

Embracing change: Policy-as-code in Kubernetes with Gatekeeper

Ara Pulido - Technical Evangelist
@arapulido



DATADOG



Datadog is a monitoring and analytics platform that helps companies improve observability of their infrastructure and applications



Policy?

Policy

Rules that governs the behavior of a software service



RBAC

Role Based Access Control

RBAC

Subject

**Kubernetes API
resources**

Verb

RBAC

Subject	Kubernetes API resources	Verb
user:ara	apiVersion:v1/core kind:Pod	create get watch



Policy

Authz

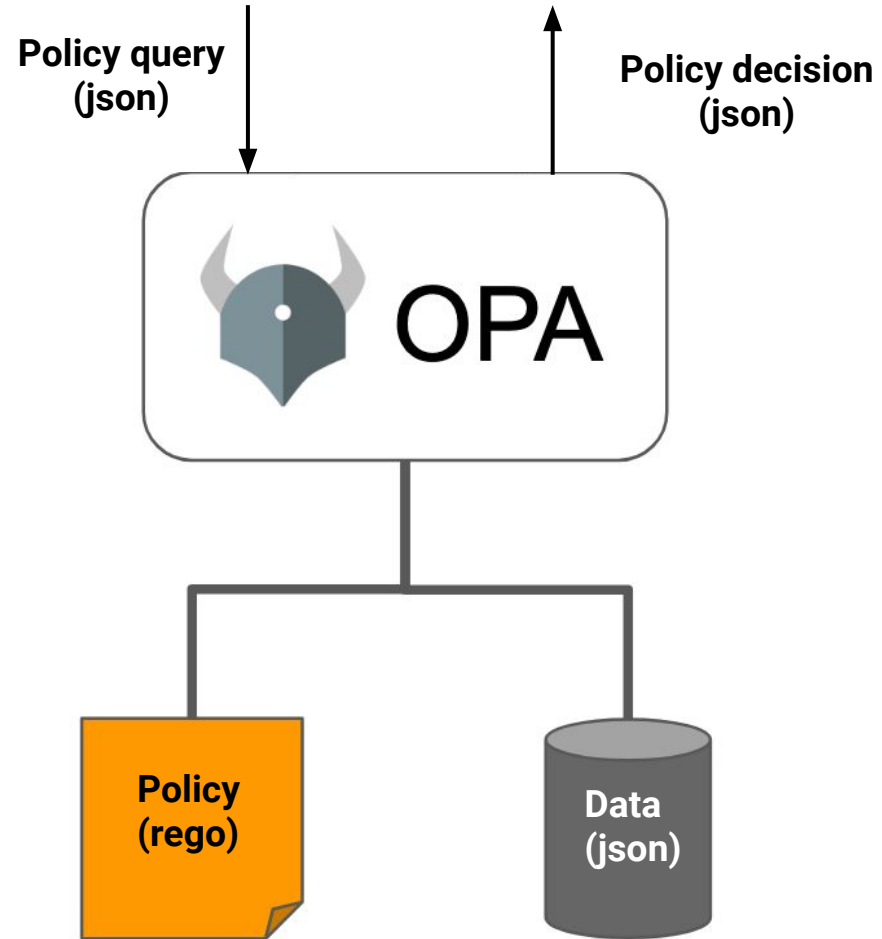
Can I run an image coming from a 3rd party registry?

Has my pod all the labels that are required by my organization?



Open Policy Agent

Domain agnostic



OPA Ecosystem

Showcase of OPA integrations, use-cases, and related projects.
Ordered by the amount of content.

Ac



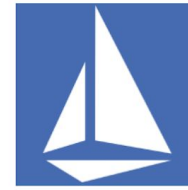
Kubernetes
Admission Control



Container Network
Authorization with
Envoy



Kafka Topic
Authorization



Container Network
Authorization with
Istio (at the Edge)



Custom Application
Authorization



Terraform
Authorization



Ceph Object Storage
Authorization



Scalr - Policy
enforcement for
Terraform



HTTP API
Authorization in PHP



Authorization for Java
Spring Security



Gloo API Gateway



Docker controls via
OPA Policies



Elasticsearch Data
Filtering



Spinnaker Pipeline
Policy Enforcement

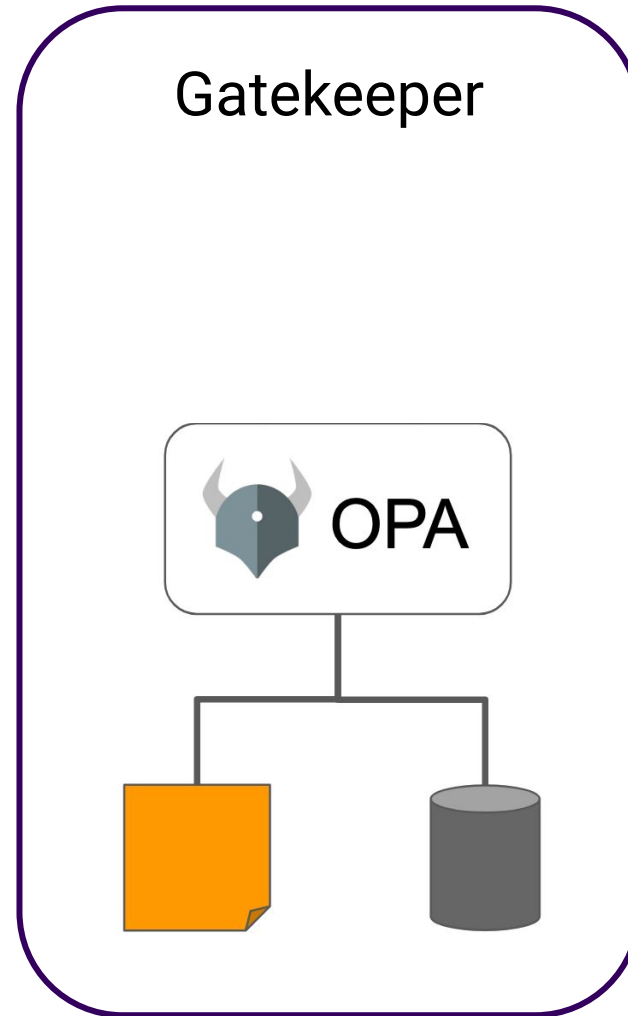


Forseti Security

GCP audit with Forseti



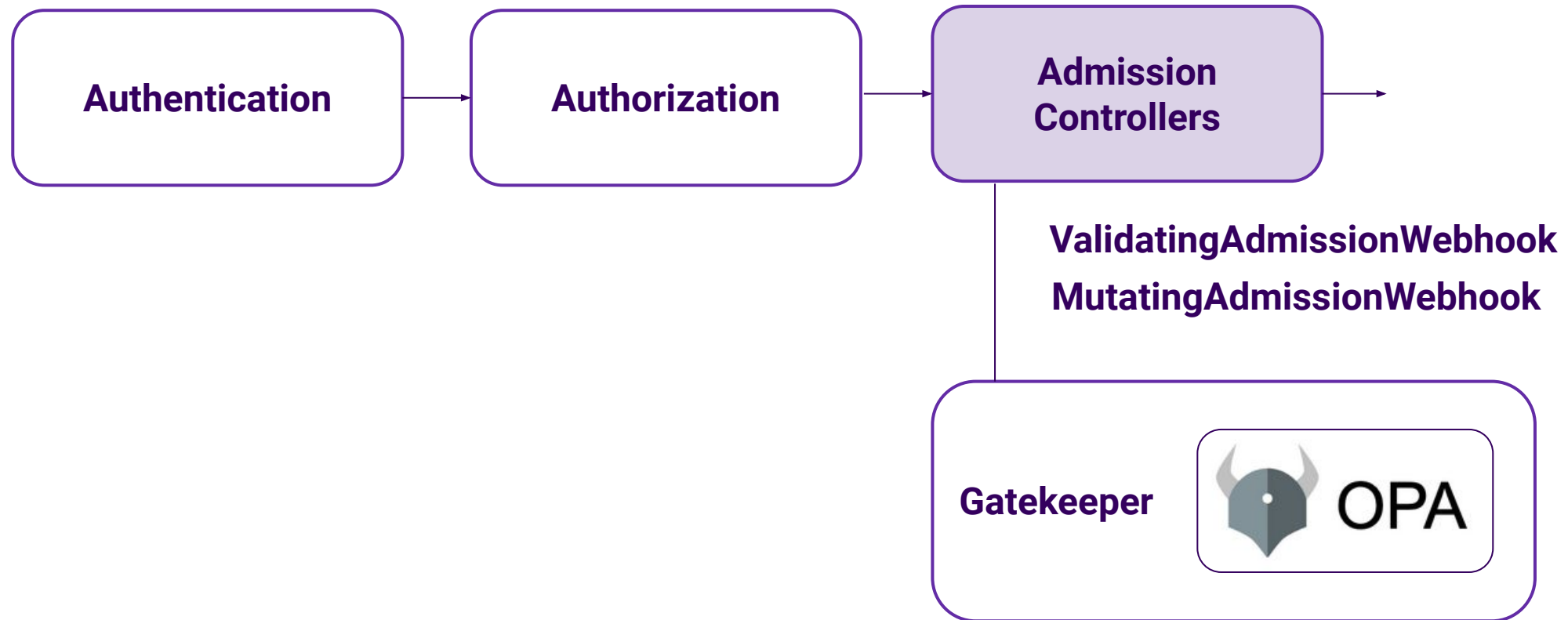
kubernetes



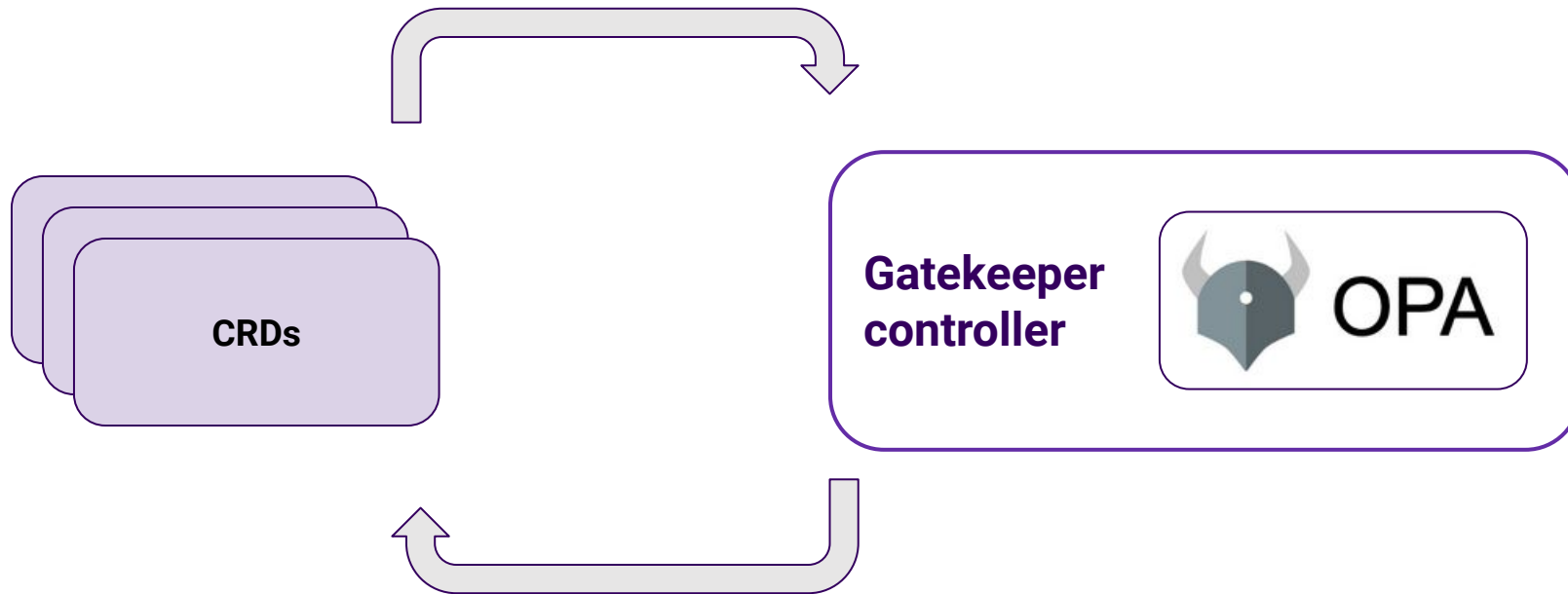


kubernetes

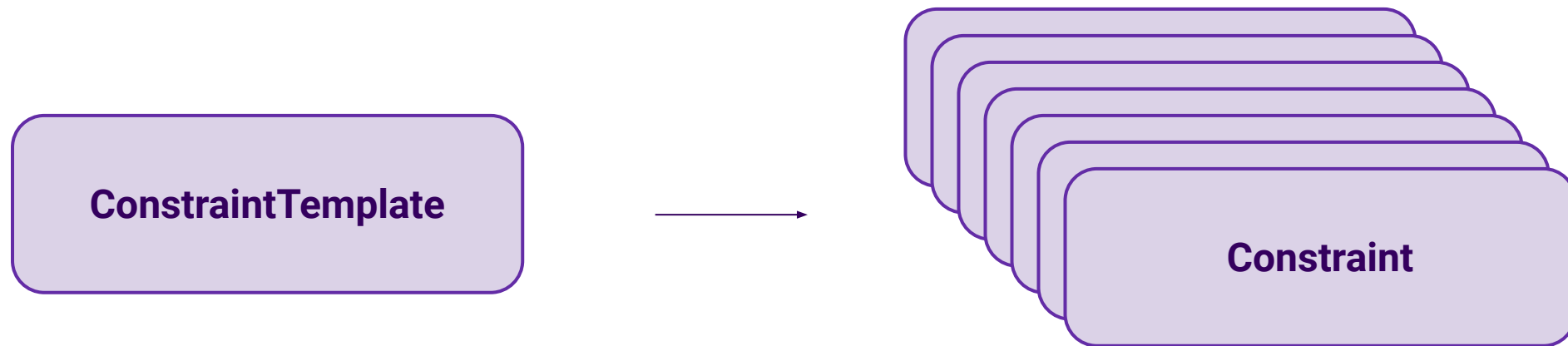
API request



Gatekeeper is Kubernetes native



Main CRDs



ConstraintTemplate

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

```
targets:
- target: admission.k8s.gatekeeper.sh
  rego: |
    package k8srequiredlabels

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

Constraints

```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sRequiredLabels
metadata:
  name: ns-must-have-gk
spec:
  match:
    kinds:
      - apiGroups: [""]
        kinds: ["Namespace"]
  parameters:
    labels: ["gatekeeper"]
```

```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sRequiredLabels
metadata:
  name: pod-required-labels
spec:
  match:
    namespace: "default"
    kinds:
      - apiGroups: [""]
        kinds: ["Pod"]
  parameters:
    labels: ["do-not-delete"]
```

Gatekeeper makes reuse of policy simple

Gatekeeper makes reuse of policy simple

Images can only come from approved registries

Gatekeeper makes reuse of policy simple

Images can only come from approved registries

Deployments require to have certain mandatory labels

Gatekeeper makes reuse of policy simple

Images can only come from approved registries

Deployments require to have certain mandatory labels

Container images must have a digest

Gatekeeper makes reuse of policy simple

Images can only come from approved registries

Deployments require to have certain mandatory labels


Container images must have a digest

Containers must have memory and CPU limits set and within a specified max value

Gatekeeper Library

master

gatekeeper-library / library / general /

 grosser and sozeran add testing (#33) ...

..

allowedrepos

block-nodeport-services

containerlimits

containerresourceratios

externalip

httpsonly

imagedigests

requiredlabels

requiredprobes

uniqueingresshost


uniqueserviceselector






kustomization.yaml

allow-privilege-escalation
apparmor
capabilities
flexvolume-drivers
forbidden-sysctls
fsgroup
host-filesystem
host-namespaces
host-network-ports
privileged-containers
proc-mount
read-only-root-filesystem
seccomp
selinux
users
volumes

https-only

🔑 master ▾ [gatekeeper-library](#) / [library](#) / [general](#) / [httpsonly](#) / [template.yaml](#)

 **grosner** add testing (#33) ... ✓

👤 5 contributors     

30 lines (28 sloc) | 1.03 KB

```
1  apiVersion: templates.gatekeeper.sh/v1beta1
2  kind: ConstraintTemplate
3  metadata:
4    name: k8shttpsonly
5    annotations:
6      description: Requires Ingress resources to be HTTPS only; TLS configuration should
7        be set and `kubernetes.io/ingress.allow-http` annotation equals false.
8  spec:
9    crd:
10     spec:
11       names:
12         kind: K8sHttpsOnly
13     targets:
14       - target: admission.k8s.gatekeeper.sh
15         rego: |
16           package k8shttpsonly
17
18           violation[{"msg": msg}] {
19             input.review.object.kind == "Ingress"
20             re_match("^(extensions|networking.k8s.io)/", input.review.object.apiVersion)
21             ingress := input.review.object
22             not https_complete(ingress)
23             msg := sprintf("Ingress should be https. tls configuration and allow-http=false anno
24           }
25
26           https_complete(ingress) = true {
27             ingress.spec["tls"]
28             count(ingress.spec.tls) > 0
29             ingress.metadata.annotations["kubernetes.io/ingress.allow-http"] == "false"
30           }
```

https-only

```
master gatekeeper-library / library / general / httpsonly / template.yaml

grosser add testing (#33) ... ✓

5 contributors

30 lines (28 sloc) 1.03 KB

1  apiVersion: templates.gatekeeper.sh/v1beta1
2  kind: ConstraintTemplate
3  metadata:
4    name: k8shttpsonly
5    annotations:
6      description: Requires Ingress resources to be HTTPS only; TLS configuration should
7        be set and `kubernetes.io/ingress.allow-http` annotation equals false.
8  spec:
9    crd:
10     spec:
11       names:
12         kind: K8sHttpsOnly
13     targets:
14       - target: admission.k8s.gatekeeper.sh
15         rego: |
16           package k8shttpsonly
17
18           violation[{"msg": msg}] {
19             input.review.object.kind == "Ingress"
20             re_match("^(extensions|networking.k8s.io)/", input.review.object.apiVersion)
21             ingress := input.review.object
22             not https_complete(ingress)
23             msg := sprintf("Ingress should be https. tls configuration and allow-http=false anno
24           }
25
26           https_complete(ingress) = true {
27             ingress.spec["tls"]
28             count(ingress.spec.tls) > 0
29             ingress.metadata.annotations["kubernetes.io/ingress.allow-http"] == "false"
30           }
```

```
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sHttpsOnly
metadata:
  name: ingress-https-only
spec:
  match:
    kinds:
      - apiGroups: ["extensions", "networking.k8s.io"]
        kinds: ["Ingress"]
```

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-demo-disallowed
spec:
  rules:
    - host: example-host.example.com
      http:
        paths:
          - backend:
              serviceName: nginx
              servicePort: 80
```

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



Observability

Out-of-the-box metrics

ConstraintTemplates, Constraints (gauge)

Webhook: #requests (count) and latency (histogram)

Audit: #violations (count) and time last run (gauge)

Sync: # resources cached (count) and time last run (gauge)

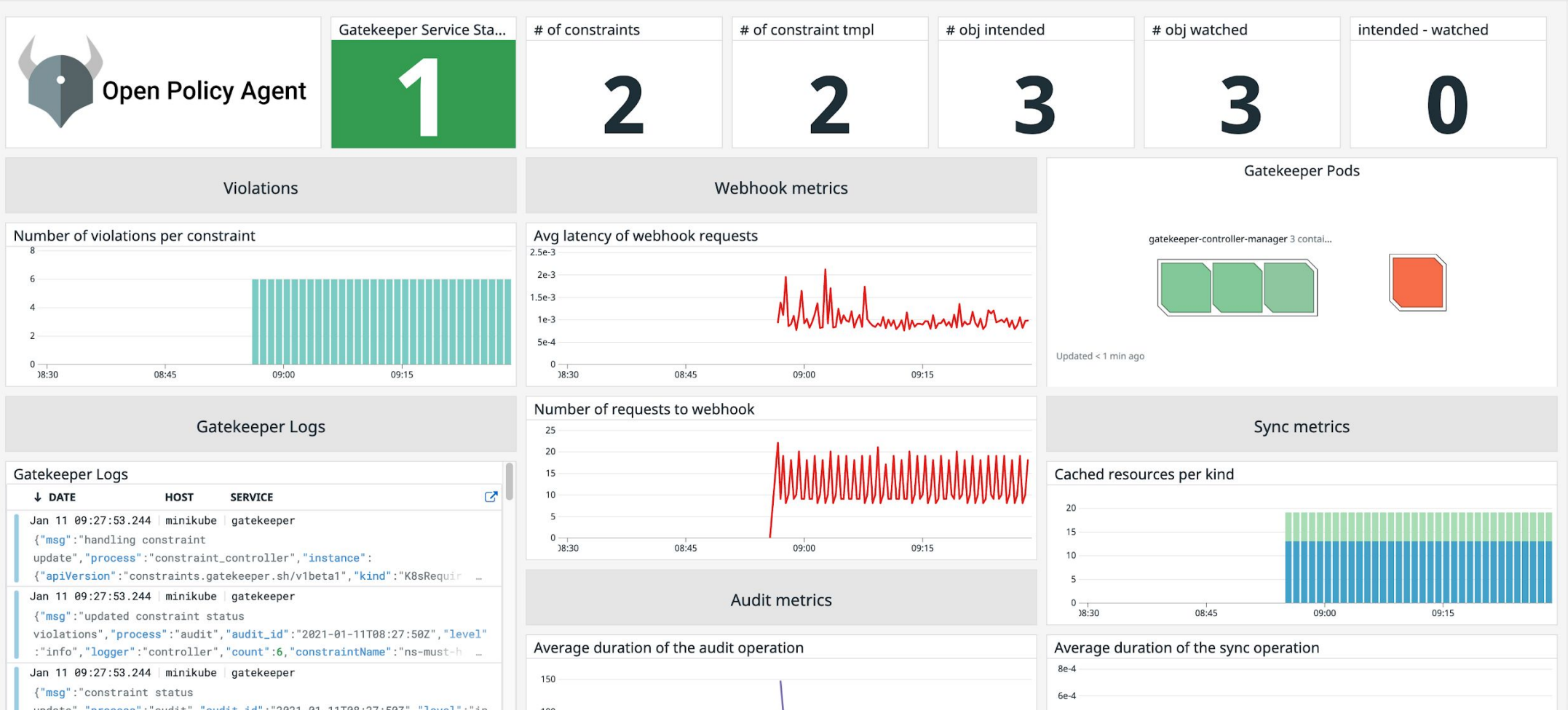
Datadog integration

★ Gatekeeper Overview

+ Edit Widgets

1h Past 1 Hour

Add Template Variables





Demo



DATADOG

datadoghq.com/careers

@arapulido