

A photograph of a roller coaster's tracks silhouetted against a sunset sky. The tracks form a series of loops and curves, with a bright light source on the right side of the frame. The overall mood is dramatic and adventurous.

OPA Gatekeeper

Policy and Governance for Kubernetes

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

Sertac Ozercan, Gatekeeper maintainer (@sozercan)

Lachie Evenson, CNCF Ambassador

(@LachlanEvenson)

A customizable Kubernetes admission webhook that
helps enforce policies and strengthen governance

Motivations

- control what end-users can do on the cluster
- help ensure clusters are in conformance with company policies

How do we help ensure conformance without sacrificing agility and autonomy?

Real World

Agile Bank

- Building the greatest P2P money transfer app to-date
- Highly regulated industry
- Both developers and admins are unhappy



Admin

- Cannot keep up with requests for infrastructure changes
- Too much time spent keeping up with changes to governance rules
- Has to manually audit to find out-of-conformance resources
- Everyone keeps making the same mistakes
- Figuring out what group is responsible for a given resource is hard



Developer



Photo by [Chunlea Ju](#) on [Unsplash](#)

- Cannot make infrastructure changes
 - They know exactly what they want, but have no permissions to do it
 - They need to wait for someone else to make a change before they can keep working
- Hard to know if a change is conformant
 - Changes are proposed, rejected, updated, and re-proposed
 - Turnaround per proposal is at least a day

User Requirements

- Free up admins' time
 - Audit & enforcement are automated
 - Common best practices are enforced
 - All resources have a clear owner
- Unblock developers
 - Self-service no longer puts conformance at risk
 - Fail-fast means that developers get instant feedback on what needs to change with instructive error messages

Agile Bank's Governance Policies

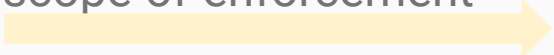
- All namespaces must have a label that lists a point-of-contact
- All pods must have an upper bound for resource usage
- All images must be from an approved repository
- Services must all have globally unique selectors
- Ensure all ingress hostnames are unique

Constraint Properties

- AND-ed together
 - Adding can only constrain, removing can only loosen
 - One rejection => whole request rejection
- Schema validation
 - Less error-prone

Constraints

Define
scope-of-enforcement



```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sRequiredLabels
metadata:
  name: all-must-have-owner
spec:
```

```
  match:
    kinds:
      - apiGroups: [""]
        kinds: ["Namespace"]
```

Describe intent



```
  parameters:
    message: "All namespaces must have an `owner` label
that points to your company username"
    labels:
      - key: owner
        allowedRegex: "^[a-zA-Z]+.agilebank.demo$"
```

Audit

- Periodically evaluates resources against constraints
- Allows for ongoing monitoring of cluster state to aid in detection and remediation of pre-existing misconfigurations
- Exposes results via the `status` field of the constraint

Audit Results

```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sRequiredLabels
metadata:
  name: all-must-have-owner
spec:
  match:
    kinds:
      - apiGroups:
          - ""
          kinds:
            - Namespace
  parameters:
    labels:
      - allowedRegex: ^[a-zA-Z]+.agilebank.demo$
        key: owner
        message: All namespaces must have an `owner` label that points to your company username
  status:
    auditTimestamp: "2019-05-11T01:46:13Z"
    totalViolations: 4
    enforced: true
  violations:
    - kind: Namespace
      message: All namespaces must have an `owner` label that points to your company username
      name: default
    - kind: Namespace
      message: All namespaces must have an `owner` label that points to your company username
      name: gatekeeper-system
    - kind: Namespace
      message: All namespaces must have an `owner` label that points to your company username
      name: kube-public
    - kind: Namespace
      message: All namespaces must have an `owner` label that points to your company username
      name: kube-system
```

Time of Audit



Violations



Dry Run

- Allows constraints to be tested in a running cluster without enforcing them
- Resources that are impacted by the dry run constraint are surfaced as violations in the `status` field of the constraint.

Audit Results

Enforcement Action

```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sRequiredLabels
metadata:
  name: ns-must-have-gk
spec:
```

```
  enforcementAction: dryrun
```

```
  match:
```

```
    kinds:
```

```
      - apiGroups: [""]
```

```
        kinds: ["Namespace"]
```

```
    parameters:
```

```
      labels: ["gatekeeper"]
```

```
status:
```

```
  auditTimestamp: "2019-08-15T01:46:13Z"
```

```
  enforced: true
```

```
  violations:
```

```
- enforcementAction: dryrun
```

```
  kind: Namespace
```

```
  message: 'you must provide labels: {"gatekeeper"}'
```

```
  name: default
```

```
- enforcementAction: dryrun
```

```
  kind: Namespace
```

```
  message: 'you must provide labels: {"gatekeeper"}'
```

```
  name: gatekeeper-system
```

Dry run violations

Handling uniqueness

Scenario: Enforce globally unique Ingress hostnames

- Some constraints are impossible to write without access to more state than just the object under test.
- By default, the audit will request each resource from the Kubernetes API during each cycle of the audit.

Data Replication

- Enabled via `audit-from-cache=true` flag
- Constraints that compare against other objects in the cluster
 - Require replication of existing objects in the cluster
- Audit queries
 - Require replication of objects to be audited for constraint violations
- Replication is defined via the config resource

```
apiVersion: config.gatekeeper.sh/v1alpha1
kind: Config
metadata:
  name: config
  namespace: gatekeeper-system
spec:
  sync:
    syncOnly:
      - kind: Service
        version: v1
      - kind: Pod
        version: v1
      - kind: Namespace
        version: v1
```

Constraint Templates

- Rego rule signature
 - If the rule matches, the constraint is violated
- Schema for Constraint Parameters


```
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: k8srequiredlabels
spec:
  crd:
    spec:
      names:
        kind: K8sRequiredLabels
      validation:
        # Schema for the `parameters` field
        openAPIV3Schema:
          properties:
            message:
              type: string
            labels:
              type: array
              items:
                type: object
                properties:
                  key:
                    type: string
                  allowedRegex:
                    type: string
```

```
targets:
- target: admission.k8s.gatekeeper.sh
  rego: |
```

```
package k8srequiredlabels

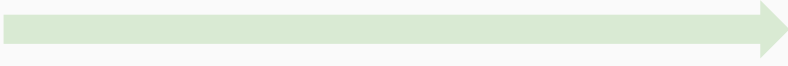
# helper libraries and additional tests (e.g. regex match) not shown

deny[{"msg": msg, "details": {"missing_labels": missing}}] {
  provided := {label | input.review.object.metadata.labels[label]}
  required := {label | label := input.constraint.spec.parameters.labels[_]}
  missing := required - provided
  count(missing) > 0
  def_msg := sprintf("you must provide labels: %v", [missing])
  msg := get_message(input.constraint, def_msg)
}
```

Schema for input parameters



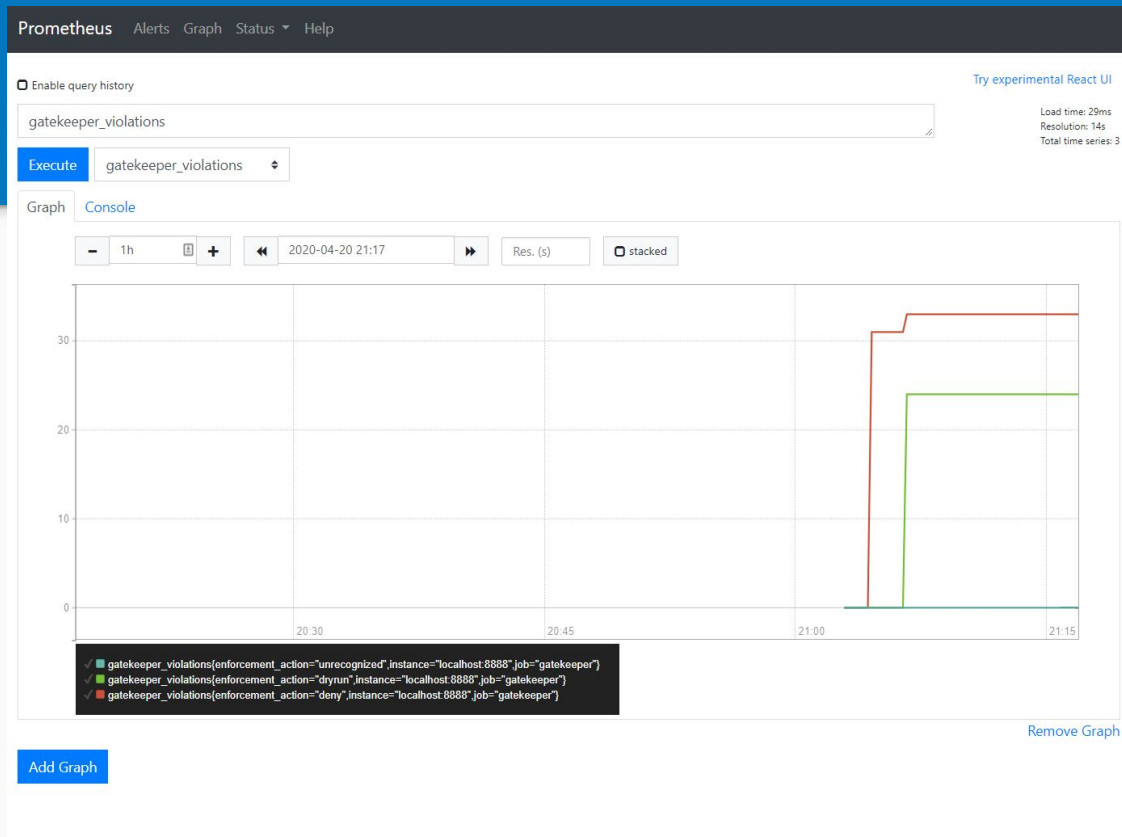
deny[{"msg": msg}]
rules are executed



Metrics

Supports Prometheus as backend for metrics:

- Violations per enforcement action
- Total number of constraint templates and constraints
- Last audit timestamp
- Audit duration
- And more...



Code Reuse!

- Gatekeeper provides the admission system
- Policies/Rules are implemented through constraints
- Constraints are parameterized and easily configurable by admins
- ConstraintTemplates provide the source code for constraints
 - Easily shared
 - Testable
 - Developed internally or sourced from the community
- Portable to other systems
 - e.g. CI/CD pipelines



Photo by [Judith Prins](#) on [Unsplash](#)

Demo

- <https://www.youtube.com/watch?v=UziULfVgQJ0>
- <https://www.youtube.com/watch?v=x81005Vf9Vo>

Project Status

- Beta
- Come help!
 - Issues
 - Feedback
 - User stories
 - Development



Photo by [Tikkho Maciel](#) on [Unsplash](#)

Cooking... but tasty

Getting started

- Policy library available here
 - <https://github.com/open-policy-agent/gatekeeper/tree/master/library/general>
- PodSecurityPolicy equivalents
 - <https://github.com/open-policy-agent/gatekeeper/tree/master/library/pod-security-policy>

Potential Growth

- Mutation
- External Data
- Authorization? (likely separate project, same general semantics)
- More audit features
- Developer tooling

Join Us!



Open Policy Agent

openpolicyagent.org

github.com/open-policy-agent/opa

OPA Gatekeeper

github.com/open-policy-agent/gatekeeper



Community

slack.openpolicyagent.org

[#kubernetes-policy](https://kubernetes.slack.com/#kubernetes-policy)

[Meetings](#) Wednesdays alternating between 9:00AM and 2:00PM PST

Questions?