Kyverno policies. The interface is divided into two main panels: "Policies" on the left and "Resources" on the right. The top navigation bar includes the Kyverno logo, version "v1.11", and several utility buttons: "DOCS", "ONBOARDING", "SHARE", "SAVE", "LOAD", "OPTIONS", and "ADVANCED".

The "Policies" panel shows a ClusterPolicy named "require-labels". The policy is defined with the following YAML structure:

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

```

The "Resources" panel shows a Pod resource named "nginx" in the "default" namespace. The resource is defined with the following YAML structure:

```
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: {}

```

At the bottom right of the interface, there is a help icon (question mark) and a "START" button.

De quoi a-t-on besoin ? (OPTION 2)

Un cluster Kubernetes, Helm pour installer kyverno et kubectl

```
# create a cluster
kind create cluster --image kindest/node:v1.28.0 --wait 1m

# install kyverno 1.11
helm upgrade --install kyverno --version 3.1.3 \
  --namespace kyverno --create-namespace \
  --wait --repo https://kyverno.github.io/kyverno kyverno
```

Hello world

[Playground link](#)

```
kubectl apply -n kyverno -f - <<EOF
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
spec:
  validationFailureAction: Enforce
  background: true
  rules:
  - name: require-team
    match:
      any:
      - resources:
          kinds:
            - Pod
    validate:
      message: 'The label `team` is required.'
      pattern:
        metadata:
          labels:
            team: '?*'
EOF
```

Image verification

[Playground link](#)

```
kubectl apply -n kyverno -f - <<EOF
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: check-image
spec:
  validationFailureAction: Enforce
  background: true
  rules:
    - name: check-image
      match:
        any:
          - resources:
              kinds:
                - Pod
      verifyImages:
        - imageReferences:
            - "ghcr.io/kyverno/test-verify-image*"
          attestors:
            - count: 1
              entries:
                - keys:
                    publicKey: |-
                      -----BEGIN PUBLIC KEY-----
                      MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrfM
                      5/KAQN0/KjHcorm/J5yctVd7iEcnessRQjU9l7hmKO6JWVGHPDguIyakZA==
                      -----END PUBLIC KEY-----
              rekor:
                ignoreTlog: true
                url: https://rekor.sigstore.dev
EOF
```

PSS

[Playground link](#)

```
kubectl apply -f - <<EOF
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: podsecurity-subrule-restricted
spec:
  background: true
  validationFailureAction: Enforce
  rules:
  - name: restricted
    match:
      any:
      - resources:
          kinds:
            - Pod
    validate:
      podSecurity:
        level: restricted
        version: latest
EOF
```

Exceptions

[Playground link](#)

```
apiVersion: kyverno.io/v2alpha1
kind: PolicyException
metadata:
  name: delta-exception
  namespace: delta
spec:
  exceptions:
  - policyName: disallow-host-namespaces
    ruleNames:
    - host-namespaces
    - autogen-host-namespaces
  match:
    any:
    - resources:
        kinds:
        - Pod
        - Deployment
        namespaces:
        - delta
        names:
        - important-tool*
```


De la validation mais pas que...

Generation

[Playground link](#)

```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-ns-quota
spec:
  rules:
    - name: generate-resourcequota
      match:
        any:
          - resources:
              kinds:
                - Namespace
      generate:
        apiVersion: v1
        kind: ResourceQuota
        name: default-resourcequota
        synchronize: true
        namespace: "{{request.object.metadata.name}}"
        data:
          spec:
            hard:
              requests.cpu: '4'
              requests.memory: '16Gi'
              limits.cpu: '4'
              limits.memory: '16Gi'
```

Cleanup

```
apiVersion: kyverno.io/v2beta1
kind: ClusterCleanupPolicy
metadata:
  name: clean-bare-pods
spec:
  match:
    any:
      - resources:
          kinds:
            - Pod
  conditions:
    all:
      - key: "{ target.metadata.ownerReferences[] || `[]` }"
        operator: Equals
        value: []
  schedule: "* * * * *
```

Cleanup

```
apiVersion: v1
kind: Namespace
metadata:
  name: foo
  labels:
    cleanup.kyverno.io/ttl: 2m
```

L' écosystème kyverno

Policy reporter

Apporte des fonctionnalités basées sur les rapports générés par kyverno:

Calcul de metrics, interface utilisateur, sauvegarde périodique, ...

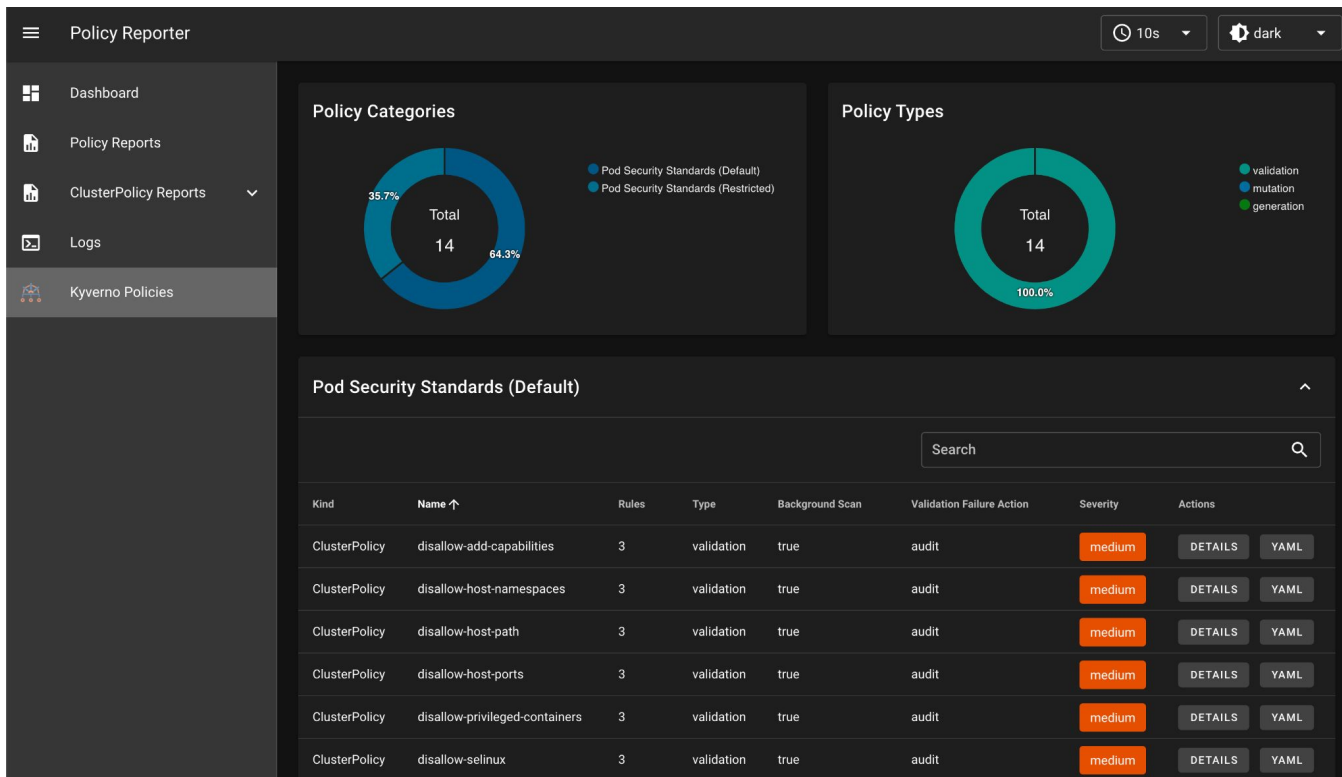
```
# install policy-reporter

helm upgrade --install policy-reporter \
  --namespace policy-reporter --create-namespace \
  --wait --repo https://kyverno.github.io/policy-reporter policy-reporter \
  --values - <<EOF
ui:
  enabled: true

kyvernoPlugin:
  enabled: true
EOF
```

Policy reporter

<https://kyverno.github.io/policy-reporter>



Kyverno CLI

- Validation offline (ou connectée à un cluster)
- Particulièrement utile dans les pipelines de CI / CD
- Permet d'évaluer une ressource par rapport à un ensemble de règles localement

Kyverno Playground

<https://playground.kyverno.io>

The screenshot displays the Kyverno Playground interface. At the top, there is a navigation bar with the Kyverno logo, version v1.11, and links for Docs, Onboarding, Share, Save, Load, Options, and Advanced. Below the navigation bar, the interface is split into two main panels: Policies and Resources.

Policies Panel: The left panel shows a ClusterPolicy named 'require-labels'. The YAML content is as follows:

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

```

Resources Panel: The right panel shows a Pod resource named 'nginx'. The YAML content is as follows:

```
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: {}

```

At the bottom right of the interface, there is a question mark icon and a 'START' button.

Kyverno JSON

- <https://github.com/kyverno/kyverno-json>
- <https://kyverno.github.io/kyverno-json/latest>
- Étend kyverno au delà des frontières de Kubernetes
- Permet d'appliquer des politiques à n'importe quel payload JSON
 - Dockerfile
 - Terraform
 - Cloudformation
 - ECS
 - Lambda
- Même principe que kyverno, ne requiert aucune écriture de code

Kyverno Chainsaw

- <https://github.com/kyverno/chainsaw>
- <https://kyverno.github.io/chainsaw/latest/>
- Le dernier né des projets open source kyverno
- Basé sur Kyverno-JSON
- Permet les tests de bout en bout des opérateurs Kubernetes
- Fonctionne de manière entièrement déclarative
- Même principe que kyverno et Kyverno-JSON, ne requiert aucune écriture de code

Mot de la fin

Mot de la fin

- Join the Kyverno slack channel
<https://slack.k8s.io/#kyverno>
- Join the weekly contributors meeting
<https://kyverno.io/community/>
- Star the project to follow
<https://github.com/kyverno/kyverno/>
- Sign up as an adopter
[Register](#)
- Contributions welcome
<https://github.com/kyverno/kyverno/issues?q=is%3Aopen+is%3Aissue+label%3A%22good+first+issue%22>



Thank you!