A Tale of Securing Containerized Workloads at Scale

Tommy McCormick
Security Engineer, Datadog

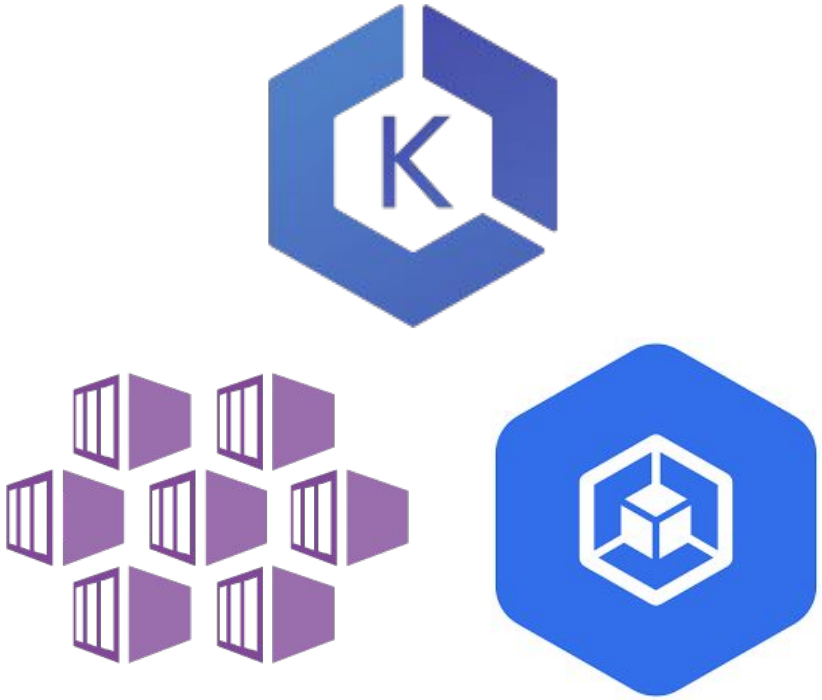
DATADOG

About Me

- Security Engineer @ Datadog, working on securing our Kubernetes platform.
- Datadog is a SaaS service, providing cloud-scale observability and security to any workload.
- Previous work in Detection Engineering, DFIR, and Cloud Security.
- Enjoy exploring cloud native technologies, hiking, and skiing.
- Work remotely out of Atlanta, Georgia, USA.

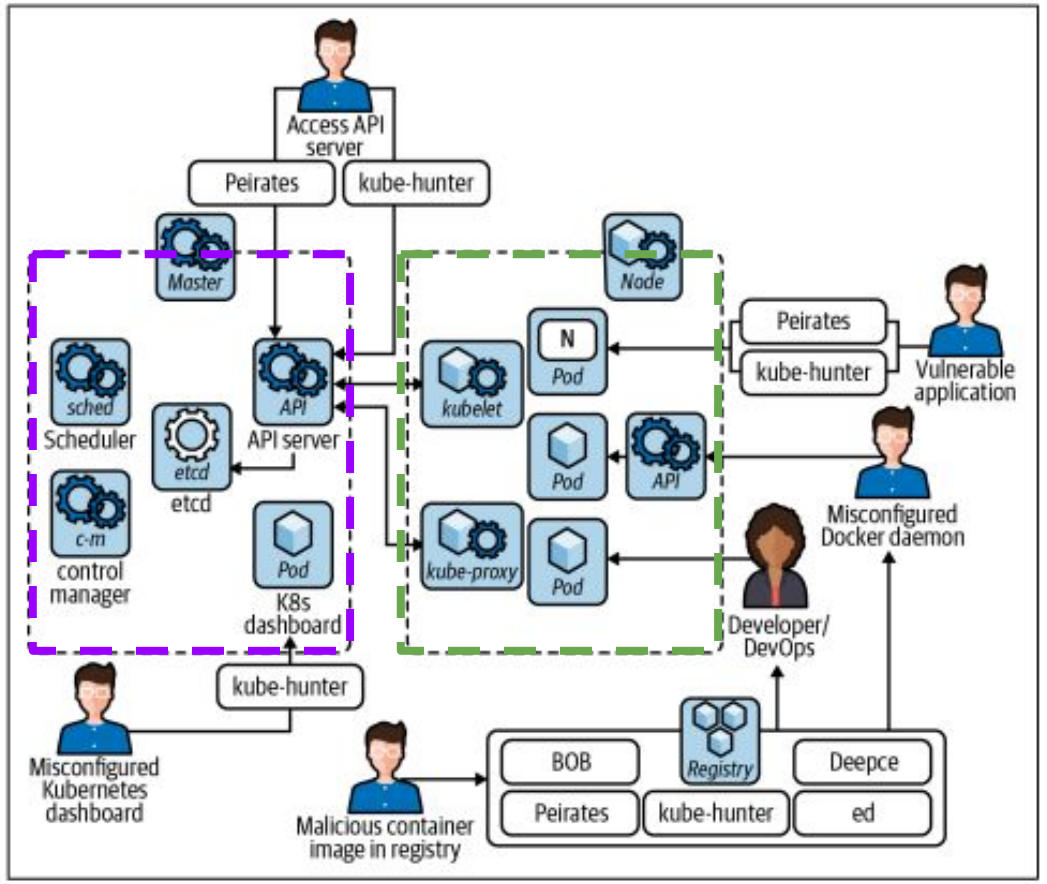
Kubernetes Today

- 96% of respondents to the 2021 CNCF Annual survey are using or evaluating K8s.
- 93% of respondents have experienced a Kubernetes security incident during the last 12 months (Redhat's 2022 State of Kubernetes Security survey).
- 90% of Datadog customers utilize cloud provider-managed solutions for Kubernetes (EKS, GKE, AKS).



Kubernetes Attack Surface

- Control Plane
  - Where your applications are *managed*.
  - etcd, api-server, scheduler, controller-manager.
- Data Plane
  - Where your applications *run*.
  - Application code, deployment configurations, container runtime, CNI plugin, Host OS.



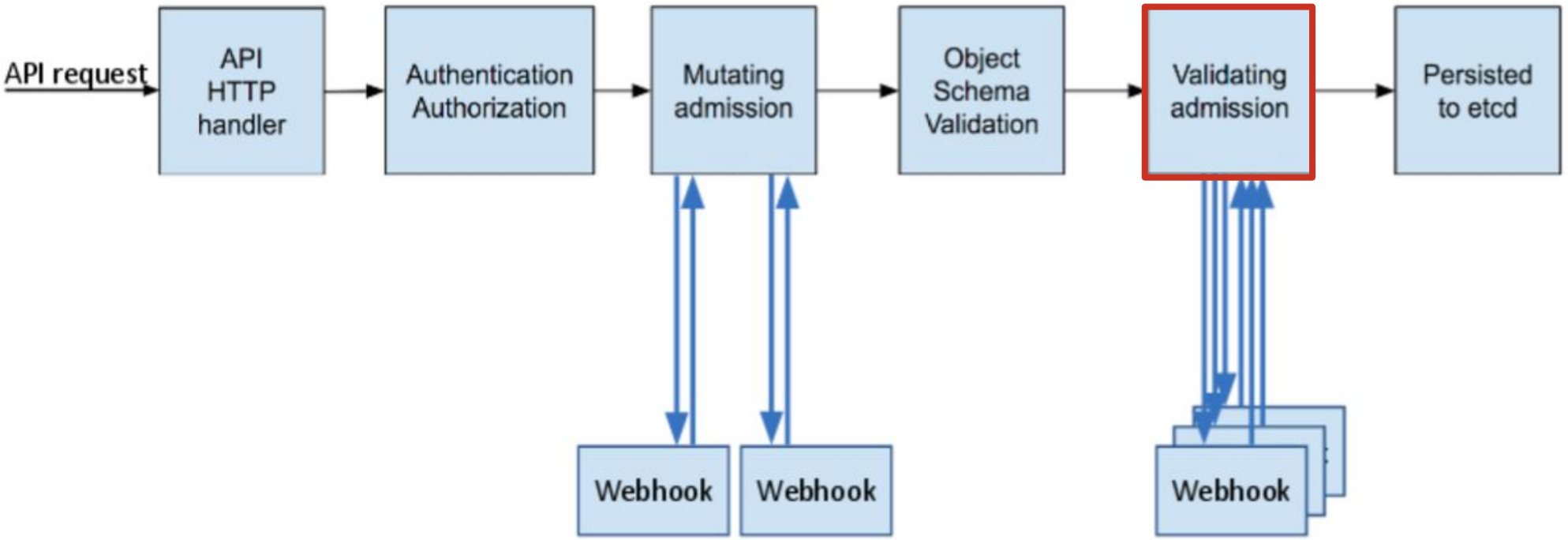
"Hacking Kubernetes" by Andrew Martin & Michael Hausenblas (https://oreil.ly/3b3ql)

Kubernetes Threat Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8s secrets	Access the K8s API server	Access cloud resources
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	cluster-admin binding	Delete K8s events	Mount service principal	Access Kubelet API	Container service account
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from proxy server	Application credentials in configuration files	Instance Metadata API	Application credentials in configuration files
Exposed sensitive interfaces	SSH server running inside container				Access managed identity credential		Writable mounts on host
	Sidecar injection				Malicious admission controller		CoreDNS
							ARP poisoning and IP spoofing

HostPath Volumes

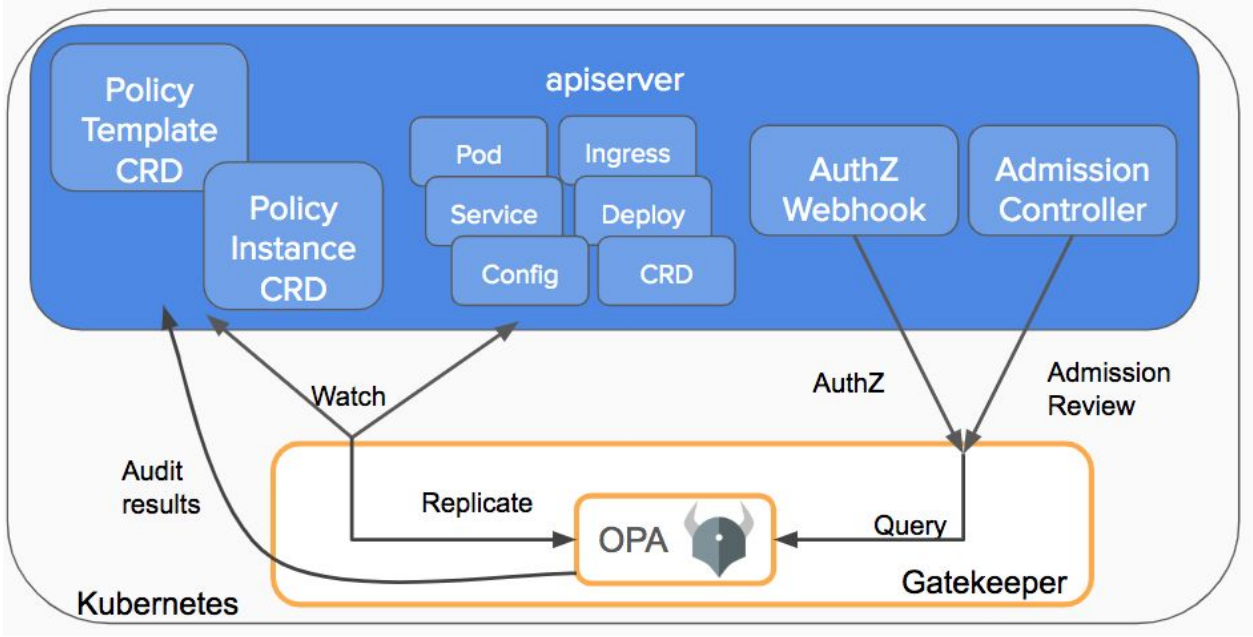
Kubernetes Admission Control



https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/

OPA Gatekeeper

- Admission controller used to validate workload configurations.
- Utilizes the Open Policy Agent policy engine, a CNCF graduated project.
- Deployed in every cluster as an “audit” deployment as well as an admission webhook.
- Policies are written in Rego and deployed as Custom Resources.



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

Rego Policy Language

```
package kubernetes.admission

deny[msg] {
    input.request.kind.kind == "Pod"
    image := input.request.object.spec.containers[_].image
    not startswith(image, "hooli.com/")
    msg := sprintf("image '%v' comes from untrusted registry", [image])
}
```

HostPath Volumes - Denied by Admission

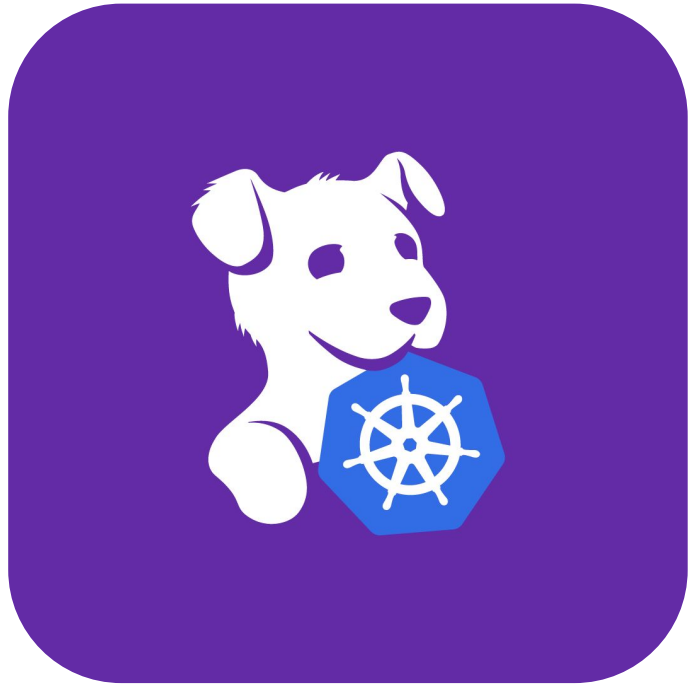
HostPath Volumes - Exceptions

- Commonly used for security agents, or other low-level daemons.
- Usually required for some key functionality of the container (e.g. access to block devices, monitoring resource usage, scanning filesystems, etc.)
- Makes implementation harder!

```
volumeMounts:
- mountPath: /etc/datadog-agent
  name: datadog-agent-config
- mountPath: /etc/datadog-agent/conf.d
  name: datadog-agent-confd
  readOnly: true
- mountPath: /etc/datadog-agent/auth
  name: datadog-agent-auth
- mountPath: /host/proc
  name: proc
  readOnly: true
- mountPath: /host/sys/fs/cgroup
  name: cgroup
  readOnly: true
- mountPath: /opt/datadog-agent/run
  name: logspointer
- mountPath: /var/log/kubernetes
  name: k8s-logs
  readOnly: true
- mountPath: /var/lib/containerd
  name: containerd-images
  readOnly: true
- mountPath: /var/lib/kubelet/pods
  name: empty-dir-volumes
  readOnly: true
```

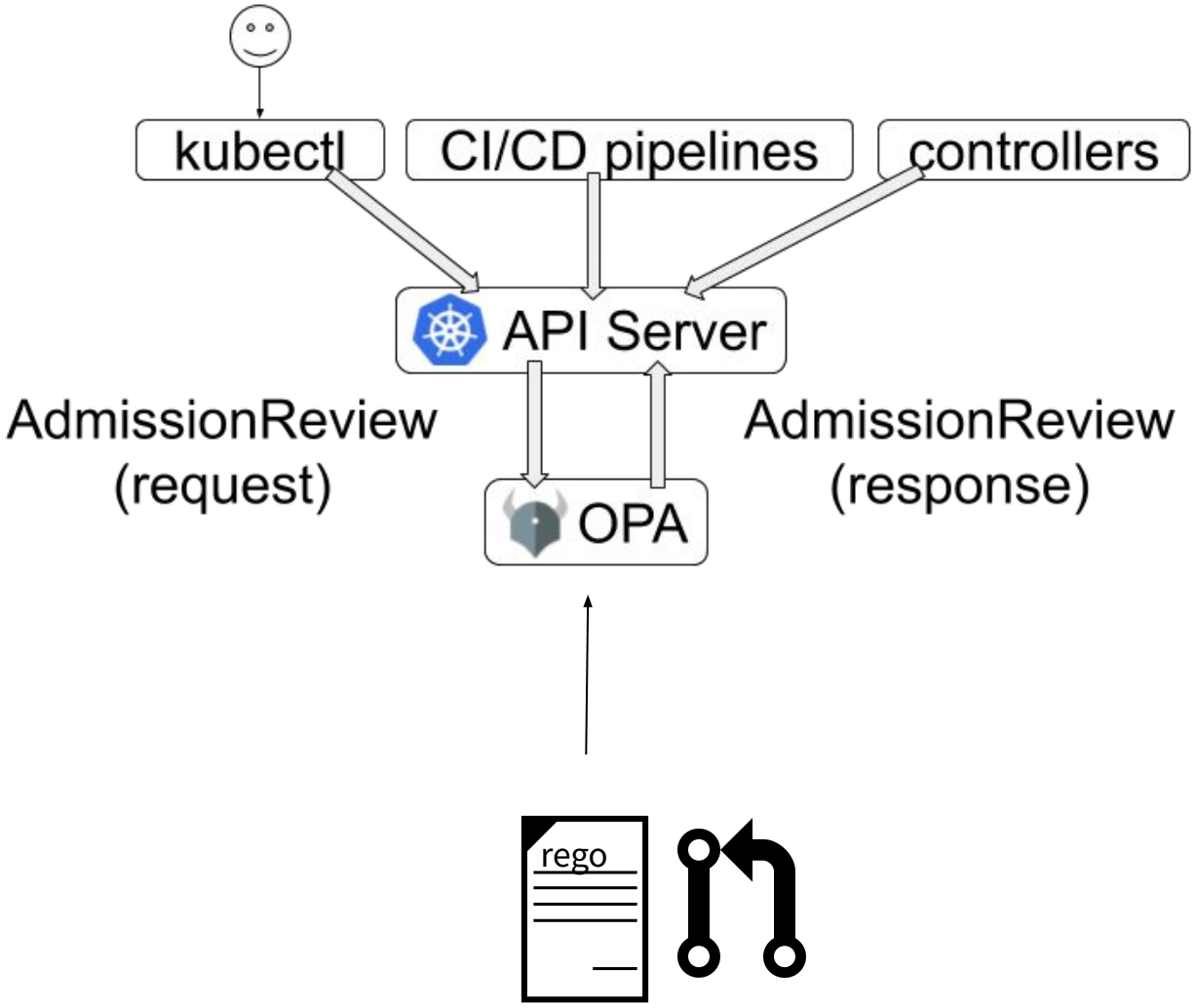
Kubernetes @ Datadog

- Hundreds of thousands of pods
- Tens of thousands of nodes
- 10s of k8s clusters
- Multi-cloud
- 2600+ engineers
- Very fast growth
- Ever-increasing compliance requirements

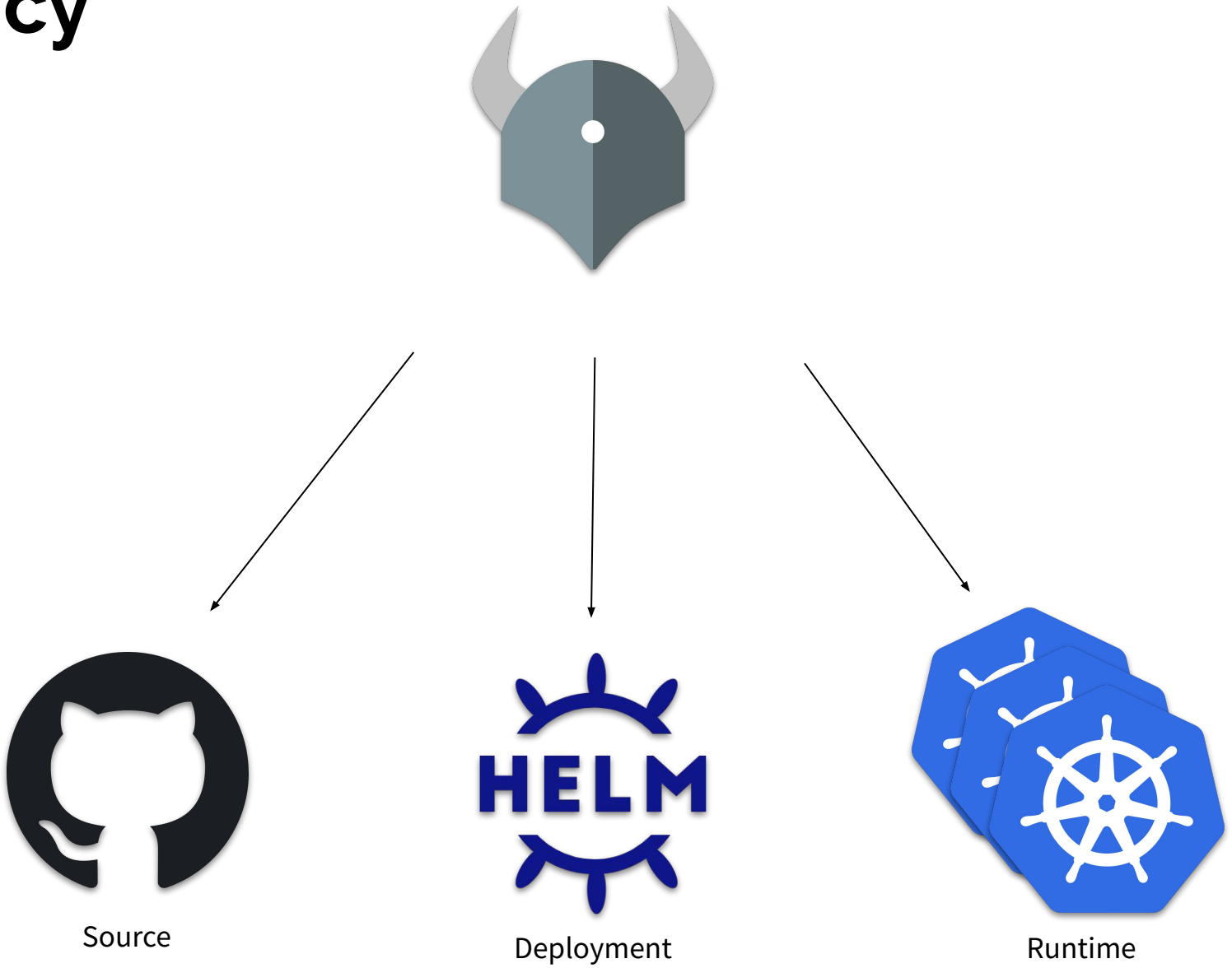


Security as a Platform

- Open documentation, contribution, and collaboration on enforced policies.
- Consistent touch points in all development stages.
- Constraints are treated like any other production code with proper testing and deployment strategies.



Consistency



Runtime Auditing

- Gatekeeper running in “audit” mode surfaces violations for all policies.
- Constraints are deployed in `dryrun` mode initially to assess violations without impact.
- Logs and metrics help identify workloads that may need hardening changes or exceptions.

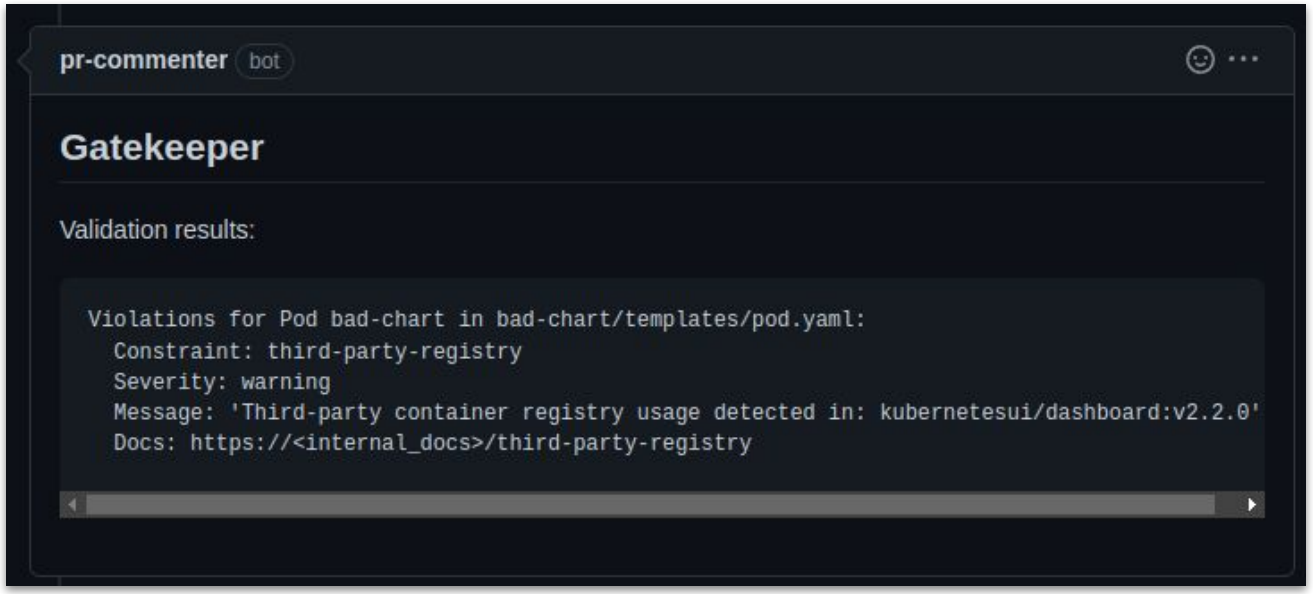
```
Third-party container registry usage detected in: kubernetesui/dashboard:v2.2.0
Docs: https://<internal_docs>/third-party-registry
```

DATE	↑ SERVICE	CONSTRAINT_NAME	RESOURCE...	RESOURCE_NAME	RESOURCE_NAMESPACE	CONTENT
Sep 10 18:18:52.592	gatekeeper-audit	third-party-registry	Pod	dashboard-metrics-scraper-5c64…	kubernetes-dashboard	Third-party container registry usage detected …
Sep 10 18:18:52.592	gatekeeper-audit	third-party-registry	Pod	kubernetes-dashboard-5f5588c88…	kubernetes-dashboard	Third-party container registry usage detected …
Sep 10 18:18:52.592	gatekeeper-audit	third-party-registry	Pod	kubernetes-dashboard-7d9f79556…	kubernetes-dashboard	Third-party container registry usage detected …
Sep 10 18:19:01.599	gatekeeper-audit	third-party-registry	Deployment	dashboard-metrics-scraper	kubernetes-dashboard	Third-party container registry usage detected …
Sep 10 18:19:01.599	gatekeeper-audit	third-party-registry	Deployment	kubernetes-dashboard	kubernetes-dashboard	Third-party container registry usage detected …

Static Analysis

- `gator` and `opa` CLIs provide static analysis of helm charts in CI.
- Surface violations and relevant docs to service owners before workloads are deployed.
- Pull requests reviewed for policy exceptions.

```
$ gator test \
  -f deployment.yaml \
  -f templates-and-constraints
```



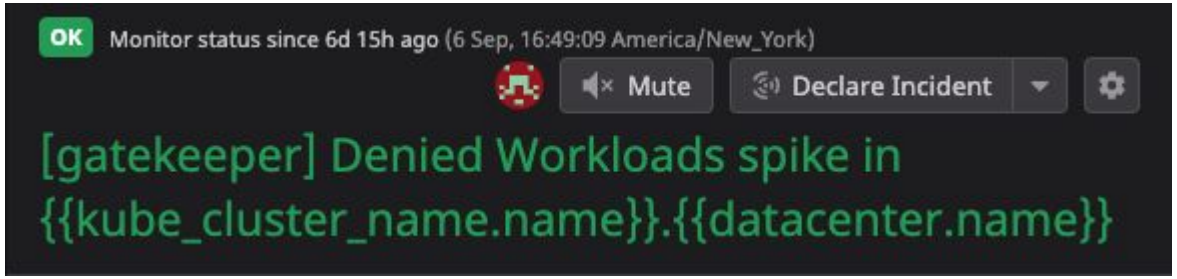
Deployment

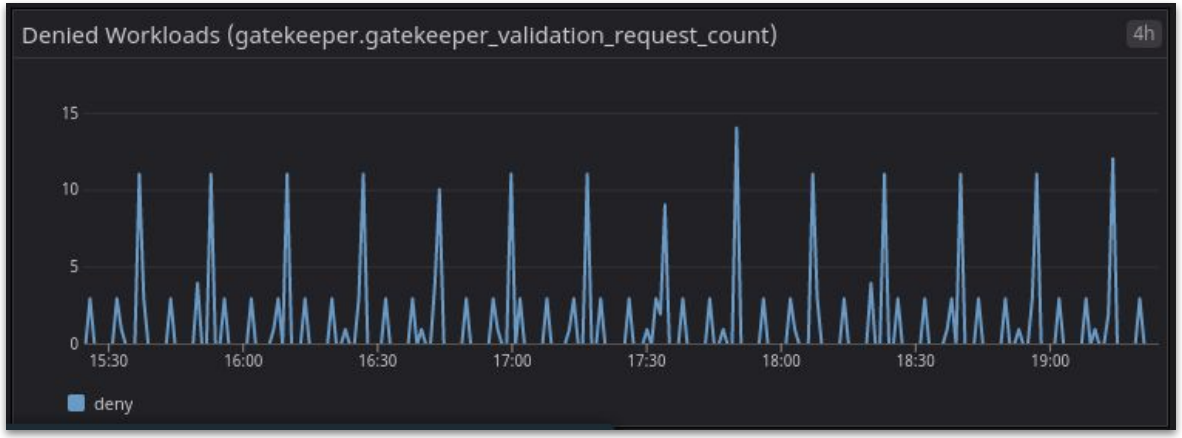
- Gatekeeper admission webhooks deny creation of all workloads violating constraints.
- Webhook response provides the same, consistent message and relevant documentation.

```
$ helm install kubernetes-dashboard kubernetes-dashboard/kubernetes-dashboard
Error from server ([third-party-registry] Third-party container registry usage
detected in: kubernetesui:v2.2.0 ): admission webhook
"validation.gatekeeper.sh" denied the request: [third-party-registry]
Third-party container registry usage detected in: kubernetesui:v2.2.0
Docs: https://<internal_docs>/third-party-registry
```

Observability

- Gatekeeper exports many helpful prometheus metrics.
- Monitoring these has been essential to driving adoption.
- Canary deployments read monitor gates to guide policy rollouts.





Testing

- Gatekeeper policies are treated like any other application code.
- Rego unit tests using `opa` CLI or Gatekeeper test suites with `gator`.
- Work in progress to utilize Kubernetes E2E testing framework for full end-to-end validation.

```yaml
kind: Suite
apiVersion: test.gatekeeper.sh/v1alpha1
metadata:
  name: third-party-registry
tests:
- name: third-party-registry-disallowed
  template: template.yaml
  constraint: samples/constraint.yaml
  cases:
  - name: example-disallowed
    object: samples/example_disallowed.yaml
    assertions:
    - violations: yes
  - name: example-allowed
    object: samples/example_allowed.yaml
    assertions:
    - violations: no
```

```
$ gator verify -v suite.yaml
=== RUN   third-party-registry-disallowed
    === RUN   example-disallowed
    --- PASS: example-disallowed        (0.003s)
    === RUN   example-allowed
    --- PASS: example-allowed   (0.002s)
--- PASS: third-party-registry-disallowed       (0.009s)
```

Admission Controller Alternatives

- Kyverno
  - “Kubernetes Native” policy enforcement
  - Policies written via a CRD.
- Kubewarden
  - WASM module-based policies versioned and fetched from a registry.
  - Can be written in any supported programming language.
- Pod Security Admission
  - Built in admission plugin, replacing Pod Security Policies.
  - Functions by labeling namespaces with various “levels” and enforcement actions for configurations described by Pod Security Standards.

Lessons Learned

- Start enforcing policies as early in the deployment process as possible.
- github.com/open-policy-agent/gatekeeper-library is great baseline policy set, but only supports Pods.
- Rego is a difficult language! Shared libraries of helper functions and robust unit tests can help avoid mistakes.
- Exempting namespaces from admission control will also exempt it from audit scans.
- Deploying “shadow constraints” in dry run mode can help audit exempted namespaces and clean up unneeded exceptions.

Thanks!

Resources

- github.com/open-policy-agent/gatekeeper-library
- https://play.openpolicyagent.org/
- https://hacking-kubernetes.info/
- https://www.redhat.com/en/resources/kubernetes-adoption-security-market-trends-overview