Enforcing Bespoke Policies in Kubernetes

Torin Sandall

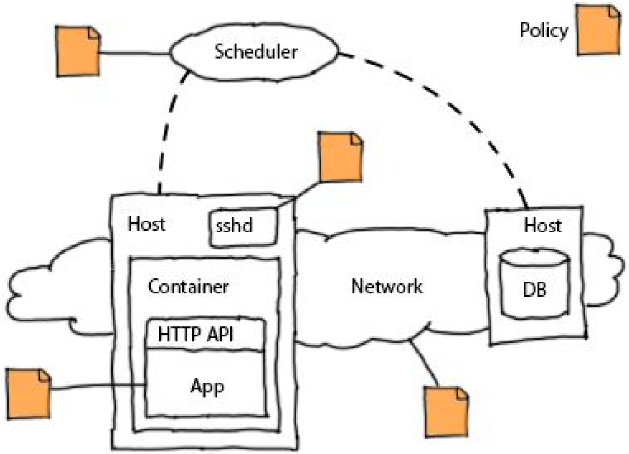
@sometorin

openpolicyagent.org

Overview

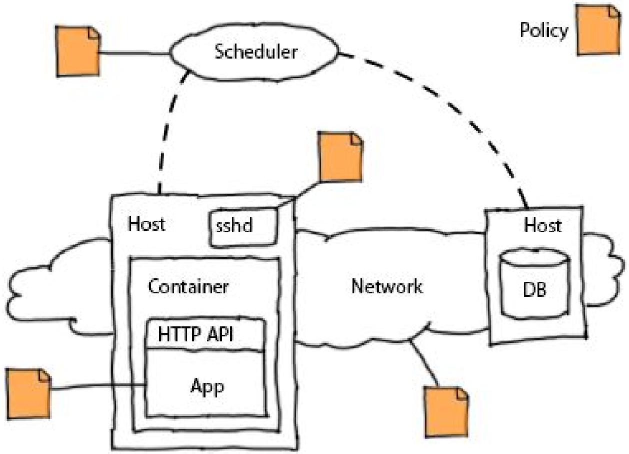
- Background: What Is Policy?
- Example Scenario
- Admission Control
- Open Policy Agent

What Is Policy?



What Is Policy?

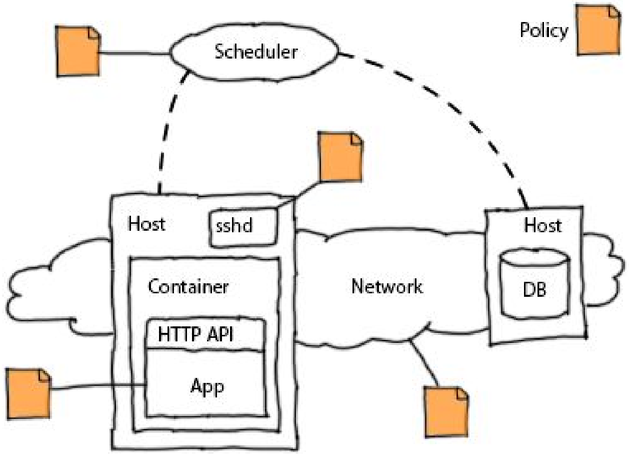
- Policies are vital to every organization
  - Policies are required across the stack
  - Policies are organization-specific
  - Policies change frequently over time



What Is Policy?

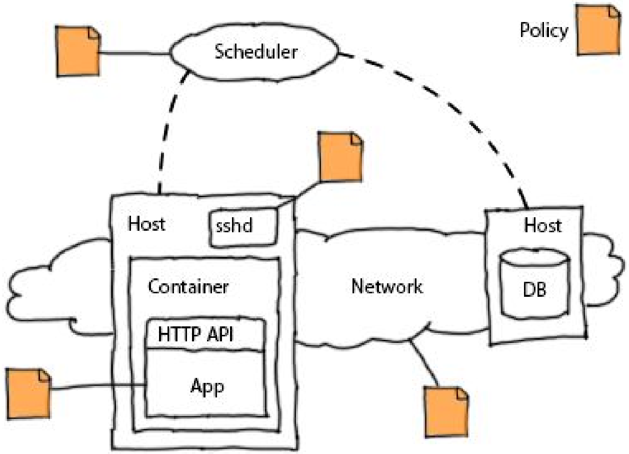
- Policies are vital to every organization
  - Policies are required across the stack
  - Policies are organization-specific
  - Policies change frequently over time

- Policy enforcement methods vary wildly
  - Weak guarantees from tribal knowledge & wikis
  - High cost from hard-coded policy decisions



What Is Policy?

- Policies are vital to every organization
  - Policies are required across the stack
  - Policies are organization-specific
  - Policies change frequently over time
- Policy enforcement methods vary wildly
  - Weak guarantees from tribal knowledge & wikis
  - High cost from hard-coded policy decisions
- Existing solutions lack expressiveness
  - Logic and data
  - Decisions
  - Composition

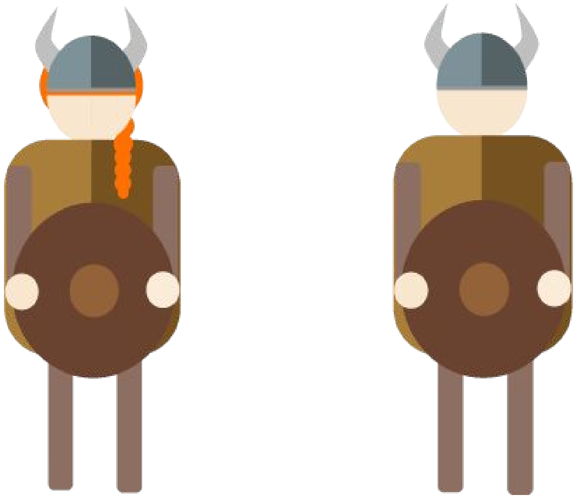


@sometorin

openpolicyagent.org

Example Scenario

- **Alice** and **Bob** work for AcmeCorp



Alice
Platform Engineer

Bob
App Engineer

Example Scenario

- **Alice** and **Bob** work for AcmeCorp

- Bob needs shell access to containers running on Kubernetes

Alice
Platform Engineer

Bob
App Engineer

Example Scenario

- **Alice** and **Bob** work for AcmeCorp

- Bob needs shell access to containers running on Kubernetes

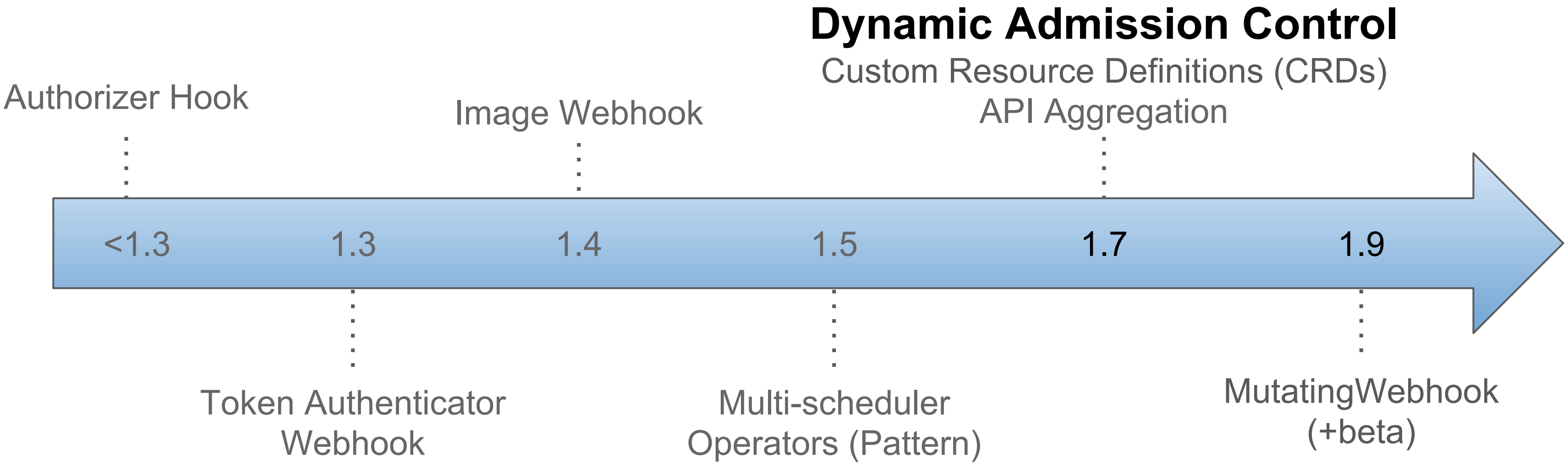
- Bob cannot be trusted with access to **privileged containers** running in the **production namespace**

Alice
Platform Engineer

Bob
App Engineer

openpolicyagent.org

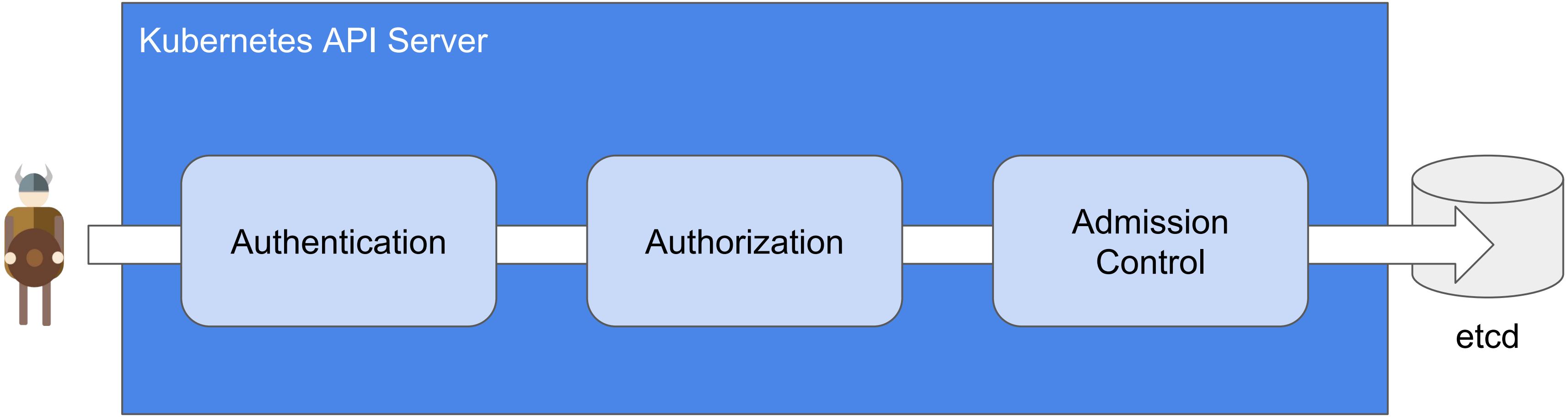
Kubernetes Extensibility



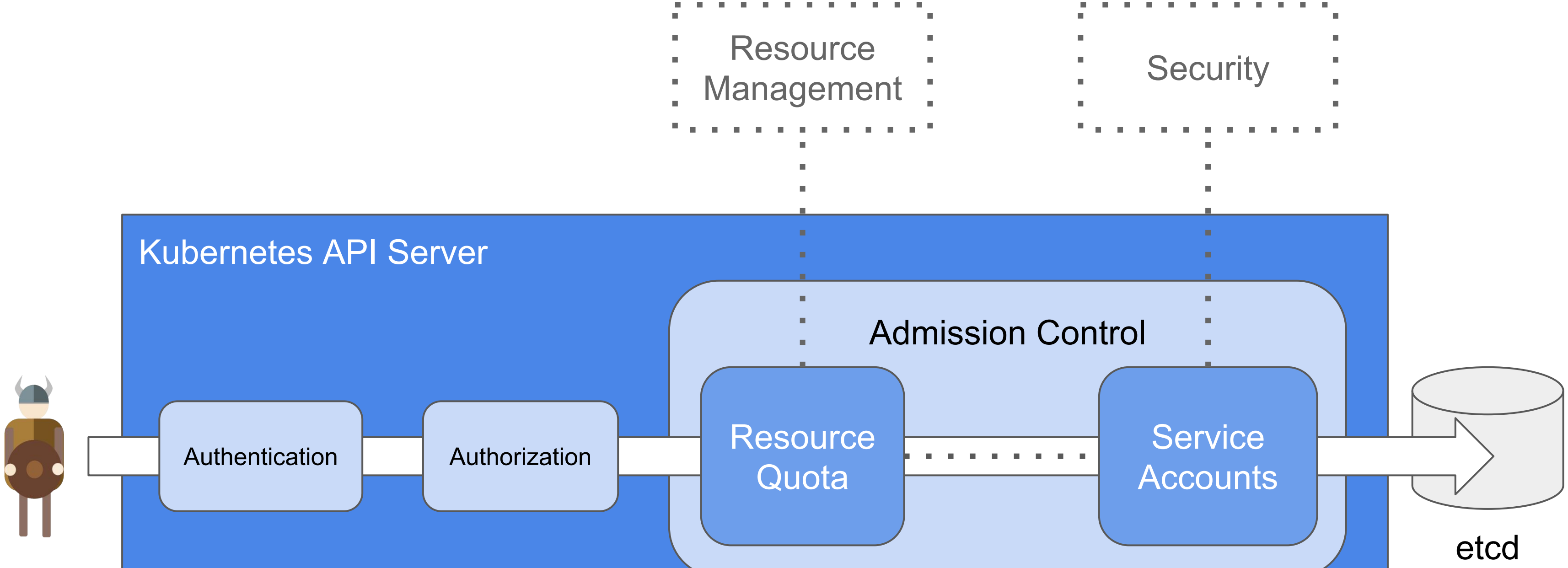
@sometorin

openpolicyagent.org

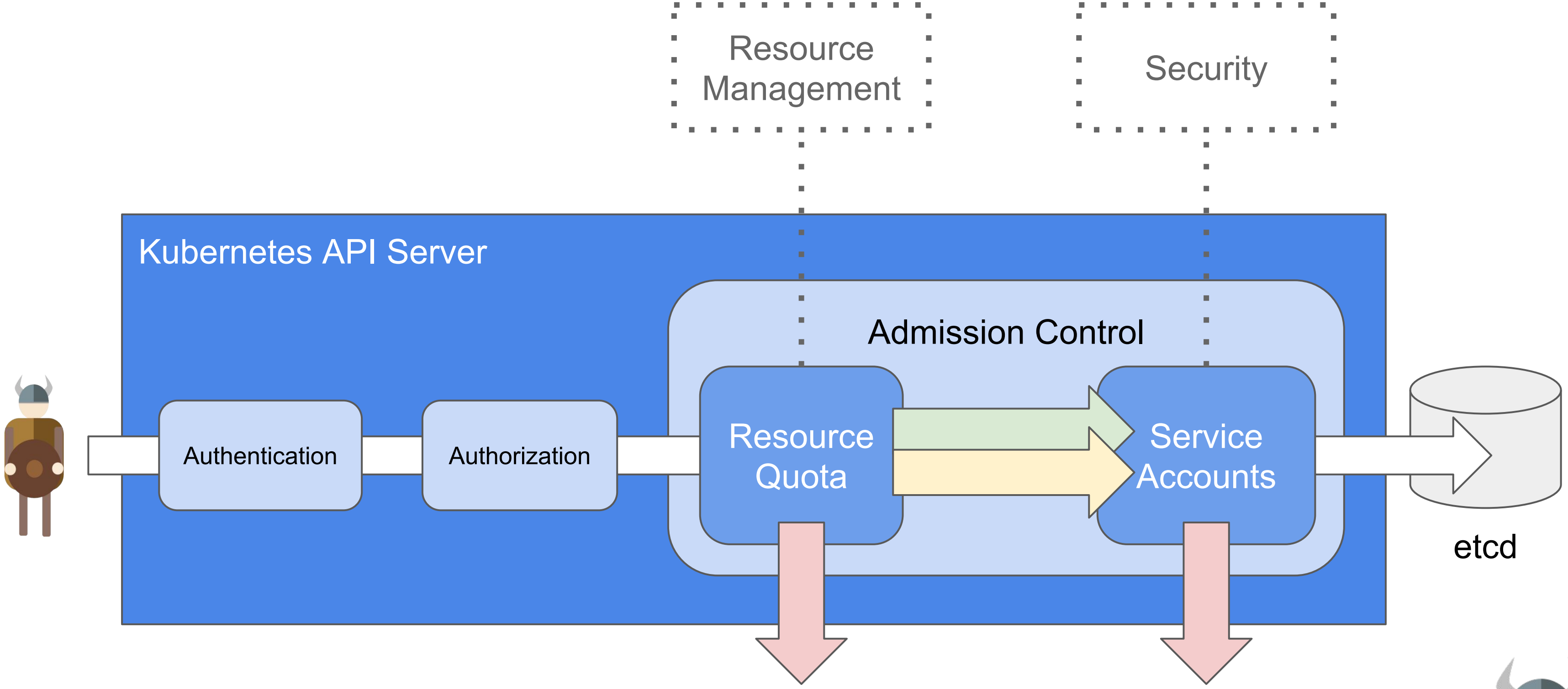
Admission Control



Admission Control



Admission Control



@sometorin

openpolicyagent.org

Admission Control: Before 1.7

- Static compilation & configuration
  - 30+ admission controllers
  - 1-4 added per release
  - Command line arguments
  - Static configuration files

admit
deny
exec
limitranger
namespace
resourcequota
securitycontext
serviceaccount
initialresources
alwayspullimages
antiaffinity
persistentvolume
security
imagepolicy
storageclass
gc
podnodeselector
defaulttolerationseconds
podpreset
initialization
noderestriction
podtolerationrestriction
schedulingpolicy
image/imagelimitrangerplugin
image/imagepolicyplugin
ingress/ingress
project/lifecycle
project/podnodeenvironment
project/projectrequestlimit
quota/quotaclusterresourceoverride
quota/clusterquota
quota/runonceduration
scheduler/podnodeconstraints
security/constraint

Admission Control: Before 1.7

- Static compilation & configuration
  - 30+ admission controllers
  - 1-4 added per release
  - Command line arguments
  - Static configuration files

- Example Scenario
  - Alice forks Kubernetes into a private repository
  - Alice implements the policy inside the plugin framework
  - Alice now has to build, push, and upgrade Kubernetes itself

admit
deny
exec
limitranger
namespace
resourcequota
securitycontext
serviceaccount
initialresources
alwayspullimages
antiaffinity
persistentvolume
security
imagepolicy
storageclass
gc
podnodeselector
defaulttolerationseconds
podpreset
initialization
noderestriction
podtolerationrestriction
schedulingpolicy
image/imagelimitrangerplugin
image/imagepolicyplugin
ingress/ingress
project/lifecycle

bobprotectionpolicy

project/podnodeenvironment
project/projectrequestlimit
quota/quotaclusterresourceoverride
quota/clusterquota
quota/runonceduration
scheduler/podnodeconstraints
security/constraint

@sometorin

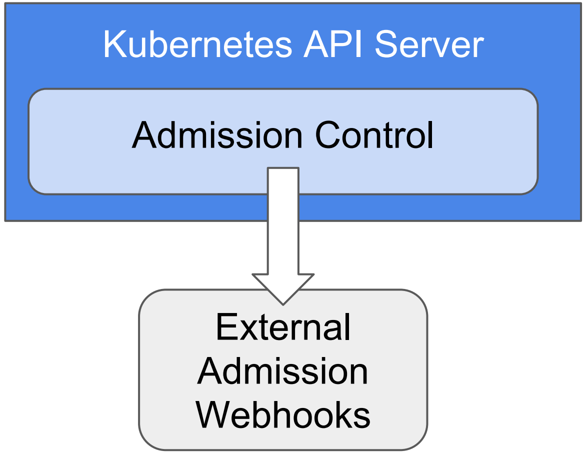
openpolicyagent.org

Admission Control: Webhooks

- Admission controllers can be implemented as webhooks that run on top of Kubernetes

- Webhooks can **allow** or **deny** incoming requests
  - Before etcd is updated
  - Before clients are notified

- Webhooks are configured **dynamically** via Kubernetes APIs



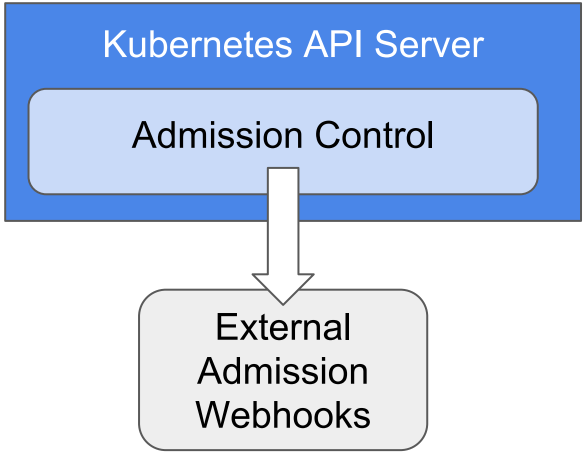
Admission Control: Webhooks

- The API Server calls webhooks whose configuration rules match the incoming request:

```
match [
  {operations: [“create”], kinds: [“pods”]},
  {operations: [“delete”], kinds: [“services”]}
]
```

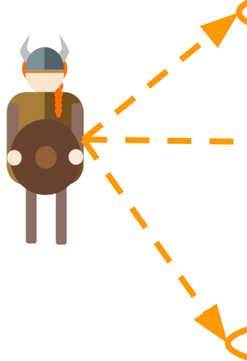
- Rules can include wildcards:

```
match [
  {operations: [“*”], kinds: [“*”]}
]
```

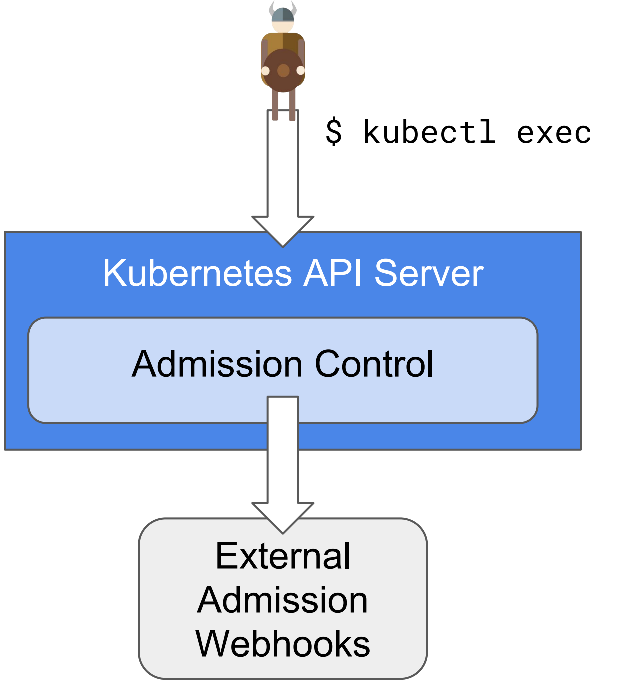


Admission Control: Webhooks

- The API Server provides the **operation, entire object, and user info** in the webhook call



```
kind: AdmissionReview
spec:
  kind: {kind: Pod, version: v1}
  name: admission-webhook-demo-373699553-8srx8
  namespace: default
  object:
    Options:
      Command: [sh]
      Container: admission-webhook-demo
      ...
    ResourcePath: pods/exec
  operation: CONNECT
  userInfo:
    groups: ['system:masters', 'system:authenticated']
    username: minikube
```



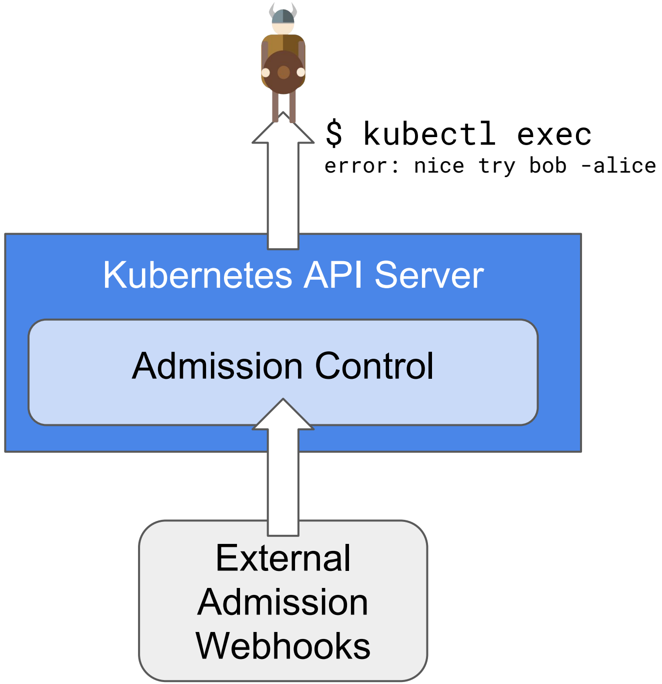
```
$ kubectl exec
```

Admission Control: Webhooks

- Webhooks respond with an **AdmissionReview** that indicates whether to **allow** or **deny** the request

```
kind: AdmissionReview
status:
  allowed: false
  reason:
    message: “nice try bob -alice”
```

- The API Server rejects the request **IF ANY** of the webhooks return a denial



```
$ kubectl exec
error: nice try bob -alice
```

Kubernetes API Server

Admission Control

External Admission Webhooks

Demo

Webhooks: Lessons Learned

- Be careful with webhook dependencies!
  - Consider performance and availability
  - Avoid side effects

Webhooks: Lessons Learned

- Be careful with webhook dependencies!
    - Consider performance and availability
    - Avoid side effects

- API server sends “internal representation” of Kubernetes objects over the wire

Webhooks: Lessons Learned

- Be careful with webhook dependencies!
  - Consider performance and availability
  - Avoid side effects

- API server sends “internal representation” of Kubernetes objects over the wire

- API server “fails open” if webhook fails (configurable in 1.9)

Webhooks: Lessons Learned

- Be careful with webhook dependencies!
  - Consider performance and availability
  - Avoid side effects
- API server sends “internal representation” of Kubernetes objects over the wire
- API server “fails open” if webhook fails (configurable in 1.9)
- Must serve POST requests at https://<ip>:<port>/ (paths supported in 1.9)

Webhooks: Lessons Learned

- Be careful with webhook dependencies!
  - Consider performance and availability
  - Avoid side effects
- API server sends “internal representation” of Kubernetes objects over the wire
- API server “fails open” if webhook fails (configurable in 1.9)
- Must serve POST requests at https://<ip>:<port>/ (paths supported in 1.9)
- Client-go vendoring has improved significantly

Webhooks...all the way down?

- Webhooks (and initializers) lay the groundwork for extensible policy enforcement

Webhooks...all the way down?

- Webhooks (and initializers) lay the groundwork for extensible policy enforcement
- Policy decisions have been decoupled from enforcement

Webhooks...all the way down?

- Webhooks (and initializers) lay the groundwork for extensible policy enforcement

- Policy decisions have been decoupled from enforcement

- Is there a better way to author policies that control who can do what?

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx-1493591563-bvl8q
  namespace: production
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    securityContext:
      privileged: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
status:
  containerStatuses:
  - name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2017-08-01T06:34:22Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  startTime: 2017-08-01T06:34:13Z
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx-1493591563-bvl8q
  namespace: production
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    securityContext:
      privileged: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
status:
  containerStatuses:
  - name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2017-08-01T06:34:22Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  startTime: 2017-08-01T06:34:13Z
```

```
# references
spec.containers[0].image
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx-1493591563-bvl8q
  namespace: production
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    securityContext:
      privileged: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
status:
  containerStatuses:
  - name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2017-08-01T06:34:22Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  startTime: 2017-08-01T06:34:13Z
```

```
# references
spec.containers[0].image

# variables and iteration
container = spec.containers[_]
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx-1493591563-bvl8q
  namespace: production
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    securityContext:
      privileged: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
status:
  containerStatuses:
  - name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2017-08-01T06:34:22Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  startTime: 2017-08-01T06:34:13Z
```

```
# references
spec.containers[0].image

# variables and iteration
container = spec.containers[_]

# expressions
container.securityContext.privileged = true
```

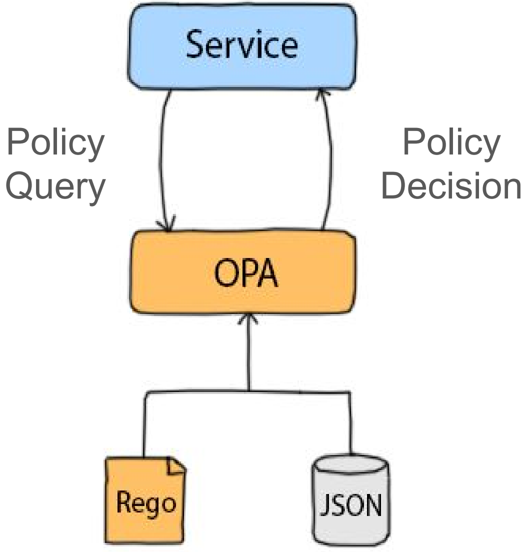
```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx-1493591563-bvl8q
  namespace: production
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    securityContext:
      privileged: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
status:
  containerStatuses:
  - name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2017-08-01T06:34:22Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  startTime: 2017-08-01T06:34:13Z
```

```
# references
spec.containers[0].image

# variables and iteration
container = spec.containers[_]

# expressions
container.securityContext.privileged = true

# functions
is_privileged(container) {
  container.securityContext.privileged = true
}
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: nginx
  name: nginx-1493591563-bvl8q
  namespace: production
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: nginx
    securityContext:
      privileged: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
status:
  containerStatuses:
  - name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2017-08-01T06:34:22Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  startTime: 2017-08-01T06:34:13Z
```

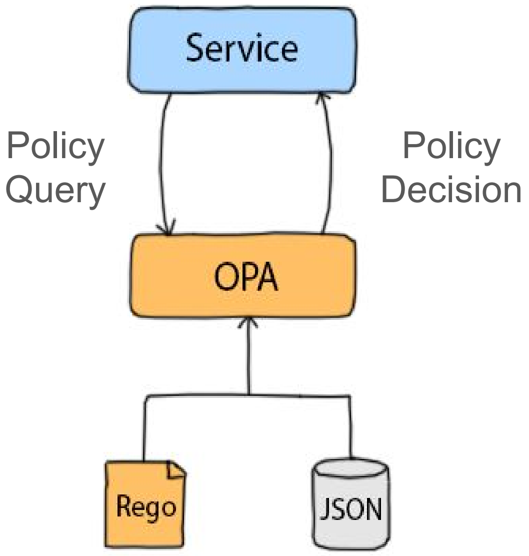
```
# references
spec.containers[0].image

# variables and iteration
container = spec.containers[_]

# expressions
container.securityContext.privileged = true

# functions
is_privileged(container) {
  container.securityContext.privileged = true
}

# rules
deny {
  review.user = “bob”
  review.operation = “CONNECT”
  review.namespace = “production”
  is_privileged(spec.containers[_])
}
```

OPA is an open source, general-purpose policy engine



OPA is an open source, general-purpose policy engine

- Declarative Language (Rego)
  - Is X allowed to call operation Y on resource Z?
  - What clusters should workload X be deployed to?
  - What annotations must be present on object X?



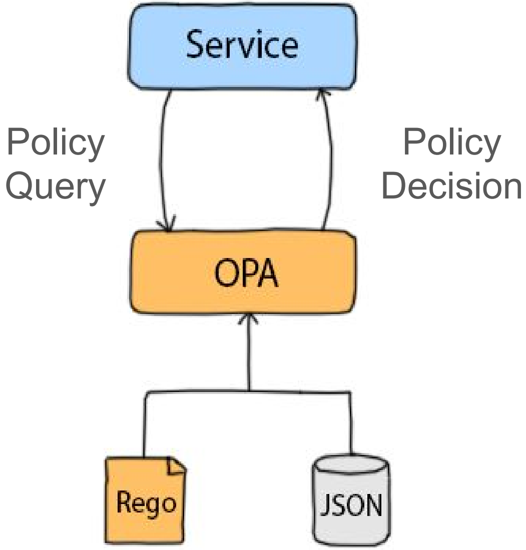
@sometorin

openpolicyagent.org

OPA is an open source, general-purpose policy engine

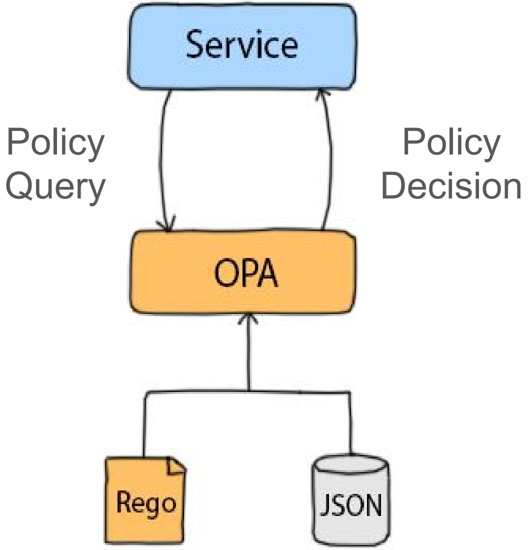
- Declarative Language (Rego)
    - Is X allowed to call operation Y on resource Z?
    - What clusters should workload X be deployed to?
    - What annotations must be present on object X?

- Library/Daemon (Go)
    - In-memory, zero runtime dependencies
    - Evaluation engine: parser, compiler, interpreter
    - Tooling: REPL, test runner, tracing



OPA is an open source, general-purpose policy engine

- Declarative Language (Rego)
  - Is X allowed to call operation Y on resource Z?
  - What clusters should workload X be deployed to?
  - What annotations must be present on object X?
- Library/Daemon (Go)
  - In-memory, zero runtime dependencies
  - Evaluation engine: parser, compiler, interpreter
  - Tooling: REPL, test runner, tracing
- Growing community
  - Sponsored by Styra and Google/Firebase
  - Used by Netflix, Medallia, Huawei, Schuberg Philis, and more
  - Integrations for Istio, Kubernetes, Terraform, PAM, AWS, and more



Service

Policy Query

Policy Decision

OPA

Rego

JSON

@sometorin

openpolicyagent.org

Demo

Standard Library

github.com/open-policy-agent/library

Hashicorp Terraform · amazon web services · kubernetes · docker · Istio

Contributions welcome.

Thank you!

slack.openpolicyagent.org

github.com/open-policy-agent/opa

tsandall/admission-webhook-demo

@sometorin

openpolicyagent.org