Policy As Code

2023.03

v1.0

CLOUD NATIVE
COMPUTING FOUNDATION

Pod Security Policy

CLOUD NATIVE
COMPUTING FOUNDATION

Présentation

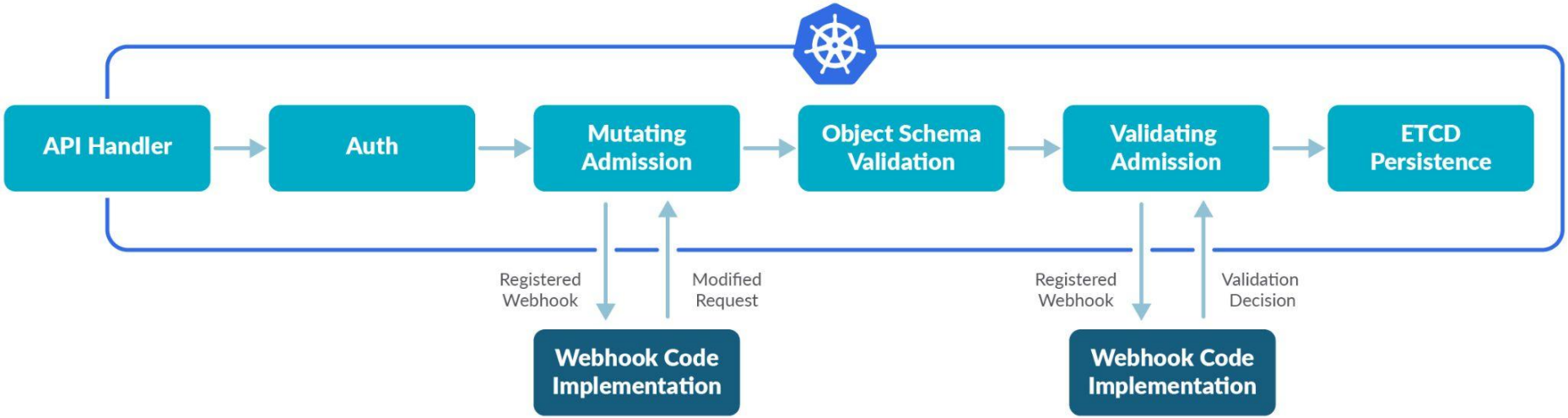
- Une ressource au niveau du cluster qui contrôle les aspects sensibles de la sécurité de la spécification des pods.
- Définit un ensemble de conditions qu'un pod doit respecter pour être accepté dans le système
- Status :
  - Beta depuis Kubernetes v1.3
  - Deprecated en v1.21
  - Supprimée en v1.25

Futur

- Pod Security Admission : Inclus dans Kubernetes
  - Beta en v1.23, par défaut en v1.24
  - Label (namespace): `pod-security.kubernetes.io/<MODE>: <LEVEL>`

- Contrôleurs externes

Admission Controller



Level : Pod Security Standards

- **Privileged** : profil permissif, permettant notamment de monter des répertoires de la machine hôte, d'exécuter le conteneur en mode privilégié, … Ce profil est à réserver aux Pods systèmes ou d'infrastructure de votre cluster.
- **Baseline** : profil de base, restreint au minimum, empêchant l'augmentation de privilèges et adapté pour les valeurs de configurations de Pods par défaut.
- **Restricted** : profil fortement restreint, adapté lorsque vous souhaitez mettre en place un durcissement pour l'exécution de Pods

Mode d'action

- **Warn** : le client est alerté par un warning lors de la création de la ressource
- **Audit** : une annotation est ajoutée sur l'événement d'audit de création du Pod
- **Enforce** : la création du Pod échoue avec un message d'erreur

Policy Engine

Echosystem

- Gatekeeper / Open Policy Agent
- Kyverno
- Kubewarden
- JsPolicy
- K-Rail

https://appvia.github.io/psp-migration/

PSP field	Pod Security Policy	Pod Security Standard (baseline)	Gatekeeper	Kyverno	Kubewarden
privileged	✓	✓	✓	✓	✓
hostPID	✓	✓	✓	✓	✓
hostIPC	✓	✓	✓	✓	✓
hostNetwork	✓	✓	✓	✓	✓
hostPorts	✓	✗	✓	✓	✓
volumes	✓	✓	✓	✓	✓
allowedHostPaths	✓	✗	✓	✓	✓
allowedFlexVolumes	✓	✗	✓	✓	✓
readOnlyRootFilesystem	✓	✗	✓	✓	✓
runAsUser	✓	✗	✓	✓	✓
runAsGroup	✓	✗	✓	✓	✓
supplementalGroups	✓	✗	✓	✓	✓
fsgroup	✓	✗	✓	✓	✓
allowPrivilegeEscalation	✓	✗	✓	✓	✓
defaultAllowPrivilegeEscalation	✓	✗	✓	✓	✓
allowedCapabilities	✓	✗	✓	✓	✓
defaultAddCapabilities	✓	✗	✓	✓	✓
requiredDropCapabilities	✓	✗	✓	✓	✓
seLinux	✓	✗	✓	✓	✓
allowedProcMountTypes	✓	✗	✓	✓	✓
apparmor	✓	✓	✓	✓	✓
seccomp	✓	✓	✓	✓	✓
forbiddenSysctls	✓	✗	✓	✓	✓
allowedUnsafeSysctls	✓	✗	✓	✓	✓

Open Policy Agent / Gatekeeper

Description

- Open Source
- Provient de Styra. Puis CNCF (Graduated)
- Gatekeeper est l'implémentation pour Kubernetes de Open Policy Agent
- Utilise Rego (!!!)
- Addons :
  - Conftest
  - Policy Kubernetes
  - Policy Terraform
  - Google Cloud Policy Controller

Validation des labels

```
package k8s_labels

import data.lib.core # as konstraint_core

policyID := "PORTEFAIX-M0001"

warn[msg] {
    not recommended_labels_provided(core.resource.metadata)
    msg = core.format_with_id(sprintf("%s/%s: should contain all recommanded Kubernetes labels", [core.kind, core.name]), policyID)
}

recommended_labels_provided(metadata) {
    metadata.labels["app.kubernetes.io/name"]
    metadata.labels["app.kubernetes.io/instance"]
    metadata.labels["app.kubernetes.io/version"]
    metadata.labels["app.kubernetes.io/component"]
    metadata.labels["app.kubernetes.io/part-of"]
    metadata.labels["app.kubernetes.io/managed-by"]
}
```

Best Practices

```
package container_liveness_probe

import data.lib.core
import data.lib.pods

policyID := "PORTEFAIX-C0002"

violation[msg] {
    pods.containers[container]
    not container_liveness_probe_provided(container)
    msg := core.format_with_id(sprintf("%s/%s/%s: Container liveness probe be specified", [core.kind, core.name, container.name]), policyID)
}

container_liveness_probe_provided(container) {
    core.has_field(container, "livenessProbe")
}
```

```
package container_resource_constraints

import data.lib.core
import data.lib.pods

policyID := "PORTEFAIX-C0008"

violation[msg] {
    pods.containers[container]
    not container_resources_provided(container)
    msg := core.format_with_id(sprintf("%s/%s/%s: Container resource constraints must be specified", [core.kind, core.name, container.name]), policyID)
}

container_resources_provided(container) {
    container.resources.requests.cpu
    container.resources.requests.memory
    container.resources.limits.memory
}
```


Kyverno

CLOUD NATIVE
COMPUTING FOUNDATION

Description

- Open Source
- Provient de Nirmata. Puis CNCF (Incubating)
- YAML pour définir les policies
- Addons :
  - Bibliothèque : https://kyverno.io/policies/
  - Policy Reporter (Elasticsearch, Loki, Policy Reporter UI, dashboards Grafana, …)
  - Support de PolicyReport CRD du Kubernetes Policy Working Group

Policy Type

- Generate
  - add network policy, add PDB, add Quota, …
- Mutate
  - add labels, nodeSelector, nodeAffinity, …
- Validate
  - PodSecurityStandards, …
  - Conformité de projet
- VerifyImages
  - Cosign Sigstore
  - KMS

Validation des labels

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: portefaix-M0001
  annotations:
    policies.kyverno.io/title: Metadata must set recommanded Kubernetes labels
    policies.kyverno.io/category: Portefaix
    policies.kyverno.io/severity: low
    policies.kyverno.io/subject: Metadata
    policies.kyverno.io/description: >-
      Metadata must set recommanded Kubernetes labels
      See: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels
spec:
  validationFailureAction: audit
  rules:
  - name: check-for-recommanded-kubernetes-labels
    match:
      resources:
        kinds:
        - "ConfigMap"
        - "Secret"
        - "Pod"
        - "Deployment"
        - "DaemonSet"
        - "StatefulSet"
        - "Service"
        - "Job"
        - "CronJob"
        - "Role"
        - "RoleBinding"
        - "ClusterRole"
        - "ClusterRoleBinding"
        - "ServiceAccount"
        - "Ingress"
        - "IngressClass"
        - "NetworkPolicy"
        - "PodDisruptionBudget"
        - "StorageClass"
    validate:
      message: "Kubernetes recommanded labels is required."
      pattern:
        metadata:
          labels:
            app.kubernetes.io/name: "?*"
            app.kubernetes.io/instance: "?*"
            app.kubernetes.io/version: "?*"
            app.kubernetes.io/component: "?*"
            app.kubernetes.io/part-of: "?*"
            app.kubernetes.io/managed-by: "?*"
```

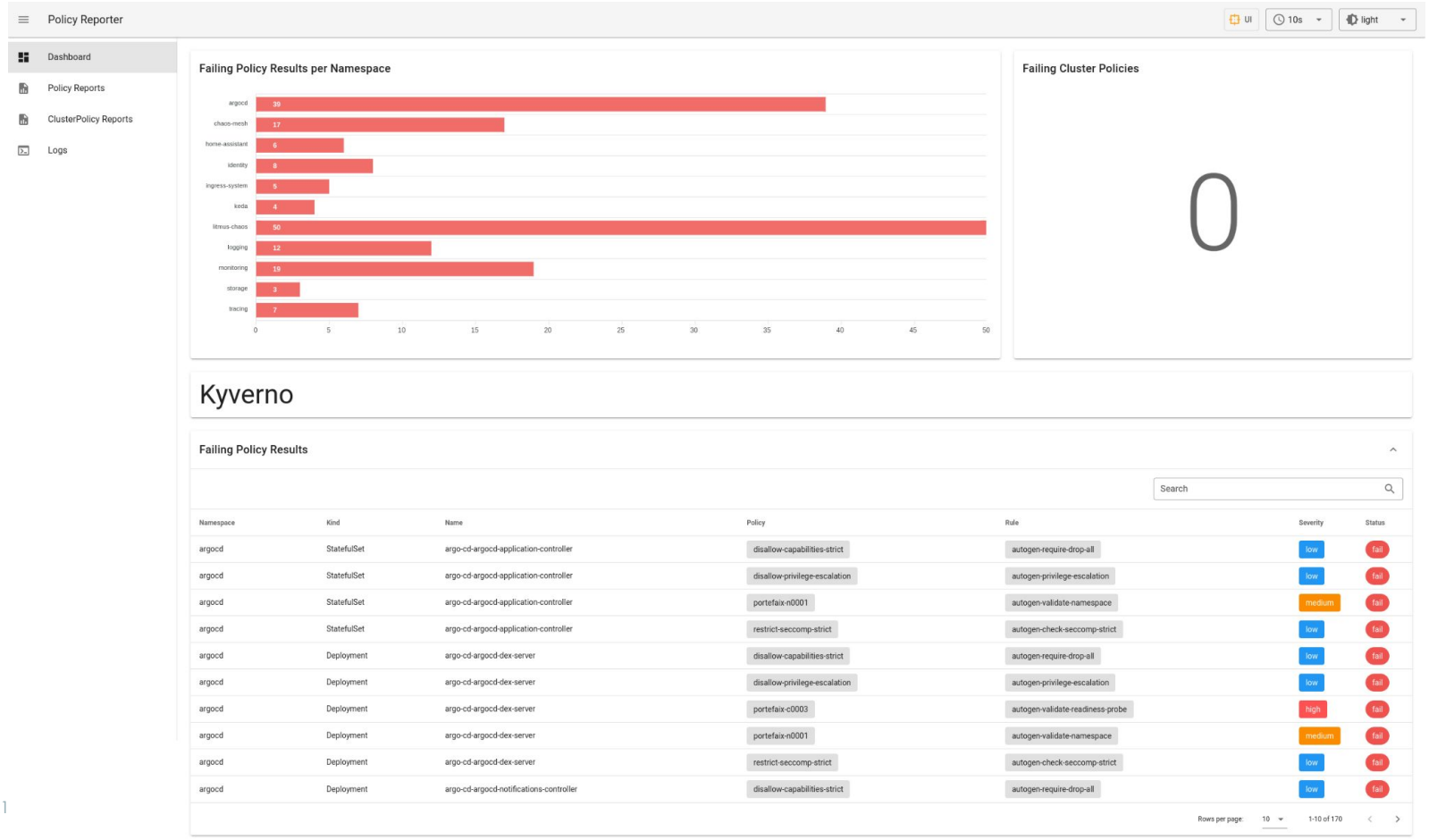
Best Practices

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: portefaix-C0002
  annotations:
    policies.kyverno.io/title: Container must set liveness probe
    policies.kyverno.io/category: Portefaix
    policies.kyverno.io/severity: high
    policies.kyverno.io/subject: Container
    policies.kyverno.io/description: >-
      Liveness probe need to be configured to correctly manage a pods
      lifecycle during deployments, restarts, and upgrades. For each pod,
      a periodic `livenessProbe` is performed by the kubelet to determine
      if the pod's containers are running or need to be restarted
spec:
  validationFailureAction: audit
  rules:
  - name: validate-liveness-probe
    match:
      resources:
        kinds:
        - Pod
    validate:
      message: "Liveness probe is required."
      pattern:
        spec:
          containers:
          - livenessProbe:
              periodSeconds: "?*"
```

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: portefaix-C0008
  annotations:
    policies.kyverno.io/title: Container resource constraints must be specified
    policies.kyverno.io/category: Portefaix
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Container
    policies.kyverno.io/description: >-
      It is important to limit resources requested and consumed by each pod.
spec:
  background: true
  validationFailureAction: audit
  rules:
  - name: check-container-resources
    match:
      resources:
        kinds:
        - Pod
    validate:
      message: "CPU and memory resource requests and limits are required."
      pattern:
        spec:
          containers:
          - resources:
              limits:
                memory: "?*"
              requests:
                cpu: "?*"
                memory: "?*"
```

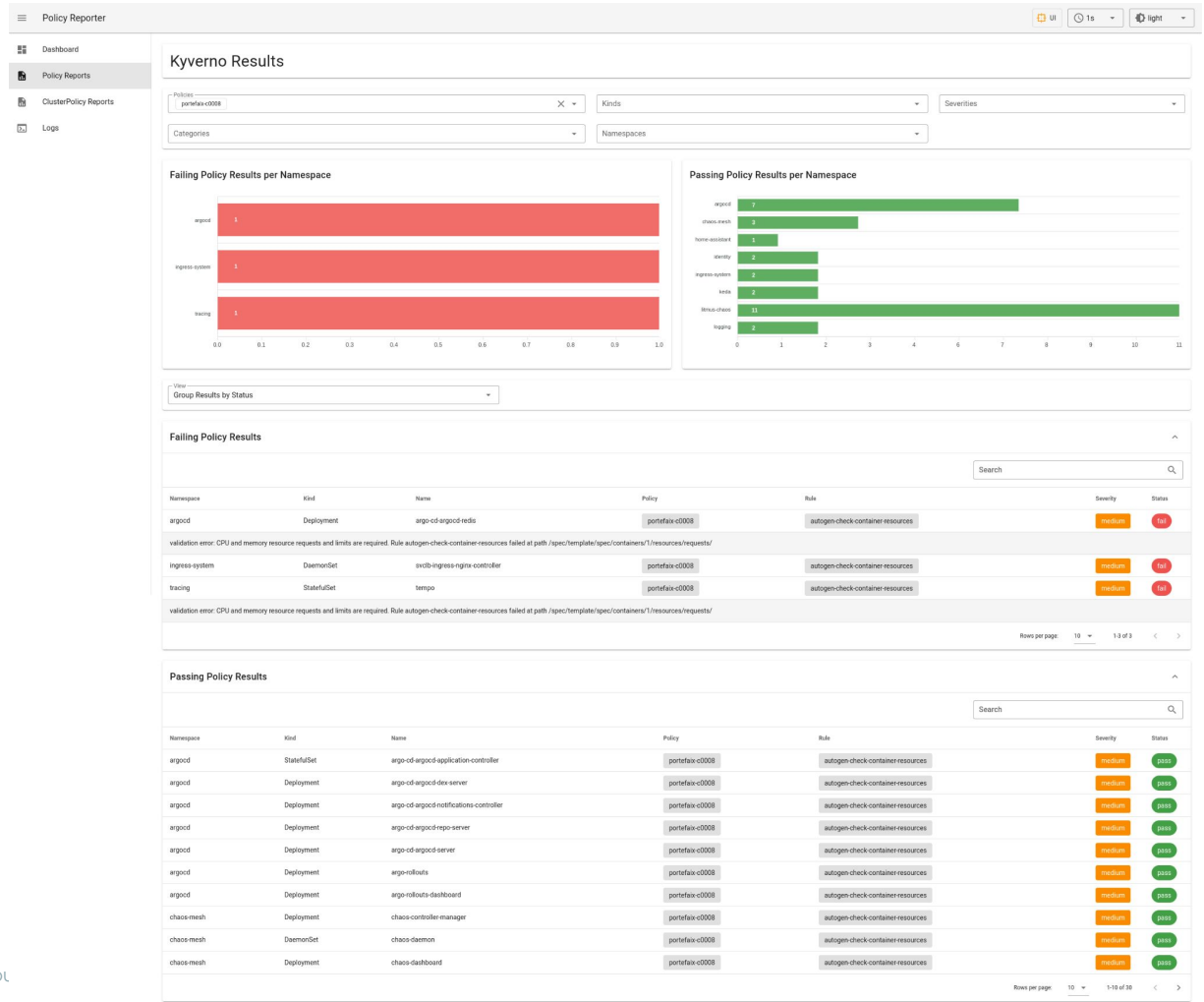
```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: portefaix-N0001
  annotations:
    policies.kyverno.io/title: Disallow Default Namespace
    policies.kyverno.io/category: Portefaix
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Namespace
    policies.kyverno.io/description: >-
      This policy validates that Pods specify a Namespace name other than `default`.
spec:
  validationFailureAction: audit
  rules:
  - name: validate-namespace
    match:
      resources:
        kinds:
        - Pod
    validate:
      message: "Using 'default' namespace is not allowed."
      pattern:
        metadata:
          namespace: "!default"
  - name: validate-podcontroller-namespace
    match:
      resources:
        kinds:
        - DaemonSet
        - Deployment
        - Job
        - StatefulSet
    validate:
      message: "Using 'default' namespace is not allowed for pod controllers."
      pattern:
        metadata:
          namespace: "!default"
```


Reporter



 |

Reporter



 |

Kubewarden

CLOUD NATIVE
COMPUTING FOUNDATION

Description

- Open Source
- Provient de Suse. Puis CNCF (Sandbox)
- Policy:
  - Écrite en Rust, Go, Typescript, ….
  - Compilée en modules WebAssembly
  - Distribuée au format OCI
  - Signée et vérifiée par Sigstore
- Addons :
  - ~~Policy Hub : https://hub.kubewarden.io/~~
  - Artifact Hub : Artifact Hub Kubewarden policies
  - UI dans Rancher Manager

Validation des labels

```yaml
---
apiVersion: policies.kubewarden.io/v1alpha2
kind: ClusterAdmissionPolicy
metadata:
  name: portefaix-M0001
spec:
  module: registry://ghcr.io/kubewarden/policies/safe-labels:v0.1.7
  rules:
    - apiGroups: ["", "apps", "batch"]
      apiVersions: ["v1"]
      resources:
      - ConfigMap
      - Secret
      - Pod
      - Deployment
      - DaemonSet
      - StatefulSet
      - Service
      - Job
      - CronJob
      - Role
      - RoleBinding
      - ClusterRole
      - ClusterRoleBinding
      - ServiceAccount
      - Ingress
      - IngressClass
      - NetworkPolicy
      - PodDisruptionBudget
      - StorageClass
      operations:
      - CREATE
      - UPDATE
  mutating: false
  settings:
    denied_labels: []
    mandatory_labels:
    - app.kubernetes.io/name
    - app.kubernetes.io/instance
    - app.kubernetes.io/version
    - app.kubernetes.io/component
    - app.kubernetes.io/part-of
    - app.kubernetes.io/managed-by
    constrained_labels: []
```

Questions

CLOUD NATIVE
COMPUTING FOUNDATION