



Kyverno Chronicles:

A DevSecOps Tale

DevOpsDays Istanbul 20/04/2024





#whoami



Consultant and Trainer @Kubermatic

[linkedin.com/in/korayoksay](https://www.linkedin.com/in/korayoksay)

CKA | CKAD | CKS

@korayoksay

Linux Foundation Instructor for CK.*

koksay

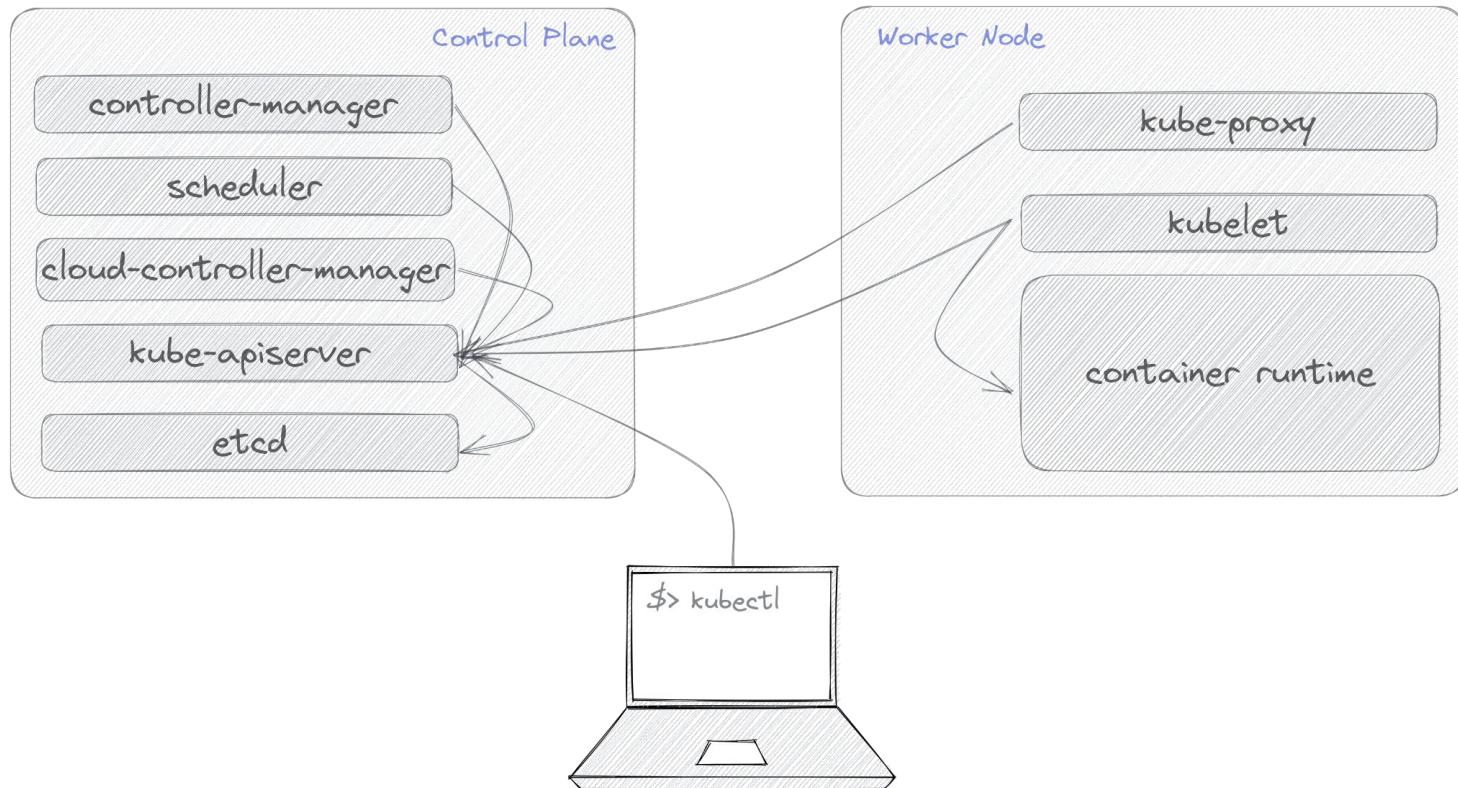
Kubernetes Contributor on [sig-k8s-infra](#)

koray@kubermatic.com

KCD Istanbul Organizer

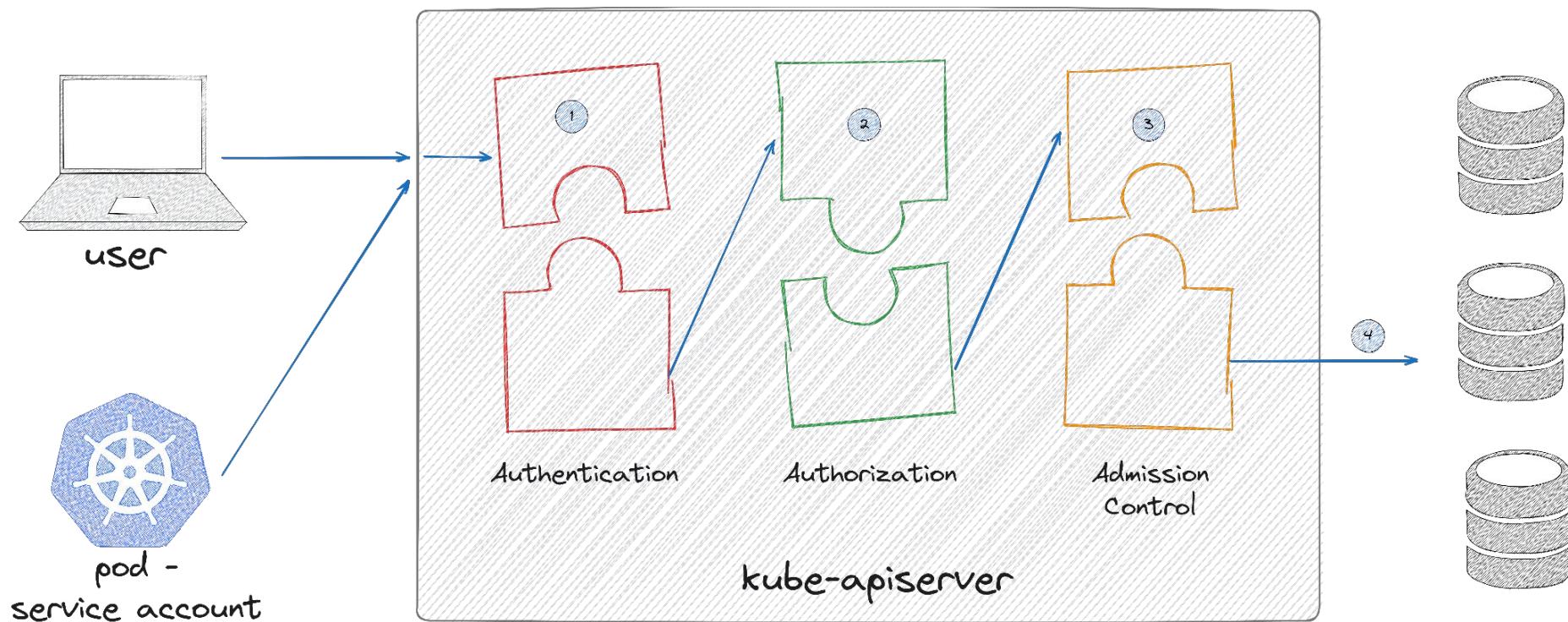


Kubernetes Architecture





Kubernetes API Access Control





Admission Controllers

Compiled-in

- DefaultStorageClass
- LimitRanger
- DefaultStorageClass
- ...



Dynamic

- Validating Admission Webhook
- Mutating Admission Webhook



What is Kyverno?



Greek for “Govern”



Policies as **Kubernetes resources**,
(no new language to learn)



Validate, mutate, generate, or
cleanup (remove) **any resource**



Verify container images for
software supply chain security



What is Kyverno?



Inspect image metadata



Synchronize configurations across
Namespaces



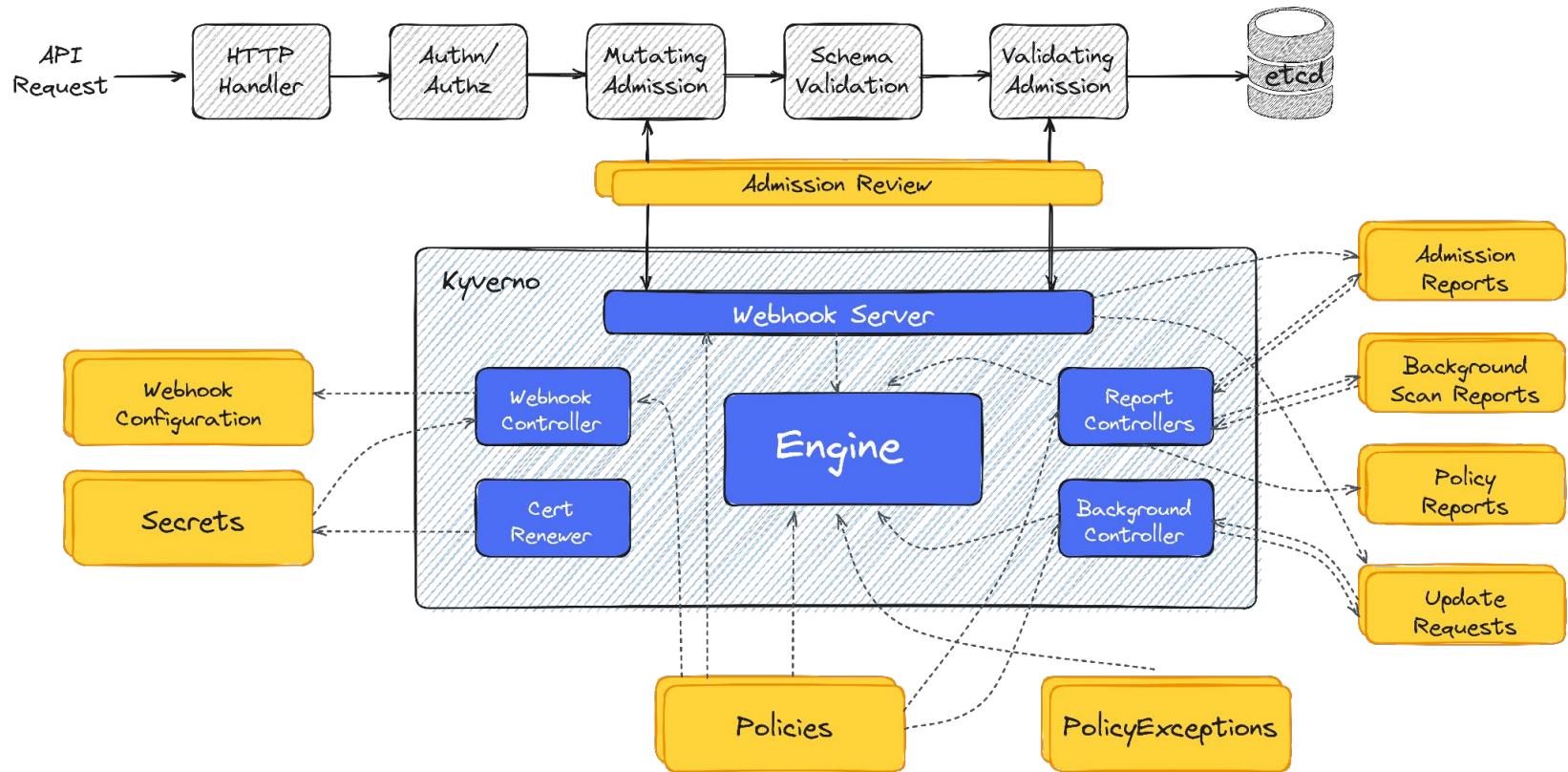
Test policies and validate resources using the **Kyverno CLI**, in your
CI/CD pipeline, **before** applying to your cluster



Deployed Kyverno and policies!



High Level Architecture





Is this High Available?



⚠️ Attention!

Make sure you deploy Kyverno with the **high available** (multiple replicas) option on the helm chart, otherwise you will have issues.

failurePolicy

Defines the API server behavior if the webhook fails to respond. Allowed values are “Ignore” or “Fail”.



Some Policies



```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-privileged-containers
spec:
  validationFailureAction: audit
  background: true
  rules:
    - name: privileged-containers
      match:
        any:
          - resources:
              kinds:
                - Pod
  validate:
    message: >-
      Privileged mode is disallowed. The fields spec.containers[*].securityContext.privileged
      and spec.initContainers[*].securityContext.privileged must be unset or set to `false` .
    pattern:
      spec:
        =(ephemeralContainers):
          - =(securityContext):
              =(privileged): "false"
        =(initContainers):
          - =(securityContext):
              =(privileged): "false"
      containers:
        - =(securityContext):
            =(privileged): "false"
```





```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: block-stale-images
spec:
  validationFailureAction: audit
  rules:
    - name: block-stale-images
      match:
        any:
          - resources:
              kinds:
                - Pod
      validate:
        message: "Images built more than 6 months ago are prohibited."
        foreach:
          - list: "request.object.spec.containers"
            context:
              - name: imageData
                imageRegistry:
                  reference: "{{ element.image }}"
      deny:
        conditions:
          all:
            - key: "{{ time_since('', '{{ imageData.configData.created }}', '') }}"
              operator: GreaterThan
              value: 4380h
```





```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: reduce-cpu-requests
  annotations:
    policies.kyverno.io/title: Reduce the CPU limits and requests by 10%
spec:
  rules:
    - name: set-cpu-limit-request
      match:
        any:
          - resources:
              kinds:
                - Pod
      exclude:
        any:
          - resources:
              kinds:
                - Pod
              selector:
                matchExpressions:
                  - key: prow.k8s.io/job
                    operator: In
                    values:
                      - periodic-cluster-api-provider-vsphere-test-integration-release
                      - ci-kubernetes-integration-1-27
      mutate:
        foreach:
          - list: request.object.spec.containers
            patchStrategicMerge:
              spec:
                containers:
                  - (name): "{{element.name}}"
                    resources:
                      limits:
                        cpu: "{{ multiply(divide('{{element.resources.limits.cpu}} || {{element.resources.requests.cpu}}', '10'), '9') }}"
                      requests:
                        cpu: "{{ multiply(divide('{{element.resources.requests.cpu}} || {{element.resources.limits.cpu}}', '10'), '9') }}
```





```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-networkpolicy
  annotations:
    policies.kyverno.io/description: >-
      By default, Kubernetes allows communications across all Pods within a cluster. The NetworkPolicy resource and a CNI plug-in that supports NetworkPolicy must be used to restrict communications. A default NetworkPolicy should be configured for each Namespace to default deny all ingress and egress traffic to the Pods in the Namespace. Application teams can then configure additional NetworkPolicy resources to allow desired traffic to application Pods from select sources. This policy will create a new NetworkPolicy resource named `default-deny` which will deny all traffic anytime a new Namespace is created.
spec:
  rules:
  - name: default-deny
    match:
      any:
        - resources:
            kinds:
              - Namespace
  generate:
    apiVersion: networking.k8s.io/v1
    kind: NetworkPolicy
    name: default-deny
    namespace: "{{request.object.metadata.name}}"
    synchronize: true
    data:
      spec:
        # select all pods in the namespace
        podSelector: {}
        # deny all traffic
        policyTypes:
        - Ingress
        - Egress
```





```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: add-networkpolicy-dns
  annotations:
    policies.kyverno.io/description: >-
      By default, Kubernetes allows communications across all Pods within a cluster. The NetworkPolicy resource and a CNI plug-in that supports NetworkPolicy must be used to restrict communications. A default NetworkPolicy should be configured for each Namespace to default deny all ingress and egress traffic to the Pods in the Namespace. Application teams can then configure additional NetworkPolicy resources to allow desired traffic to application Pods from select sources. This policy will create a new NetworkPolicy resource named `default-deny` which will deny all traffic anytime a new Namespace is created.
spec:
  rules:
  - name: add-netpol-dns
    match:
      any:
        - resources:
            kinds:
              - Namespace
    generate:
      apiVersion: networking.k8s.io/v1
      kind: NetworkPolicy
      name: allow-dns
      namespace: "{{request.object.metadata.name}}"
      synchronize: false
      data:
        spec:
          podSelector:
            matchLabels: {}
          policyTypes:
            - Egress
          egress:
            - to:
                - namespaceSelector:
                    matchLabels:
                      name: kube-system
          ports:
            - protocol: UDP
              port: 53
```





```
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: disallow-default-namespace
  annotations:
    policies.kyverno.io/description: >-
      Kubernetes Namespaces are an optional feature that provide a way to segment and isolate cluster resources across multiple applications and users. As a best practice, workloads should be isolated with Namespaces. Namespaces should be required and the default (empty) Namespace should not be used. This policy validates that Pods specify a Namespace name other than `default`. Rule auto-generation is disabled here due to Pod controllers need to specify the `namespace` field under the top-level `metadata` object and not at the Pod template level.
spec:
  validationFailureAction: audit
  background: true
  rules:
  - name: validate-namespace
    match:
      any:
        - resources:
            kinds:
              - Pod
    validate:
      message: "Using 'default' namespace is not allowed."
      pattern:
        metadata:
          namespace: "!default"
  - name: validate-podcontroller-namespace
    match:
      any:
        - resources:
            kinds:
              - DaemonSet
              - Deployment
              - Job
              - StatefulSet
    validate:
      message: "Using 'default' namespace is not allowed for pod controllers."
      pattern:
        metadata:
          namespace: "!default"
```





For more: <https://kyverno.io/policies/>



Validating Admission Policy

FEATURE STATE: Kubernetes v1.28 [beta]

This page provides an overview of Validating Admission Policy.

What about ...

What is Validating Admission Policy?

Validating admission policies offer a declarative, in-process alternative to validating admission webhooks.

Validating admission policies use the Common Expression Language (CEL) to declare the validation rules of a policy. Validation admission policies are highly configurable, enabling policy authors to define policies that can be parameterized and scoped to resources as needed by cluster administrators.

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

<https://kyverno.io/blog/2024/02/26/generating-kubernetes-validatingadmissionpolicies-from-kyverno-policies/>



<https://killercoda.com/koksay/course/kyverno>



Thanks!

Do you have any questions?

[linkedin.com/in/korayoksay](https://www.linkedin.com/in/korayoksay)

@korayoksay

koksay

koray@kubernatic.com



CREDITS: This presentation template was created by
Slidesgo, and includes icons by **Flaticon**, and infographics
& images by **Freepik**