Kubernetes Security: Open Policy Agent

October, 2020

Gunjan Patel, Cloud Architect
Robert Haynes, Cloud Security Evangelist

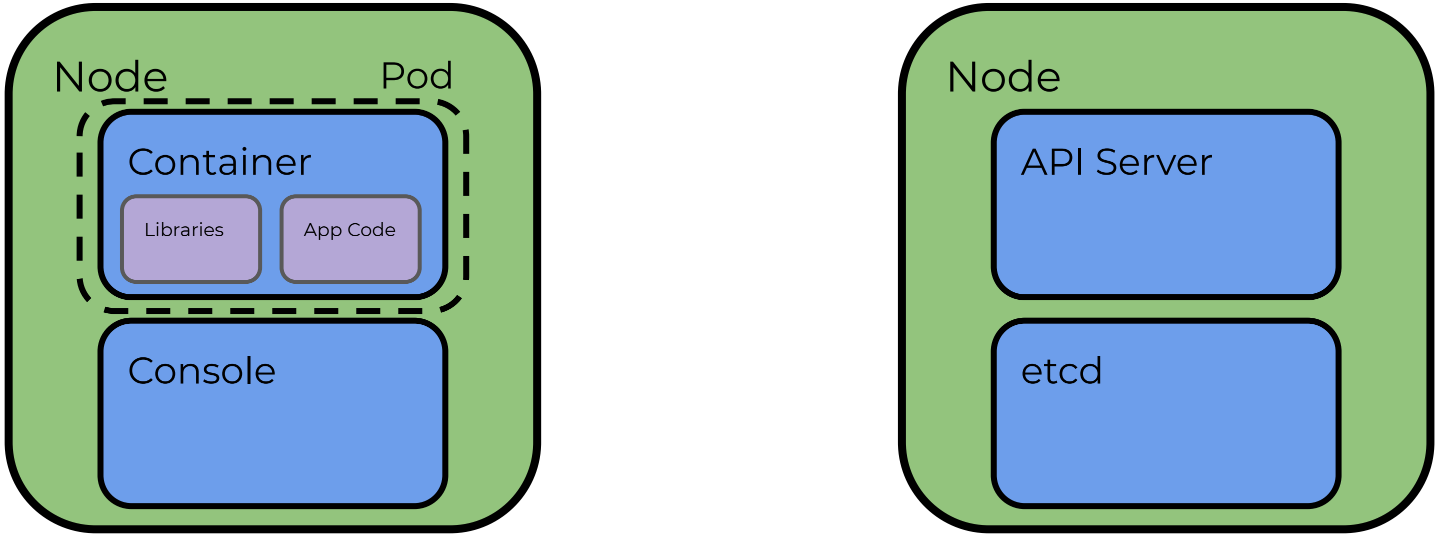
Agenda

- Kubernetes: A short security overview
- Open Policy Agent in Kubernetes: A Control Plane Firewall
- OPA - quick overview
- The Rego language
- Practical Admission Control Policies
- Examples and Demo

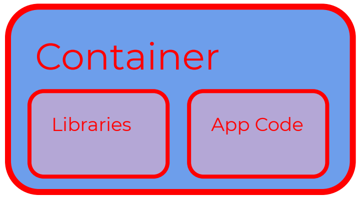
|

Kubernetes Security

Kubernetes common attack vectors

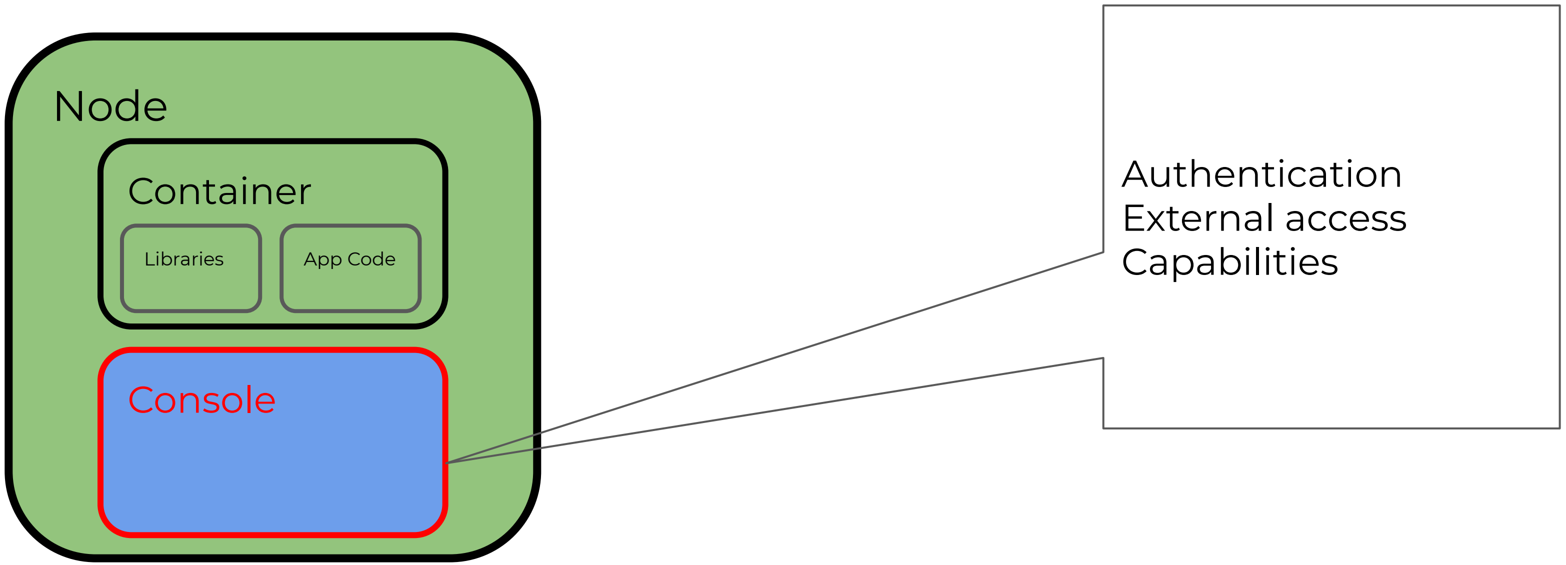


Kubernetes common attack vectors

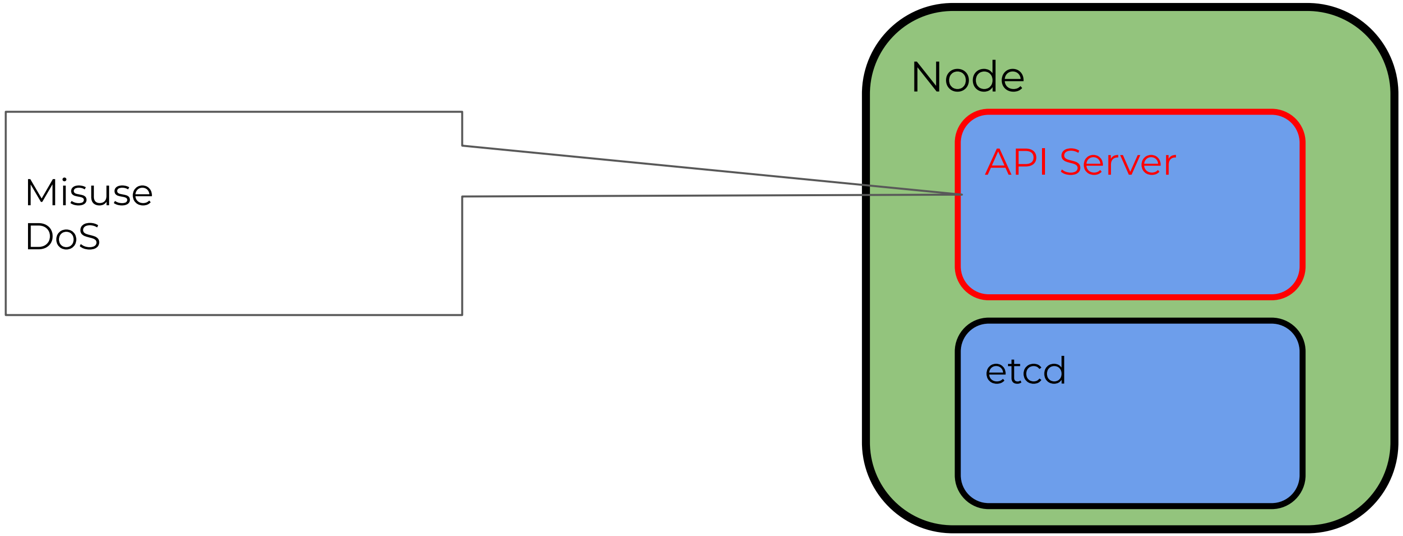


Code
Libraries
Filesystem mounts
Privileged
Network connectivity
Secrets

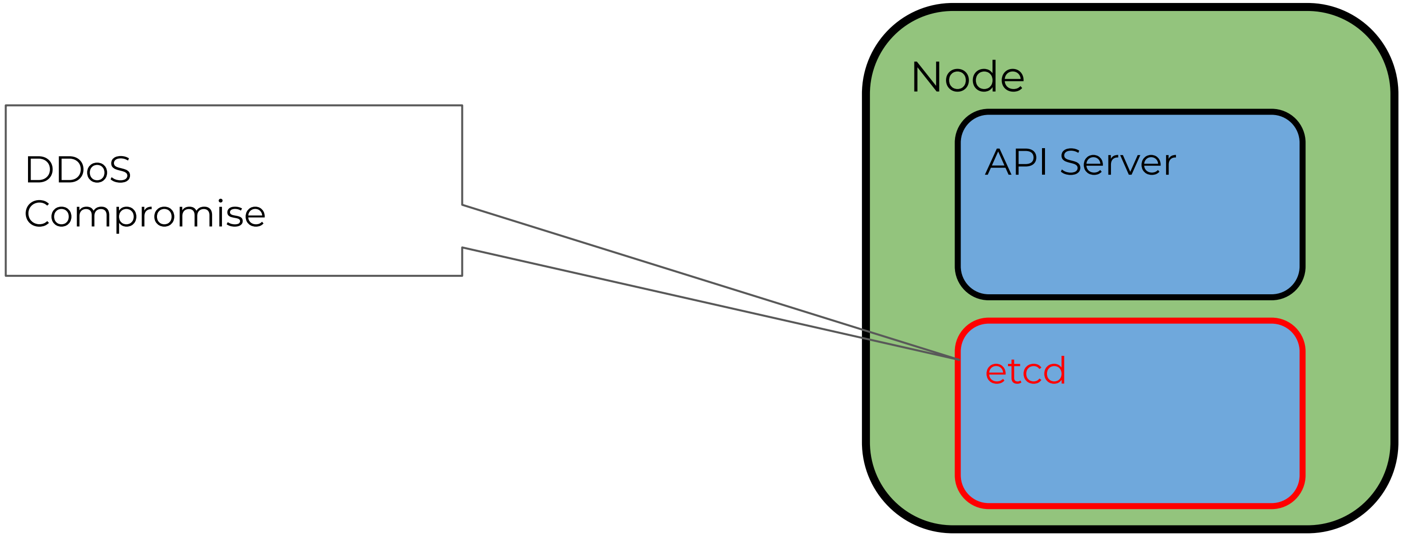
Kubernetes common attack vectors



Kubernetes common attack vectors



Kubernetes common attack vectors



Kubernetes Mitigations and Controls

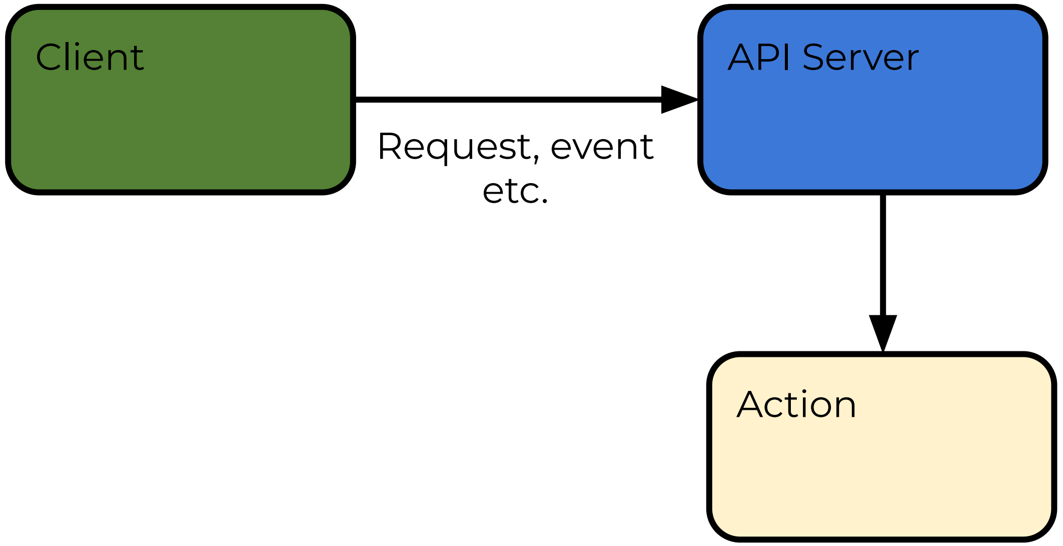
Attack Area	Primary Mitigations
Container Compromise	Cloud Native Firewalls Container image scanning Runtime Defense
Console Compromise	Cloud Firewalls Access Control
Node Compromise	Firewalls Runtime Defense Admission Controllers
API Misuse	Kubernetes RBAC Admission Controllers
etcd attack	Firewall TLS Encryption Limit access

Kubernetes Mitigations and Controls

Attack Area	Primary Mitigations
Container Compromise	Cloud Native Firewalls Container image scanning Runtime Defense
Console Compromise	Cloud Firewalls Access Control
Node Compromise	Firewalls Runtime Defense **Admission Controllers**
API Misuse	Kubernetes RBAC **Admission Controllers**
etcd attack	Firewall TLS Encryption Limit access

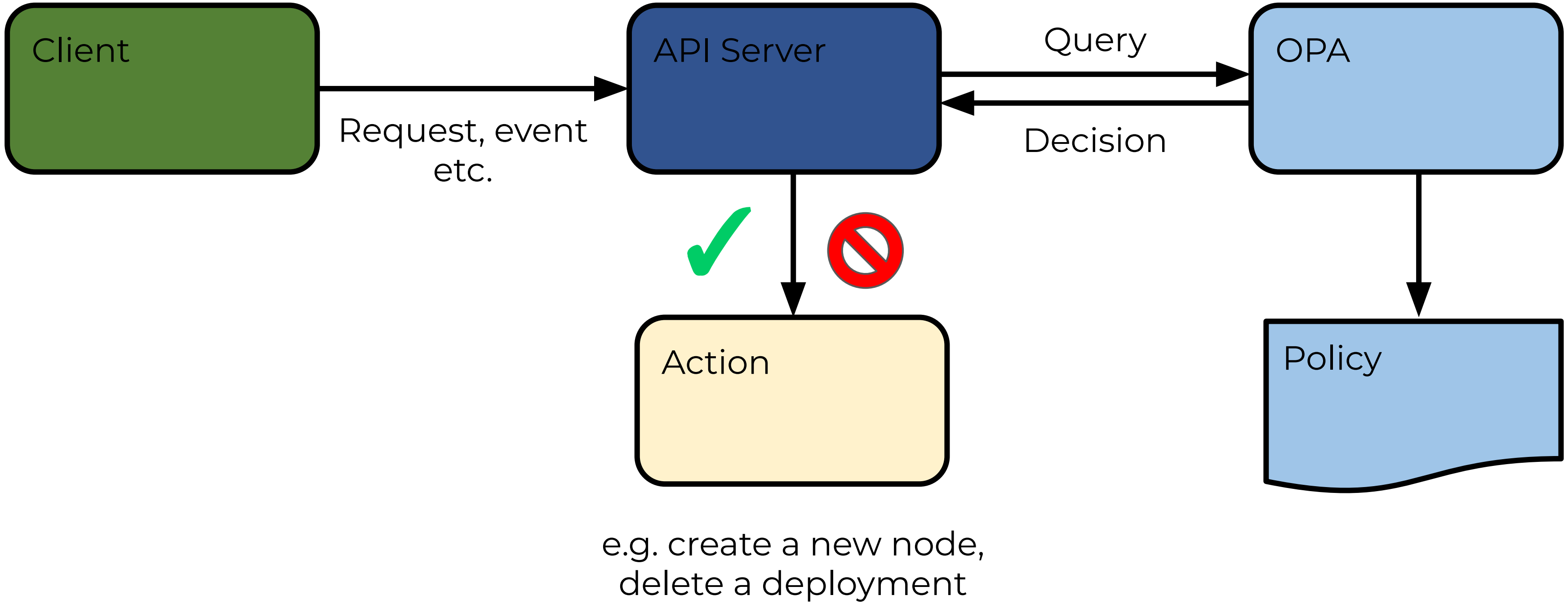
Open Policy Agent

API Server

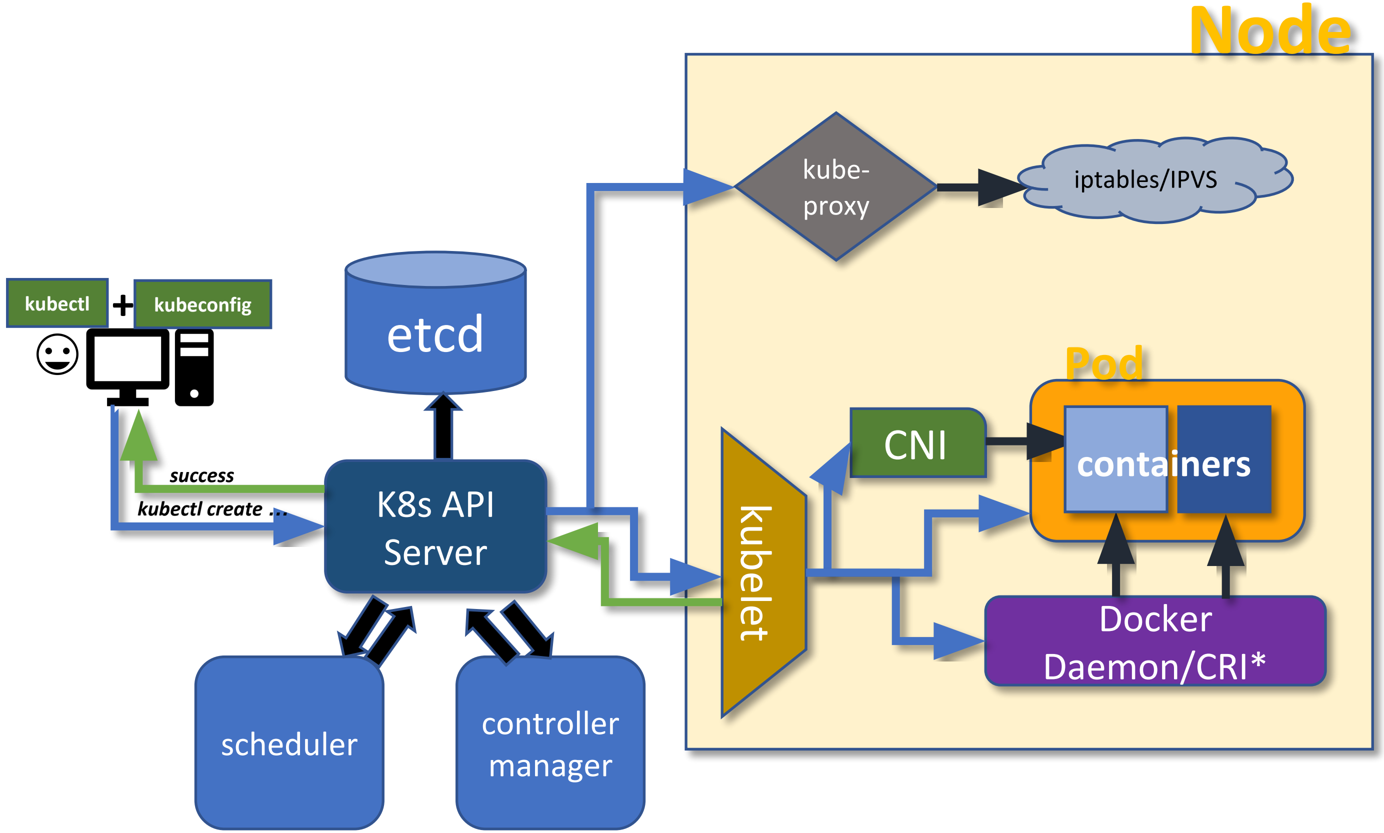


e.g. create a new node, delete a deployment

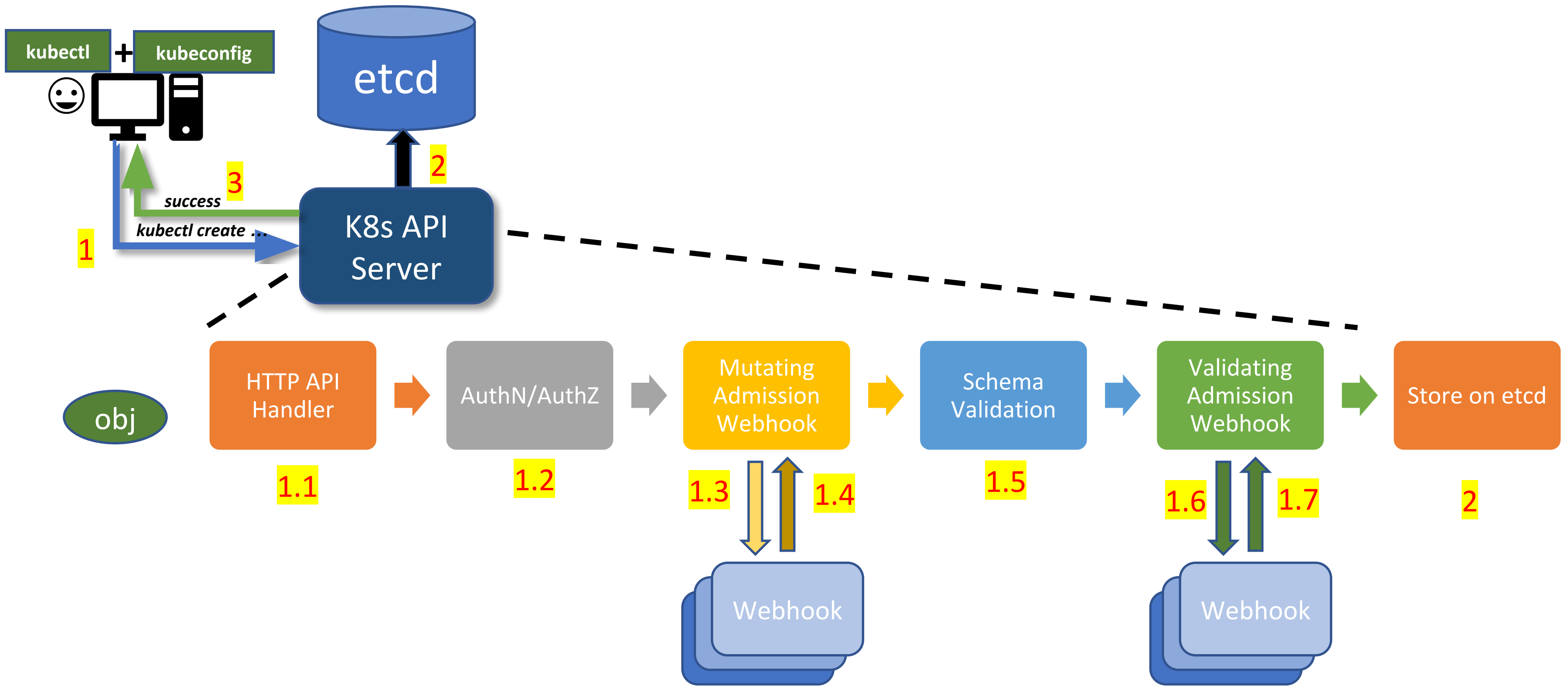
Add Open Policy Agent



Inner workings of kubernetes from 9000 ft



How does it REALLY work? (Zoom in)



The Rego Language

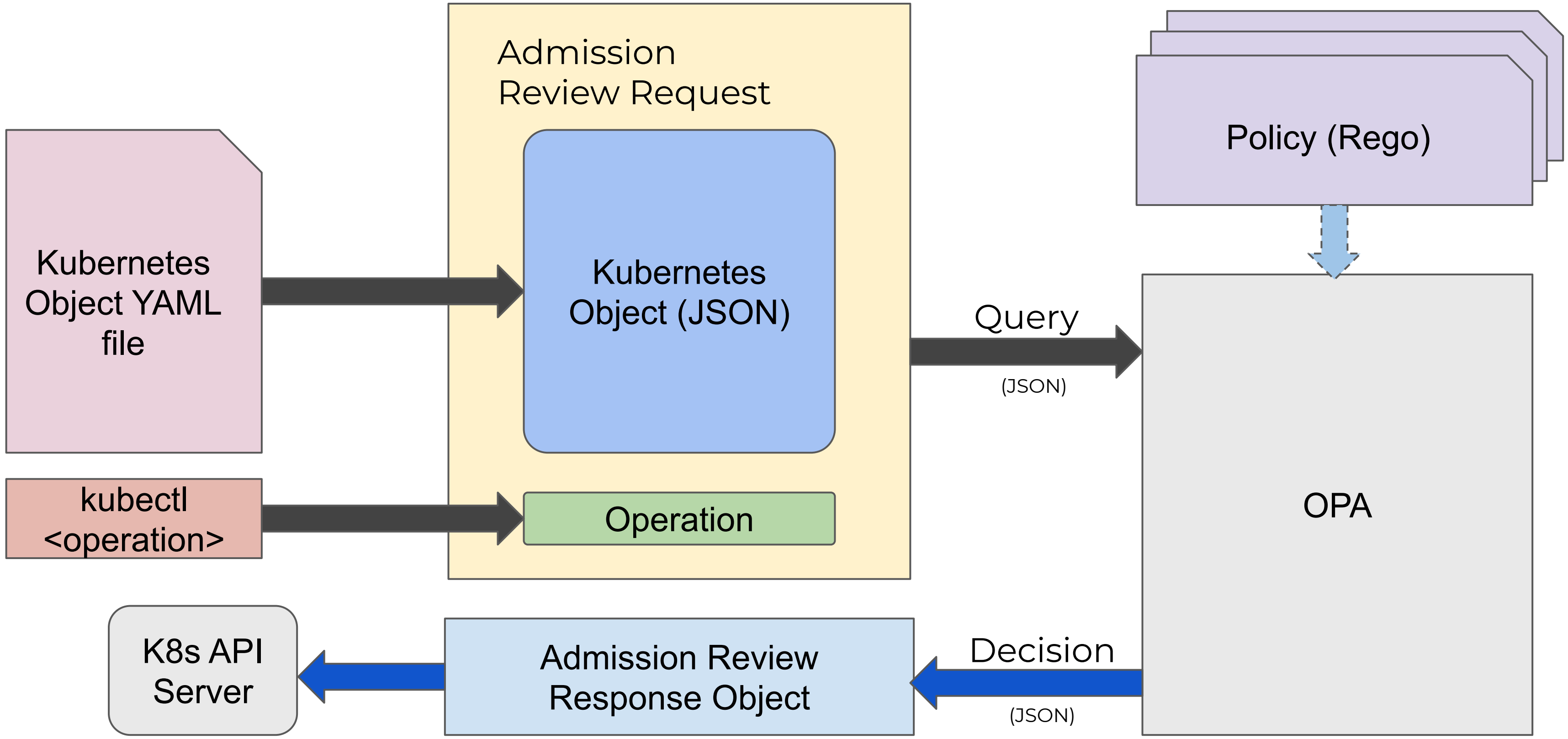
Kubernetes YAML -> JSON

```yaml
apiVersion: v1
kind: Pod
metadata:
 name: static-web
 labels:
   env: dev
spec:
 containers:
   - name: web
     image: nginx
     ports:
       - name: web
         containerPort: 80
         protocol: TCP
```

```json
{
    "apiVersion": "v1",
    "kind": "Pod",
    "metadata": {
        "labels": {
            "env": "dev"
        },
        "name": "static-web",
        "namespace": "default"
    },
    "spec": {
        "containers": [
            {
                "image": "nginx",
                "name": "web",
                "ports": [
                    {
                        "containerPort": 80,
                        "name": "web",
                        "protocol": "TCP"
                    }
                ]
            }
        ]
    }
}
```

Rego Playground Link: https://play.openpolicyagent.org/p/vkp1IdExtK

How are Requests Processed?



|

Rego Language: The Basics

Variables

```
x := 42
allow := true
prefix := “cncf.io/”
```

Equality

```
x == 42
allow != false
“cncf.io” == “cncf.io/”
port >= 30000
```

Lookup

```
val := arr[0]
“foo” == arr[0]
val := obj["foo"]
obj.foo.bar.baz
not obj.foo.bar.baz
```

Built-ins

```
startswith(image, “cncf.io/”)
endswith(image, "latest")
contains(image, “internal”)
trim(list, “ “)
split(path “/”)
count(list)
```

Iteration

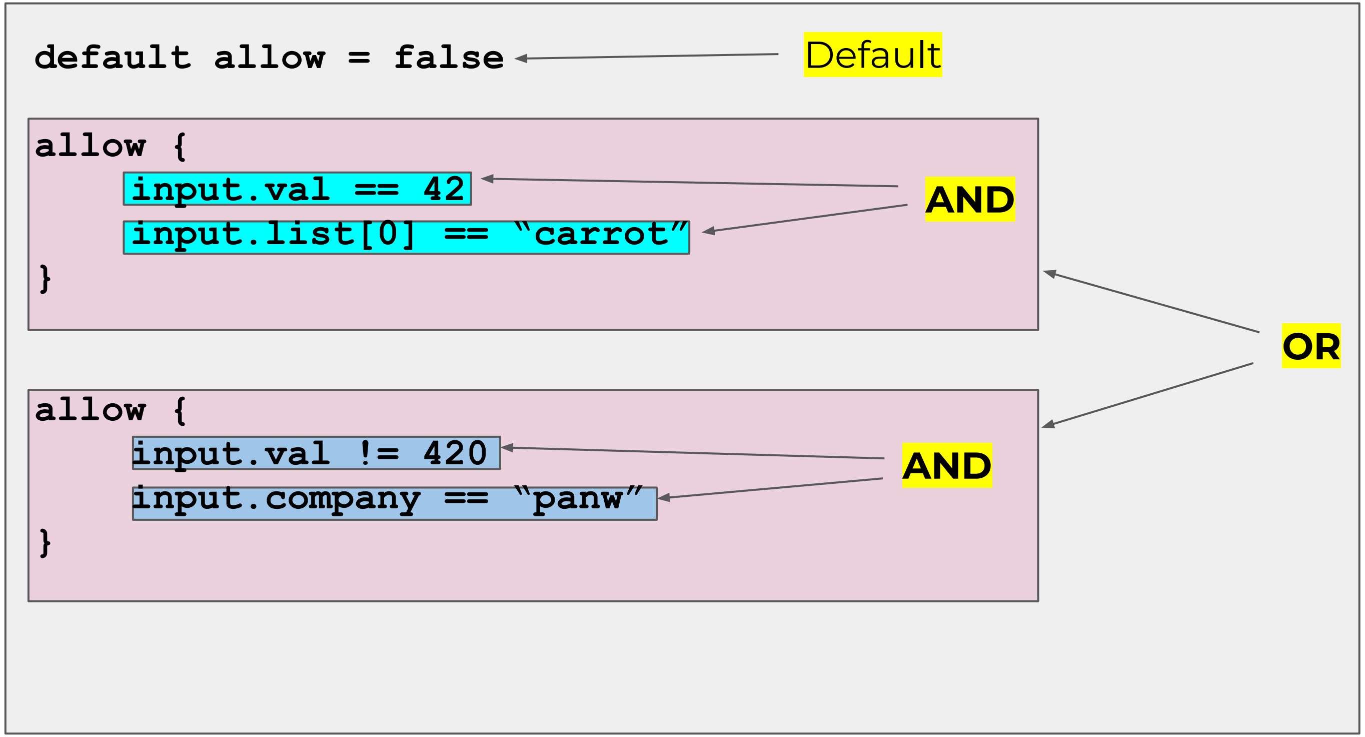
```
x := [“a”, “b”, “c”]

x[index]
x[_]

some i
x[i]

some i, j
x[i]
x[j]
```

Rego Rules



```
default allow = false

allow {
    input.val == 42
    input.list[0] == "carrot"
}

allow {
    input.val != 420
    input.company == "panw"
}
```

Rego Sample

```
match[{"msg": msg}] {
  input.request.operation == "CREATE"
  input.request.kind.kind == "Pod"
  input.request.resource.resource == "pods"
  priv := input.request.object.spec.containers[_].securityContext.privileged
  priv == true
  msg := "Privileged pods denied"
}
```

Kubernetes Security Best Practices

Kubernetes Security Best Practices

1. Only Run containers from a trusted source
2. Don’t run privileged containers for applications
3. Don’t mount the host filesystem
4. Make sure the container filesystem is read-only
5. Don’t allow ‘-dev’, ‘-latest’, or ‘-master’ image tags in prod
6. Block Services of type NodePort

Kubernetes Security Best Practices: https://github.com/gunjan5/cloud-native-security

Use Cases, Examples, Demo

1. Enforce a Trusted Registry

```
match[{"msg": msg}] {
    input.request.kind.kind == "Pod"
    image := input.request.object.spec.containers[_].image
    not startswith(image, "hooli.com")
    Cport := input.request.object.spec.containers[_].ports[_].containerPort
    msg := sprintf("image fails to come from trusted registry: %v", [Cport])
 }
```

2. Prevent Privileged Pods

```
match[{"msg": msg}] {
      input.request.operation == "CREATE"
      input.request.kind.kind == "Pod"
      input.request.resource.resource == "pods"
      input.request.object.spec.containers[_].securityContext.allowPrivilegeEscalation
      msg := "Privilege escalation pod created"
}
```

3. Prevent sensitive host system mounts

```
match[{"msg": msg}] {
      input.request.operation == "CREATE"
      input.request.kind.kind == "Pod"
      input.request.resource.resource == "pods"
      hostPath := input.request.object.spec.volumes[_].hostPath.path
      res := [startswith(hostPath, "/etc"), startswith(hostPath, "/var"),  hostPath ==
"/"]
      res[_]
      msg := "Pod created with sensitive host file system mount"
}
```

4. Make the container filesystem read only

```
match[{"msg": msg}] {
    input.request.operation == "CREATE"
    input.request.kind.kind == "Pod"
    input.request.resource.resource == "pods"
    name := input.request.object.spec.containers[_].name
    sc := input.request.object.spec.containers[_].securityContext
    not sc.readOnlyRootFilesystem
    msg := sprintf("container %s  must have a read-only root filesystem defined",
[name] )
}
```

4. Prevent NodePort Services

```
# Prevent NodePort Services

 match[{"msg": msg}] {

  input.request.operation == "CREATE"

  input.request.object.kind == "Service"

  NP := input.request.object.spec.type

  NP == "NodePort"

  msg := "No Services can be created with type NodePort"

 }
```

5. Don’t allow ‘dev’, ‘latest’, or ‘master’ image tags in prod

```
# Restrict Image tags

match[{"msg": msg}] {

    input.request.kind.kind == "Pod"

    image := input.request.object.spec.containers[_].image

    res := [ endswith(image, "latest"), endswith(image, "master"), endswith(image,
"dev")]

    res[_]

    msg := sprintf(" The image \"%v\" is tagged dev, prod, or latest which are not
allowed.", [image])

 }
```

Wrap, resources, and questions

Resources

Example Policies

https://github.com/twistlock/sample-code/tree/master/opa-rego-policies

OPA

https://github.com/open-policy-agent/opa

Container Security Best Practices

https://github.com/gunjan5/container-security

Overview Blog (vendor content)

https://blog.paloaltonetworks.com/prisma-cloud/open-policy-agent-support/

Prisma Cloud Admission Controller (vendor content)

https://docs.paloaltonetworks.com/prisma/prisma-cloud/20-04/prisma-cloud-compute-edition-admin/access_control/open_policy_agent.html

?

Thank you

paloaltonetworks.com