Kyverno – Policy As Code

Install Kyverno on EKS

AWS_PROFILE=vuninhnguyen helm install kyverno kyverno/kyverno -n kyverno --create-namespace --set replicaCount=1

1. Setup Label Validation in Namespace Policy as following:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> cat require_label_ns.yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: require-labels
  annotations:
    policies.kyverno.io/title: Require labels
    policies.kyverno.io/category: Best practice
    policies.kyverno.io/severity: medium
    policies.kyverno.io/subject: Namespace
    policies.kyverno.io/description: >-
      Require a app.kubernetes.io/costcentre label
spec:
  validationFailureAction: Enforce
  background: false
  rules:
  - name: check-for-labels
    match:
      resources:
        kinds:
        - Namespace
    validate:
      message: "The label `app.kubernetes.io/costcentre` is required."
      pattern:
        metadata:
          labels:
            app.kubernetes.io/costcentre: "?*"
```

2. Apply the policy and test as following:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl apply -f require_label_ns.yaml
clusterpolicy.kyverno.io/require-labels configured
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl create ns kyverno-test-ns
Error from server: admission webhook "validate.kyverno.svc-fail" denied the request:

resource Namespace//kyverno-test-ns was blocked due to the following policies

require-labels:
  check-for-labels: 'validation error: The label `app.kubernetes.io/costcentre` is
    required. rule check-for-labels failed at path /metadata/labels/app.kubernetes.io/costcentre/'
```

3. Test to create new namespace without label described and get failed:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl apply -f create-kyverno-test-ns-without-label.yaml
Error from server: error when creating "create-kyverno-test-ns-without-label.yaml": admission webhook "validate.kyverno.svc-fail" denied the request:

resource Namespace//kyverno-testing was blocked due to the following policies

require-labels:
  check-for-labels: 'validation error: The label `app.kubernetes.io/costcentre` is
    required. rule check-for-labels failed at path /metadata/labels/app.kubernetes.io/costcentre/'
```

4. Re-test with label included and succeed:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> cat create-kyverno-test-ns-with-label.yaml
apiVersion: v1
kind: Namespace
metadata:
  labels:
    app.kubernetes.io/costcentre: "engineering"
    name: kyverno-testing
  name: kyverno-testing
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl apply -f create-kyverno-test-ns-with-label.yaml
namespace/kyverno-testing created
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl get ns --show-labels
NAME                      STATUS   AGE    LABELS
default                   Active   127m   kubernetes.io/metadata.name=default
kube-node-lease           Active   127m   kubernetes.io/metadata.name=kube-node-lease
kube-public               Active   127m   kubernetes.io/metadata.name=kube-public
kube-system               Active   127m   kubernetes.io/metadata.name=kube-system
kyverno                   Active   115m   kubernetes.io/metadata.name=kyverno
kyverno-testing           Active   11s    app.kubernetes.io/costcentre=engineering,kubernetes.io/metadata.name=kyverno-testing,name=kyverno-testing
```

5. Create a Policy to validate and mutate the resource Quota on namespace

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> cat add-ns-quota.yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-ns-quota
  annotations:
    policies.kyverno.io/title: Add Quota
    policies.kyverno.io/category: Multi-Tenancy
    policies.kyverno.io/subject: ResourceQuota
    policies.kyverno.io/description: >-
      This policy will generate ResourceQuota resources
      when a new Namespace is created.
spec:
  rules:
  - name: generate-resourcequota
    match:
      resources:
        kinds:
        - Namespace
    generate:
      apiVersion: v1
      kind: ResourceQuota
      name: default-resourcequota
      synchronize: true
      namespace: "{{request.object.metadata.name}}"
      data:
        spec:
          hard:
            requests.cpu: '100m'
            requests.memory: '1Gi'
            limits.cpu: '500'
            limits.memory: '1.5Gi'
            requests.storage: '10Gi'
            persistentvolumeclaims: 5
```

6. Test to create a namespace without quota defined

   As you see we get denied because of the namespace manifest doesn't include the label as required in the previous policy

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl apply -f add-ns-quota.yaml
clusterpolicy.kyverno.io/add-ns-quota created
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl create ns kyverno-test-without-quota
Error from server: admission webhook "validate.kyverno.svc-fail" denied the request:

resource Namespace//kyverno-test-without-quota was blocked due to the following policies

require-labels:
  check-for-labels: 'validation error: The label `app.kubernetes.io/costcentre` is
    required. rule check-for-labels failed at path /metadata/labels/app.kubernetes.io/costcentre/'
```

7. Re-test a new namespace with required label and get succeed

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> cat create-kyverno-test-ns-with-label-without-quota.yaml
apiVersion: v1
kind: Namespace
metadata:
  labels:
    app.kubernetes.io/costcentre: "engineering"
    name: kyverno-testing-no-quota
  name: kyverno-testing-no-quota
```

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl apply -f create-kyverno-test-ns-with-label-without-quota.yaml
namespace/kyverno-testing-no-quota created
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl -n kyverno-testing-no-quota resourceQuotas -oyaml
Error: flags cannot be placed before plugin name: -n
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main) [1]> kubectl -n kyverno-testing-no-quota get  resourceQuotas -oyaml
apiVersion: v1
items:
- apiVersion: v1
  kind: ResourceQuota
  metadata:
    creationTimestamp: "2024-06-28T03:07:16Z"
    labels:
      app.kubernetes.io/managed-by: kyverno
      generate.kyverno.io/policy-name: add-ns-quota
      generate.kyverno.io/policy-namespace: ""
      generate.kyverno.io/rule-name: generate-resourcequota
      generate.kyverno.io/trigger-group: ""
      generate.kyverno.io/trigger-kind: Namespace
      generate.kyverno.io/trigger-namespace: ""
      generate.kyverno.io/trigger-uid: c974569e-4656-42bd-9104-301cd24dfeb7
      generate.kyverno.io/trigger-version: v1
    name: default-resourcequota
    namespace: kyverno-testing-no-quota
    resourceVersion: "20597"
    uid: f5d3e19b-0068-4665-909e-08e2ff55d00e
  spec:
    hard:
      limits.cpu: "500"
      limits.memory: 1536Mi
      persistentvolumeclaims: "5"
      requests.cpu: 100m
      requests.memory: 1Gi
      requests.storage: 10Gi
  status:
    hard:
      limits.cpu: "500"
# Please edit the object below. Lines beginning with a '#' will be ignored,
```

8. Add the custom label for new created pod:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> cat add-pods-default-label.yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-label-pods
  annotations:
    policies.kyverno.io/title: Add default label
    policies.kyverno.io/category: Best practice
    policies.kyverno.io/severity: low
    policies.kyverno.io/subject: Label
    policies.kyverno.io/description: >-
      This policy performs a simple mutation which adds a label
      `type=user` to Pods, Services, ConfigMaps, and Secrets.
spec:
  rules:
  - name: add-label
    match:
      resources:
        kinds:
        - Pod
    mutate:
      patchStrategicMerge:
        metadata:
          labels:
            type: user
```

9. Test to create a pod with the following manifest:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> cat create-pod-to-test-auto-add-label.yaml
apiVersion: v1
kind: Pod
metadata:
  name: busybox1
  labels:
    app: busybox1
spec:
  containers:
  - image: busybox:latest
    command:
      - sleep
      - "3600"
    imagePullPolicy: IfNotPresent
    name: busybox
  restartPolicy: Always
```

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl apply -f create-pod-to-test-auto-add-label.yaml
pod/busybox1 created
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
busybox1   1/1     Running   0          21s
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl describe pod busybox1
Name:             busybox1
Namespace:        default
Priority:         0
Service Account:  default
Node:             ip-10-0-3-80.us-east-2.compute.internal/10.0.3.80
Start Time:       Fri, 28 Jun 2024 11:24:59 +0700
Labels:           app=busybox1
                  type=user
Annotations:      <none>
Status:           Running
```

10. All Policy installed on cluster:

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main)> kubectl get clusterpolicy
NAME              ADMISSION   BACKGROUND   VALIDATE ACTION   READY   AGE    MESSAGE
add-label-pods    true        true         Audit             True    104m   Ready
add-ns-quota      true        true         Audit             True    87m    Ready
require-labels    true        false        Enforce           True    126m   Ready
```

```
ninhnv@ninhnv-macpro ~/t/m/kyverno-policy (main) [1]> kubectl get ns --show-labels
NAME                      STATUS   AGE    LABELS
default                   Active   156m   kubernetes.io/metadata.name=default
kube-node-lease           Active   156m   kubernetes.io/metadata.name=kube-node-lease
kube-public               Active   156m   kubernetes.io/metadata.name=kube-public
kube-system               Active   156m   kubernetes.io/metadata.name=kube-system
kyverno                   Active   143m   kubernetes.io/metadata.name=kyverno
kyverno-testing           Active   28m    app.kubernetes.io/costcentre=engineering,kubernetes.io/metadata.name=kyverno-testing,name=kyverno-testing
kyverno-testing-no-quota  Active   82m    app.kubernetes.io/costcentre=engineering,kubernetes.io/metadata.name=kyverno-testing-no-quota,name=kyverno-testing-no-quota
```