Open Policy Agent (OPA)

Introduction - KubeCon Shanghai 2019
Torin Sandall
Engineer at Styra
Co-creator of OPA

@tsandall on OPA slack
@sometorin

openpolicyagent.org
# OPA: Community

## Inception
Project started in 2016 at Styra.

## Goal
Unify policy enforcement across the stack.

## Users
- Netflix
- Chef
- Medallia
- Cloudflare
- State Street
- Pinterest
- Intuit
- Capital One
- ...and many more.

## Use Cases
- Admission control
- Authorization
- ACLs
- RBAC
- IAM
- ABAC
- Risk management
- Data Protection
- Data Filtering

## Today
- CNCF project (Incubating)
- 60+ contributors
- 900+ slack members
- 2,000+ stars
- 20+ integrations
Example: Application

- Bob (customer)
- Alice (developer)

Diagram:
- Portal
- Payments
- Accounts
- Promotions
- Notifications
- S3
- SQL
- Partner APIs

(openpolicyagent.org)
Example: Application

bob (customer)

alice (developer)

payments
accounts
promotions
notifications
S3
SQL
Partner APIs

openpolicyagent.org
Example: Application

- bob (customer)
  - portal
    - payments
      - S3
    - accounts
    - promotions
    - notifications
  - authz
- alice (developer)
  - authz
  - Partner APIs
  - SQL

openpolicyagent.org
Example: Application

Obvious questions...

- How do you enforce new policies from infosec, compliance, or legal?
- How do you delegate control to your end-users?
- How do you roll-out policy changes?
- How do you leverage context, e.g., HR/User DB?
- How do you render UIs based on policy?
- How do you test your policies for correctness?
- What about 100+ services written in Java, Ruby, ...
Example: Kubernetes Platform

amanda (admin)

alice (developer)
Example: Kubernetes Platform

amanda (admin)

alice (developer)
Example: Kubernetes Platform

CONFLICTING INGRESS HOSTNAMES

VULNERABLE CONTAINER IMAGES

ALLOW ALL EGRESS TRAFFIC

UNLIMITED CPU AND MEMORY

amanda (admin)

alice (developer)
OPA: Unified Policy Enforcement Across the Stack

- Admission Control
- Container Execution, SSH, sudo
- Data Protection & Data Filtering

Microservice APIs

Risk Management

Forseti Security

openpolicyagent.org
OPA: General-purpose Policy Engine

Enforcement is decoupled from decision-making.

```
{  
  "method": "GET",
  "path": ["salary", "bob"],
  "user": "alice"
}
```

GET /salary/bob HTTP/1.1
Authorization: alice

true or false

OPA: General-purpose Policy Engine

Enforcement is decoupled from decision-making.
OPA: General-purpose Policy Engine

Enforcement is decoupled from decision-making.

Input can be **ANY** JSON value

Output can be **ANY** JSON value
OPA: Features

- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?
OPA: Features

- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?

- **Library (Go), sidecar/host-level daemon**
  - Policy and data are kept in-memory
  - Zero decision-time dependencies
OPA: Features

- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?

- **Library (Go), sidecar/host-level daemon**
  - Policy and data are kept in-memory
  - Zero decision-time dependencies

- **Management APIs for control & observability**
  - Bundle service API for sending policy & data to OPA
  - Status service API for receiving status from OPA
  - Log service API for receiving audit log from OPA
OPA: Features

- **Declarative Policy Language (Rego)**
  - Can user X do operation Y on resource Z?
  - What invariants does workload W violate?
  - Which records should bob be allowed to see?

- **Library (Go), sidecar/host-level daemon**
  - Policy and data are kept in-memory
  - Zero decision-time dependencies

- **Management APIs for control & observability**
  - Bundle service API for sending policy & data to OPA
  - Status service API for receiving status from OPA
  - Log service API for receiving audit log from OPA

- **Tooling to build, test, and debug policy**
  - opa run, opa test, opa fmt, opa deps, opa check, etc.
  - VS Code plugin, Tracing, Profiling, etc.

openpolicyagent.org
Example policy

"Employees can read their own salary and the salary of anyone they manage."
## OPA: Integrations

<table>
<thead>
<tr>
<th>Integration</th>
<th>Feature</th>
<th>Policy</th>
</tr>
</thead>
</table>
| **Admission Control** | "Restrict ingress hostnames for payments team."
| | "Ensure container images come from corporate repo."
| **API Authorization** | "Deny test scripts access to production services."
| | "Allow analysts to access APIs serving anonymized data."
| **SSH & sudo** | "Only allow on-call engineers to SSH into production servers."
| **Data Protection** | "Trades exceeding $10M must be executed between 9AM and 5PM and require MFA."
| **Data Filtering** | "Users can access files for past 6 months related to the region they licensed."

*Icons for: kubernetes, Terraform, docker, envoy, Istio, Kong, spring, Linux PAM, ceph, kafka, MINIO, SQLite, elastic*
Use Case: Authorization

Uses OPA to enforce **access control** in microservices across a **variety of languages and frameworks** for **thousands of instances** in their cloud infrastructure. Netflix takes advantage of OPA's ability to bring in **contextual information** and data from remote resources in order to evaluate policies in a **flexible and consistent manner**. For a description of how Netflix has architected access control with OPA check out [this talk from KubeCon Austin 2017](https://openpolicyagent.org).
Use Case: API Authorization

Integrates OPA to implement **IAM-style access control** and enumerate **user->resource permissions** in Chef Automate V2. The integration utilizes OPA's Partial Evaluation feature to reduce evaluation time (in exchange for higher update latency.)
Use Case: k8s Admission Control

Uses OPA as a validating and mutating admission controller to implement various security, multi-tenancy, and risk management policies across approximately 50 clusters and 1,000 namespaces. For more information on how Intuit uses OPA see this talk from KubeCon Seattle 2018.
Thank You! Questions?

**Tuesday, June 25 • 18:15 - 18:50**

- Gatekeeper: Flexible, Shareable Policy for Kubernetes - Craig Peters, Microsoft

- Deep Dive: Kubernetes Policy WG - Zhipeng Huang, Huawei

- [open-policy-agent/opa](https://open-policy-agent.org)

- [slack.openpolicyagent.org](https://slack.openpolicyagent.org)
How Does OPA Work?

Example policy

"Employees can read their own salary and the salary of anyone they manage."
Example policy

Employees can read their own salary and the salary of anyone they manage.
OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "bob"
OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "bob"

allow = true {
  input.method = "GET"
  input.path = ["salary", employee_id]
  input.user = employee_id
}

openpolicyagent.org
OPA: Declarative Language (Rego)

**Example policy**

Employees can read their own salary and the salary of anyone they manage.

**Input Data**

- method: "GET"
- path: ["salary", "bob"]
- user: "bob"

allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = "bob"
}
OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "alice"

Different user now!
OPA: Declarative Language (Rego)

**Example policy**

Employees can read their own salary and the salary of anyone they manage.

**Input Data**

- method: "GET"
- path: ["salary", "bob"]
- user: "alice"

allow = true {
  input.method = "GET"
  input.path = ["salary", "bob"]
  input.user = "bob"
}

This statement will "FAIL"

Different user now!
OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "alice"

Context Data

{"managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
}}
OPA: Declarative Language (Rego)

Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "alice"

Context Data

{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "alice"

Context Data

{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
Example policy

Employees can read their own salary and the salary of anyone they manage.

Input Data

method: "GET"
path: ["salary", "bob"]
user: "alice"

Context Data

{
  "managers": {
    "bob": ["alice", "fred"]
    "alice": ["fred"]
  }
}
import data.managers

allow = true {
    input.method = "GET"
    input.path = ["salary", employee_id]
    input.user = employee_id
}

allow = true {
    input.method = "GET"
    input.path = ["salary", "bob"]
    input.user = "alice"
}