

POLICY BASED CONTROL

FOR CLOUD NATIVE ENVIRONMENT

WITH OPA



Docker DevTools Day 3.0
Aug 24, 2024

- ▶ Kunal Das
- ▶ Sr DevOps Engineer @ Cyncly
- ▶ HashiCorp Certified Terraform Associate
- ▶ Microsoft Certified DevOps Expert



POLICY ENFORCEMENT CHALLENGES

Traditional



Firewall



Access Control



Manual Approval

Cloud-Native



Granular checks



Dynamic Control



Automated Approval

OPA AND GATEKEEPER

Open Policy Agent (OPA)

- General-purpose policy engine
- Evaluates policies written in Rego language
- Can be used across various systems and platforms

Gatekeeper

- Kubernetes-native policy management
- Implements OPA as an admission controller
- Extends Kubernetes API with custom resources



Gatekeeper



OPA

Custom Resource Definitions (CRDs)

ConstraintTemplate

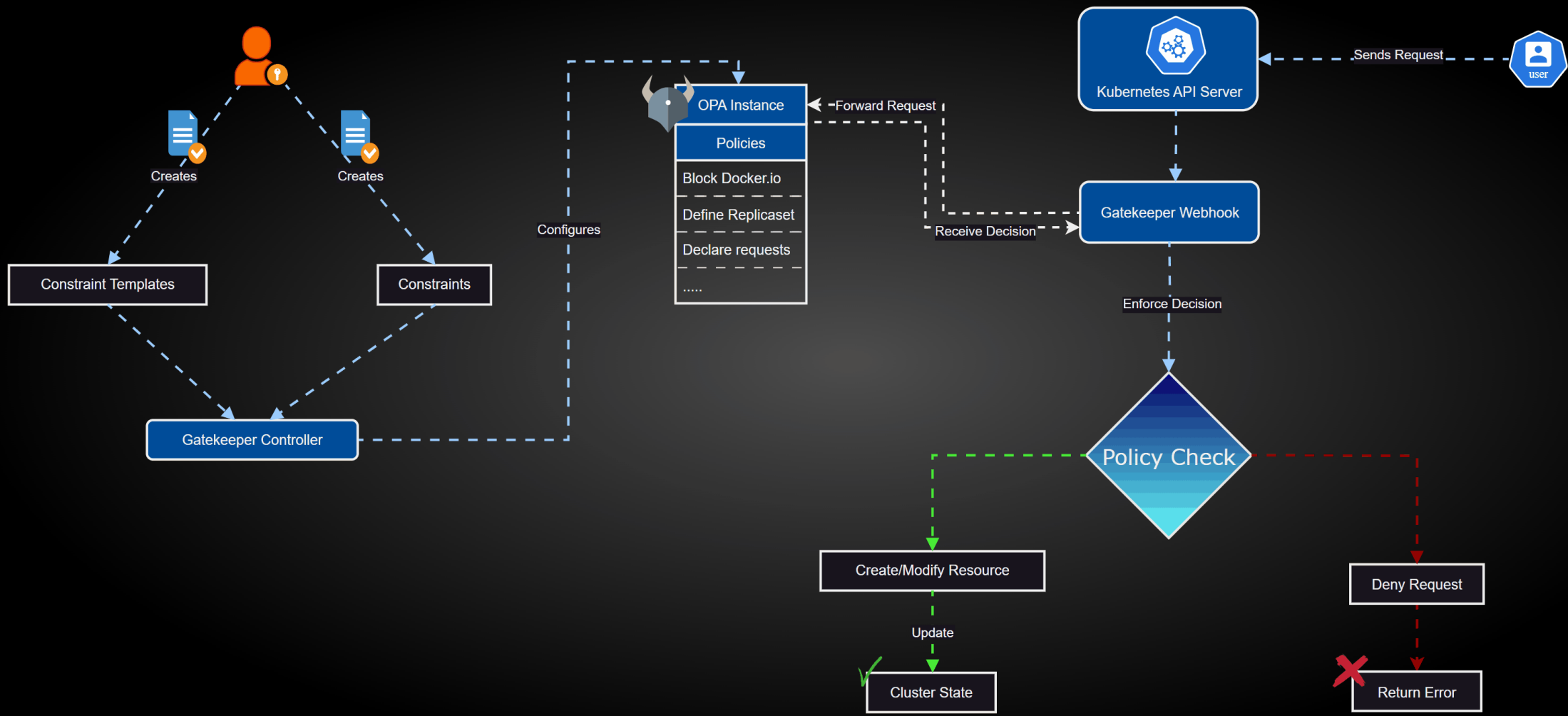
- Defines the policy logic using Rego
- Creates a new custom resource type

Constraint

- Instance of a ConstraintTemplate
- Specifies which resources the policy applies to

```
1 apiVersion: constraints.gatekeeper.sh/v1beta1
2 kind: K8sRequiredLabels
3 metadata:
4   name: ns-must-have-gk
5 spec:
6   match:
7     kinds:
8       - apiGroups: [""]
9         kinds: ["Namespace"]
10   excludedNamespaces:
11     - kube-system
12     - kube-public
13     - kube-node-lease
14     - default
15     - gatekeeper-system
16     - kured
17   parameters:
18     labels: ["createdby"]
```

```
1 apiVersion: templates.gatekeeper.sh/v1beta1
2 kind: ConstraintTemplate
3 metadata:
4   name: k8srequiredlabels
5 spec:
6   crd:
7     spec:
8       names:
9         kind: K8sRequiredLabels
10      validation:
11        openAPIV3Schema:
12          properties:
13            labels:
14              type: array
15              items: string
16      targets:
17        - target: admission.k8s.gatekeeper.sh
18          rego: |
19            package k8srequiredlabels
20            violation[{"msg": msg, "details": {"missing_labels": missing}}] {
21              provided := {label | input.review.object.metadata.labels[label]}
22              required := {label | label := input.parameters.labels[_]}
23              missing := required - provided
24              count(missing) > 0
25              msg := sprintf("%v is missing from your definition 🙄🙄", [missing])
26            }
```



USAGE SCENARIOS

Resource Constraints

- Enforcing CPU/memory limits on pods
- Restricting storage class usage

Security Enforcement

- Ensuring pods run as non-root
- Enforcing network policies

Compliance Requirements

- Mandatory labeling of resources
- Restricting use of latest image tags

Cost Optimization

- Enforcing resource quotas
- Limiting expensive cloud services usage

Multi-tenancy

- Namespace-based restrictions
- Enforcing isolation between tenants

Best Practices

- Enforcing liveness and readiness probes
- Ensuring high availability configurations

DEMO

```
1 apiVersion: constraints.gatekeeper.sh/v1beta1
2 kind: K8sRequiredLabels
3 metadata:
4   name: ns-must-have-gk
5 spec:
6   match:
7     kinds:
8       - apiGroups: [""]
9         kinds: ["Namespace"]
10   excludedNamespaces:
11     - kube-system
12     - kube-public
13     - kube-node-lease
14     - default
15     - gatekeeper-system
16     - kured
17 parameters:
18   labels: ["createdby"]
```

1

```
1 apiVersion: v1
2 kind: Namespace
3 metadata:
4   name: mynamespace
5 labels:
6   createdby: kunal
```

4

```
1 apiVersion: v1
2 kind: Namespace
3 metadata:
4   name: badns
```

3

```
1 apiVersion: templates.gatekeeper.sh/v1beta1
2 kind: ConstraintTemplate
3 metadata:
4   name: k8srequiredlabels
5 spec:
6   crd:
7     spec:
8       names:
9         kind: K8sRequiredLabels
10      validation:
11        openAPIV3Schema:
12          properties:
13            labels:
14              type: array
15              items: string
16 targets:
17   - target: admission.k8s.gatekeeper.sh
18     rego: |
19       package k8srequiredlabels
20       violation[{"msg": msg, "details": {"missing_labels": missing}}] {
21         provided := {label | input.review.object.metadata.labels[label]}
22         required := {label | label := input.parameters.labels[_]}
23         missing := required - provided
24         count(missing) > 0
25         msg := sprintf("\n%v is missing from your defination 🙄🙄", [missing])
26       }
```

2



Dd1

CHALLENGES

Learning Curve

- Policies are written on Rego.

Performance Impact

- Potential latency in admission requests
- Resource consumption in large-scale deployments

Policy Management

- Maintaining consistency across multiple clusters
- Version control and change management of policies.

Integration Complexity

- Incorporating with existing CI/CD pipelines
- Aligning with other security tools and practices

REFERENCES

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

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

<https://kubernetes.io/blog/2019/08/06/opa-gatekeeper-policy-and-governance-for-kubernetes/>

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

Thank You

Reach out to me in case any doubts,
suggestions or feedback !!

