



(공유) Kyverno Policy 적용

🌀 Introduction

Kubernetes용 Policy Engine인 Kyverno를 활용하여 위험도 높은 사용자 Operation을 System 레벨에서 방지 (Resource Lock)

- **Kyverno란?**

Kubernetes 전용의 Policy Engine으로, Kubernetes의 Admission Control 단계에서 Mutating(변경)/Validating(검증) 기능을 수행 예를 들어, 1) Resource가 생성될 때마다 Label “app=my-app” 이 자동으로 추가되게끔 **Mutating Policy**를 구현하거나 사용자가 주요 Resource 삭제 요청을 했더라도 Kubernetes에서 이를 자동으로 거절하도록 즉, 2) Resource가 삭제되지 않도록(Lock)하는 **Validating Policy** 구현할 수 있음

- **Precondition**

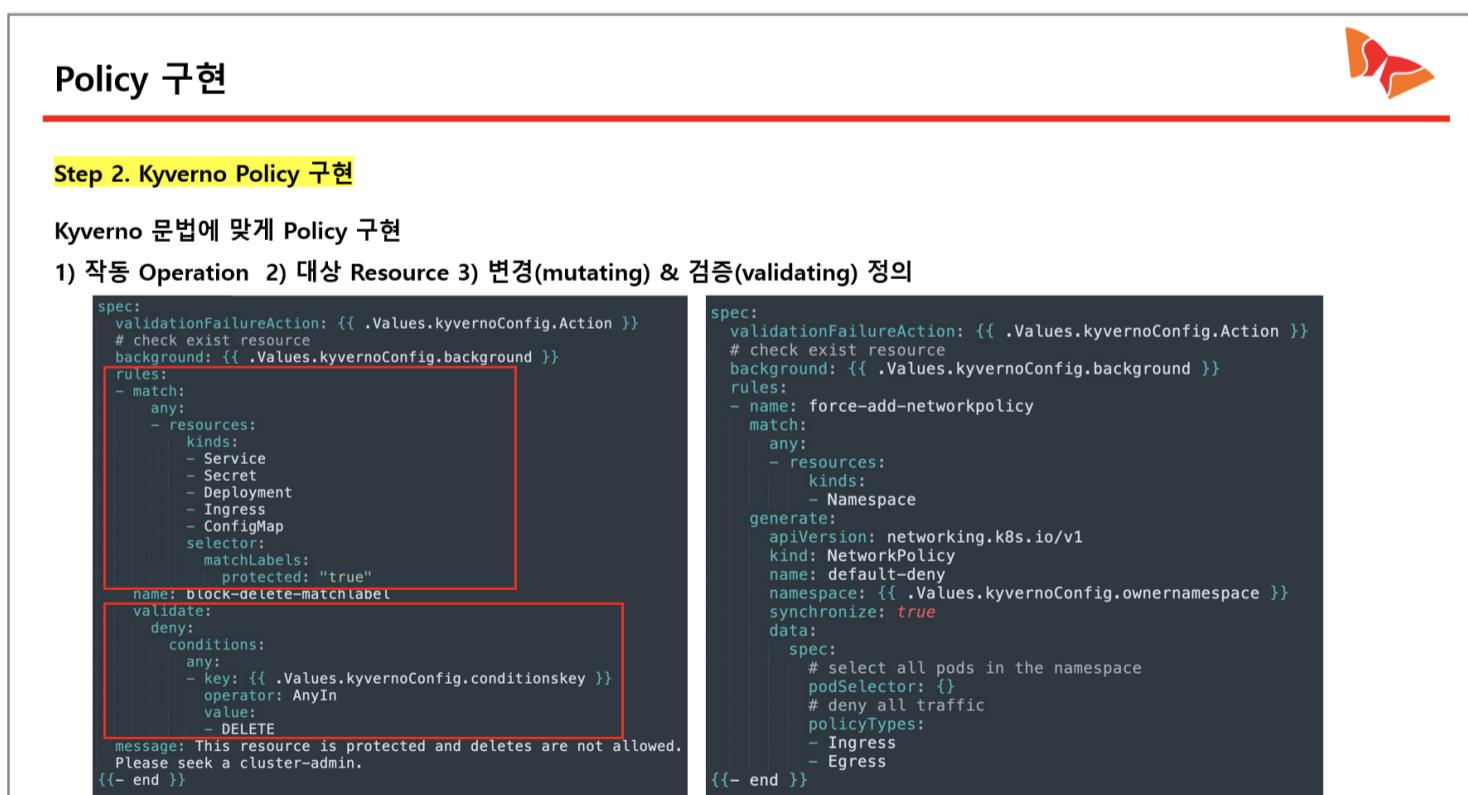
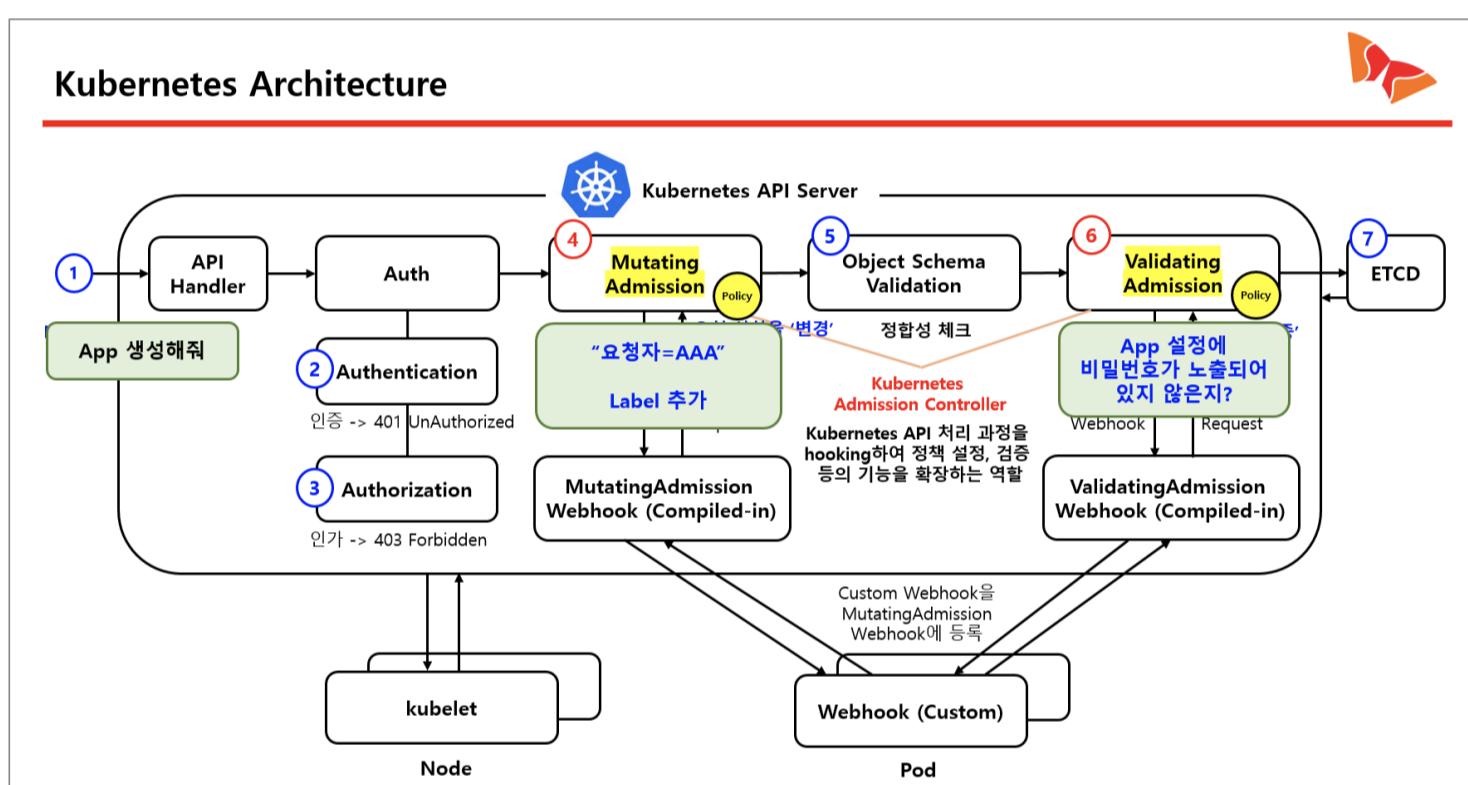
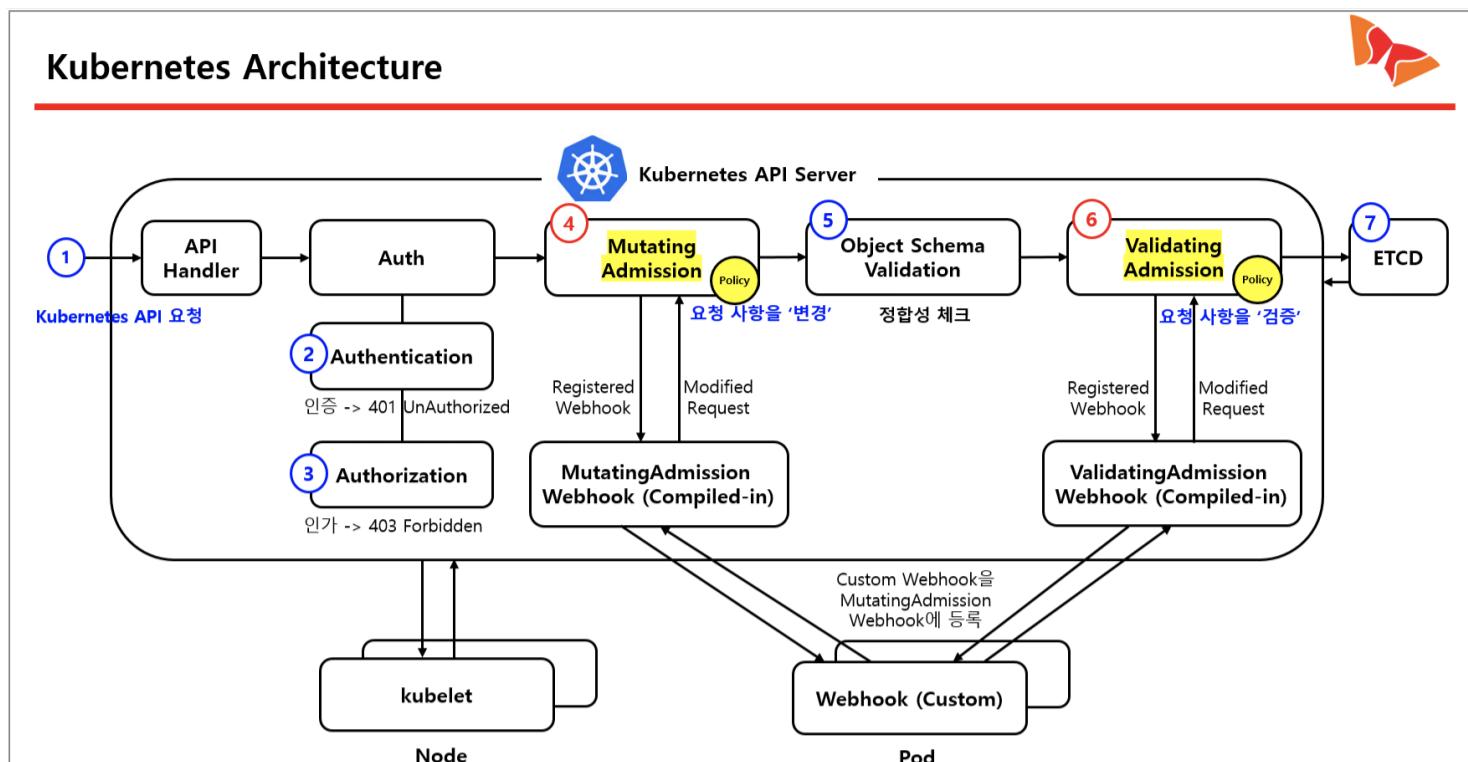
Kubernetes Architecture에 대한 사전 이해가 필요하고, Kyverno 컨셉 및 문법을 익혀야 함

Kubernetes Docs : <https://kubernetes.io/docs/concepts/security/controlling-access/>

Kyverno Docs : <https://kyverno.io/docs/>

- ✓ LCL(Learning Collabo Lab) 프로젝트를 통해 운영 환경에 적용할 표준 Policy를 연구, 각 운영 환경에 전파하여 운영 부담을 덜고자 함
- ✓ 특히 고객사 운영 환경에서 발생했던 장애의 개선 과제로써도 활용하여 운영자 Operation 오류로 인한 동일 이슈를 방지하는데 목적을 둠
- ✓ 참고로 OPA(Open Policy Agent)도 널리 사용되고 있으며 Code 재사용 여부, 관리 범위 축소 등을 고려해 Kyverno로 구현

◎ LCL : Kubernetes Admission Controller 기능을 활용한 운영 Policy 연구 (요약)



Policy 구현



Step 3. TEST

변경(Mutating) & 검증(Validating) 기능 작동, 미리 정의한 Message 출력

```
Bash v
# 테스트
% kubectl delete svc lb-svc
Error: uninstallation completed with 2 error(s): admission webhook "validate.kyverno.svc-fa
policy Service/kube-system/private-ingress-nginx-controller" for resource violation:
block-updates-deletes:
block-updates-deletes: This resource is protected and changes are not allowed. Please
seek a cluster-admin.
```

LoadBalancer 타입의 Service 삭제 시도 시 Error 발생

1) 변경 (Mutating)

- ✓ Label 추가
- ✓ Application 분산 설정 추가
- ✓ Job에 TTLSecondsAfterFinished 설정 추가
- ✓ Image의 Pull 정책 강제 설정
- ✓ emptyDir 크기 제한
-

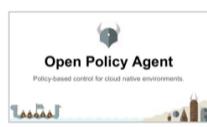
2) 검증 (Validating)

- ✓ Namespace 삭제 여부 확인
- ✓ LoadBalancer 타입의 Service 삭제 여부 확인
- ✓ Deprecated API 사용 여부 확인
- ✓ Image Tag-Version) 지정 여부 확인
- ✓ Ingress의 Host 중복 여부 확인
-

OPA vs Kyverno

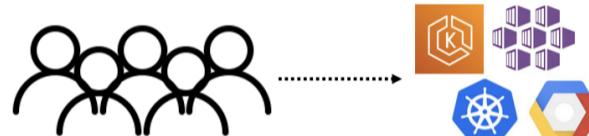


VS

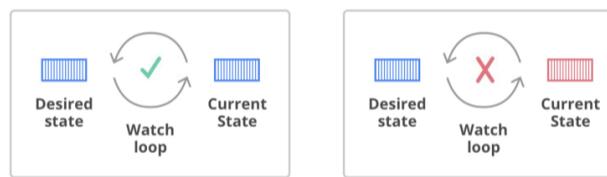


1) 사용자 친화적인 정책을 사용

- 최근 Container 환경이 급격히 늘면서 운영자의 수도 많아짐
- Kubernetes 용으로 만들어져 별도의 Knowledge가 필요하지 않음



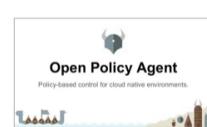
2) Kubernetes의 선언형(Declarative) 디자인을 따름



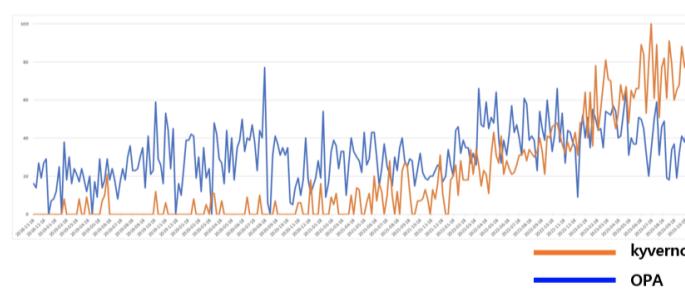
OPA vs Kyverno



VS



3) 최근 Kyverno의 사용률이 OPA를 넘어서고 있음



4) 최근 많이 사용되면서 Reference가 다양해짐

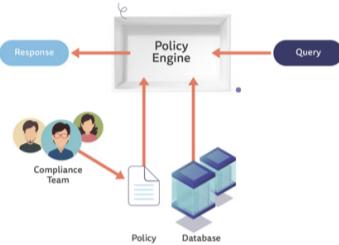


활용 방안 및 효과



Policy as Code(PaC) 강화

- ✓ 대다수 Application들이 MSA 기반으로 설계되면서 동일 서비스 내에서 수많은 Module이 생성되고 있음
- ✓ Module 간 정책 적용 및 관리 복잡성이 증가하며, 이로 인한 관리 Risk 증가
- ✓ PaC 적용 및 강화를 통해 Kubernetes Cluster 내 Module 간 일관된 정책 적용 및 관리/보안 Risk 감소



Kubernetes PSP 대체

- ✓ PSP(Pod Security Policies)는 Pod 명세의 보안 관련 측면을 제어하는 Cluster 수준의 리소스
- ✓ Kubernetes v1.25 부터 PSP가 완전히 제거되어, Cluster 정책 관리 대안이 필요
- ✓ Kyverno와 같은 Kubernetes 전용의 정책 엔진을 적용하여 강화된 보안 체계 확보



활용 방안 및 효과



운영 안정성 및 통제력 확보

- ✓ 서비스 가용성을 침해할 수 있는 **운영자(Human)** 실수를 사전에 차단하여 서비스 장애 예방
- ✓ 정책 관리 감독 주체를 System으로 이관함으로써, 운영자 부담 완화 및 운영 보안 수준 강화
- ✓ 운영 체계 변화(Offshoring 확대 등)를 대비한 **Advanced Management/Controllability** 환경 구현



Cost Down, Quality Up !

- ✓ Open Source S/W 기반 구현을 통한 추가 비용 Zero
- ✓ 장애 및 불필요한 Communication 감소를 통한 Operation Cost Down
- ✓ Policy 기반 관리 체계 확보를 통한 운영 품질 향상
- ✓ 운영 품질 개선을 통한 고객 신뢰 확보 및 Managed Service 확대 적용을 통한 Biz. Value 창출



◎ Postmortem : 개선 과제

장애명 : [REDACTED] 접속 불가 등 작업 대상 오류 장애 [REDACTED] Postmortem

개발/운영 구(舊)/신(新) 서버 등 변경 작업 대상 착오로 인한 서비스 접속 불가

| 장애 개요 | 개선 사항 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">고객사 : [REDACTED]장애 내용 요약 : 작업 대상 오류로 인한 서비스 장애 발생<ul style="list-style-type: none">- K8s 기반 시스템의 변경작업을 위해 사전 테스트 목적으로 개발환경에서 수행하려 하였으나, 운영 환경에 적용하여 시스템 접속 불가 발생 ([REDACTED] 서비스)- DB서버 교체 작업을 위한 사전작업 중 신규 서버에 변경해야 하는 것을 기 운영 서버에서 작업하여 서비스 불가 장애 발생 ([REDACTED]) | <p>“장애 재발 방지를 위한 안전망 확보 및 빠른 장애 인지”</p> <ul style="list-style-type: none">운영자 실수가 장애로 이어지지 않기 위한 안전망 구현<ul style="list-style-type: none">- CLI 기반으로 자원의 생성 및 삭제가 상대적으로 용이한 K8s 환경 특성을 고려하여, 주요 리소스 생성/변경/삭제 관련 정책(Policy) 생성 및 운영 환경 일괄 적용※ Kyverno와 같은 K8s Admission Controller Tool 활용- 서비스 장애와 직결되는 K8s 주요 리소스 삭제 방지 및 Cloud Storage 자원에 대해서는 Deletion Lock or Soft Deletion (삭제 후 일정 기간 내 복구 가능) 기능 활성화 |
| <p>장애 조치과정에서 아쉬웠던 점</p> <ul style="list-style-type: none">운영 환경의 주요 리소스 변경 및 삭제 보호 장치 부재<ul style="list-style-type: none">- 변경 작업 대상 착오가 있었다 하더라도, 주요 리소스 Lock 등의 보호 장치가 있었다면 서비스 장애로 이어지지 않도록 예방할 수 있었음.장애 발생 시 선제적 인지 불가로 인한 장애 시간 증가<ul style="list-style-type: none">- 서비스 모니터링 적용 미비로 인해 작업 오류에 대한 운영자 인지 불가 (사용자 Call을 통한 장애 접수 및 장애 시간 증가) | <p>사용자 측면의 장애 인지 방안 확보 및 전파 체계 확보</p> <ul style="list-style-type: none">- Public Domain과 달리 Private Domain 사용 서비스 대상 URL HealthCheck 모니터링 적용 미흡- 고객사 폐쇄망 내 Proxy 서버 구성 등을 활용하여 Private URL HealthCheck 모니터링 환경 구성- 단순 Main Page 접속 가능 여부가 아닌 서비스 장애를 판단할수 있는 Custom URL 개발 필요- Opsgenie, Slack 등의 Tool을 활용하여 Infra & Appl. 운영자에게 신속한 Call 전파 및 Comm. 환경 구현 |

◎ Install

Kubernetes 운영 환경에 설치하여 사용

오픈 소스 [Kyverno](#)를 먼저 설치하고, 운영 환경에 맞게 Policy를 구현하여 묶은 [CP Policy Engine](#)을 올림

Kyverno, CP Policy Engine 모두 Helm Packaging 되어있으며 아래는 '24년 1월 Kubernetes v1.28에서 검증한 버전으로 작성

1. Kyverno : Kubernetes용 Policy 구현을 위한 베이스 Engine
2. CP Policy Engine : 운영 환경 기준 Policy들의 묶음

운영 환경이므로 admissionController는 반드시 최소 2개 이상(3개 권장) 구성

1개로 단일 구성할 경우 admissionController 이슈 발생 시 모든 API 요청이 Kyverno 서비스 호출 단계에서 block되는 상황이 발생할 수 있음

```
# 0. 운영 Add-on 설치용 Namespace 생성
```

```
kubectl create ns cp-system
```

```
# 1. Kyverno 설치
```

```
helm repo add kyverno https://kyverno.github.io/kyverno/
helm repo update
helm install kyverno kyverno/kyverno --namespace cp-system \
--version 3.1.4 \
--set admissionController.replicas=3 \
--set backgroundController.replicas=2 \
--set cleanupController.replicas=2 \
--set reportsController.replicas=2 --debug
```

```
# 참고  
https://kyverno.io/docs/installation/methods/
```

2. CP Policy Engine 설치

```
helm repo add cp-policy-engine {공용 Git / cp-policy-engine 주소}  
helm repo update & helm repo list  
helm install cp-policy-engine -n cp-system cp-addon/cp-policy-engine
```

“2. CP Policy Engine 설치” → 본 자료에서는 Helm Package 파일로 설명

```
% tree  
.---- README.md  
|---- cp-policy-engine  
|    |---- Chart.yaml  
|    |---- templates  
|    |    |---- _helpers.tpl  
|    |    |---- cp-mutating_force-add-affinity.yaml  
|    |    |---- cp-mutating_force-add-creator.yaml  
|    |    |---- cp-validating_block-create-ephcontainer.yaml  
|    |    |---- cp-validating_block-delete-crd.yaml  
|    |    |---- cp-validating_block-delete-ns.yaml  
|    |    |---- cp-validating_block-delete-resource.yaml  
|    |    |---- cp-validating_block-update-resource.yaml  
|    |    |---- cp-validating_disallow-set-defaultns.yaml  
|    |    |---- cp-validating_disallow-set-deprecated-registry.yaml  
|    |    |---- cp-validating_require-set-imagetag.yaml  
|    |    |---- cp-validating_require-set-multireplica.yaml  
|    |    |---- cp-validating_require-set-storageclass.yaml  
|    |---- values.yaml  
|---- cp-policy-engine-1.1.1.tgz  
|---- index.yaml
```

```
# values.yaml  
  
kyverno:  
  enabled: true  
  
# Kyverno Config  
kyvernoConfig:  
  # Validation failure action (`audit`, `enforce`)  
  # Warnings : enforce/audit are deprecated, use Enforce/Audit instead  
  validationFailureAction: Enforce  
  
  # Define validationFailureActionOverrides for specific policies  
  # The overrides for `all` will apply to all policies  
  validationFailureActionOverrides:  
    all: []  
    # all:  
    #     - action: audit  
    #       namespaces:  
    #         - cp-system  
    # blockDeleteMatchlabel:
```

```

#     - action: audit
#       namespaces:
#         - app-system

# Policies background mode (`true`, `false`)
background: false

# Policies Title
annotations:
  policies.kyverno.io/title: kyverno

# -- Additional policies
conditionskey: "'{{request.operation || ''BACKGROUND''}}'"

# Define validationFailureActionByPolicy for specific policies
kyvernoPolicy:
  # Specifies whether a 'block-delete-allsvc' should be created

  # Mutating
  forceAddCreator: true
  forceAddAffinity: true

  # Validating
  blockDeleteNs: true
  blockDeleteCrd: true
  blockDeleteResource: true
  blockUpdateResource: true
  blockCreateEphcontainer: true
  disallowSetDefaultns: true
  disallowSetDeprecatedrepo: true
  requireSetImagetag: true
  requireSetMultireplica: true
  requireSetStorageclass: true

```

추가로, Nginx Ingress 자원이 Namespace “kube-system” 으로 설치되어 있는 경우(혹은 주요 자원이 kube-system에 설치되어 있는 경우)

다음 작업을 추가로 수행해 주어야 함  **바로가기** : [Reference](#)

```

# kubectl edit cm kyverno -n cp-system

resourceFilters: '[/*/*,cp-system,*] [Event,*,*] [/*/*,kube-system,*] [/*/*,kube-public,*]...
-----  
-> 이 부분 삭제 후 저장

```

🌀 UnInstall

```

# Resource 삭제 정책 중단

kubectl edit cpol validating-block-delete-resource
validationFailureAction 설정이 Enforce일 경우 Audit으로 변경 후 저장

# cp-policy-engine 삭제

helm uninstall -n cp-system cp-policy-engine

```

🌀 Mutating Policy

▼ 1. 주요 Resource 생성 시 Annotation에 Creator 정보 추가

ClusterPolicy/Mutating/-force-add-creator

적용 예시

```
metadata:  
  annotations:  
    creator: system:serviceaccount:cp-system:ckops-admin
```

```
{{- if .Values.kyvernoPolicy.forceAddCreator -}}  
apiVersion: kyverno.io/v1  
kind: ClusterPolicy  
metadata:  
  {{- with .Values.kyvernoConfig.annotations }}  
  annotations:  
    {{- toYaml . | nindent 4 }}  
    policies.kyverno.io/title: Force Add Creator Annotation  
    policies.kyverno.io/category: Standard  
    policies.kyverno.io/subject: Annotation  
    policies.kyverno.io/description: This policy adds the creator information annotation to the resource.  
  {{- end }}  
  name: mutating-force-add-creator  
spec:  
  validationFailureAction: {{ .Values.kyvernoConfig.validationFailureAction }}  
  # check exist resource  
  background: {{ .Values.kyvernoConfig.background }}  
  rules:  
  - name: force-add-creator  
    match:  
      resources:  
        kinds:  
        - Deployment  
        - ReplicaSet  
        - Pod  
        - Ingress  
        - Service  
        - ConfigMap  
        - Secret  
    mutate:  
      patchStrategicMerge:  
        metadata:  
          annotations:  
            creator: "request.userInfo.username"  
  {{- end }}
```

▼ 2. Deployment 생성 시 podAntiAffinity 추가 (yaml 생략)

ClusterPolicy/mutating-force-add-affinity

🌀 Validating Policy

▼ 1. 모든 Namespace 삭제 방지

ClusterPolicy/Validating/block-delete-ns

```
{{- if .Values.kyvernoPolicy.blockDeleteNs -}}
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
{{- with .Values.kyvernoConfig.annotations -}}
annotations:
{{- toYaml . | nindent 4 -}}
policies.kyverno.io/title: Block Delete All Namespaces
policies.kyverno.io/category: Standard
policies.kyverno.io/subject: Namespace
policies.kyverno.io/description: This policy restricts deletes to any Namespace resource
{{- end -}}
name: validating-block-delete-ns
spec:
validationFailureAction: {{ .Values.kyvernoConfig.validationFailureAction }}
# check exist resource
background: {{ .Values.kyvernoConfig.background }}
rules:
- match:
  any:
    - resources:
      kinds:
        - Namespace
  name: block-delete-ns
  validate:
    deny:
      conditions:
        any:
          - key: {{ .Values.kyvernoConfig.conditionskey }}
            operator: AnyIn
            value:
              - DELETE
  message: |
=====
[ClusterPolicy/Validating/block-delete-ns]
This policy restricts deletes to any Namespace resource.

Information :
  This Cluster has policies applied using Kyverno.
  Kyverno is a policy engine for Kubernetes that allows you to enforce custom policies.

  We are applying policies such as preventing the deletion of all namespaces,
  blocking the removal of resources with specific labels, and restricting the ImagePullPolicy.

  If you would like to know more details, Please seek a Admin. (email : ckops@sk.com)
=====

{{- end -}}
```

▼ 2. 모든 CRD 삭제 방지 (yaml 생략)

ClusterPolicy/Validating/block-delete-crd

▼ 3. Deployment, RS, Pod, STS, DaemonSet 제외 모든 Resource 삭제 방지 (yaml 생략)

ClusterPolicy/Validating/block-delete-resource

▼ 4. Secet, NetworkPolicy 등 일부 Resource 수정 방지 (yaml 생략)

ClusterPolicy/Validating/block-update-resource

▼ 5. Ephemeral Container 사용 금지 (yaml 생략)

ClusterPolicy/Validaint/block-create-ephcontainer

▼ 6. Default Namespace 사용 금지 (yaml 생략)

ClusterPolicy/Validating/disallow-set-defaultns

▼ 7. Container Image로 k8s.gcr.io Registry 사용 금지 (yaml 생략)

ClusterPolicy/Validating/disallow-set-deprecated-registry

▼ 8. Container Image에 latest 태그 금지 (yaml 생략)

CusterPolicy/Validating/require-set-imagetag

▼ 9. Pod Replicas 2개 이상 설정 (yaml 생략)

ClusterPolicy/Validating/require-set-multireplica

▼ 10. StorageClass 지정 필수 (yaml 생략)

ClusterPolicy/Validaing/require-set-storageclass

🌀 Reference

Kyverno에는 Mutating/Validating 작동에 기본적으로 Filtering되는 자원이 있음

- kubectl get cm kyverno -n cp-systmem

.....

```
resourceFilters: '[  
  /cp-system,] [Event,] [/kube-system,]  
  ....
```

위와 같이 Kyverno를 설치한 cp-system과 System용인 kube-system, Event 자원 등 몇가지는 Mutating/Validating 작동에서 제외 됨 즉, 자원 삭제 방지 Policy를 걸어 두었어도 cp-system 에 생성되는 자원은 filtering되어 삭제가 수행되므로 사전 확인 필요

kube-system 자원에도 삭제 방지가 필요할 경우, 반드시 아래와 같이 작업하여 Filter List에서 제외시켜 주어야 함

```
# kubectl edit cm kyverno -n cp-system  
resourceFilters: '[/*,cp-system,*] [Event,*,*] [/*,kube-system,*] [/*,kube-public,*]...  
-----  
-> 이 부분 삭제 후 저장
```

<https://kyverno.io/docs/>

<https://github.com/kyverno/kyverno>

<https://www.openpolicyagent.org/>

<https://github.com/open-policy-agent/opa>

<https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers/>

(끝)