K8Guard
An Auditing System For Kubernetes

Medya Ghazizadeh

Linux Foundation ONS March 26 - 29, 2018
Los Angeles, California
About Me
Apache V2 Legal and Disclaimers!

Can: Commercial Use, Modify, Distribute, Sublicense, Private Use, Use Patent Claims, Place Warranty.

Can NOT: Hold Liable, Use Trademark

Must: Include Copyright, Include License, State Changes, Include Notice.
K8Guard was built in the same retail store that makes this
The Name K8Guard
Pre-K8S Journey

- Migrated from run time configuration (Chef) to Netflix’s Immutable Pattern.
  - Convert chef cookbooks to RPMs
  - Prevent run time errors
- Use Spinnaker to deploy everywhere
- Canary Deployments and Chaos Monkey-ed everything.
- Promote 12 Factor App Design.
Three Things About K8S
Kubernetes is like the best tetris player for your infrastructure
The Shiny Problem definition
Shiny Problem 1: * in Ingress

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test
spec:
  rules:
  - host: *
    http:
      paths:
      - path: /foo
        backend:
          serviceName: s1
          servicePort: 80
```
Shiny Problem 2: 15 GB Image size

Why is that a problem?

How to avoid big image sizes?
Shiny Problem 3: Using Externally hosted images

- Deployment failed image failed to pull from docker hub
- Malicious code in Un-Scannable images
Shiny Problem 4: Root/Privileged Containers

Too much permission will kill you!
If you don’t know how to use Kubernetes!
Shiny Problem 5: Mount a HostPath volume

“hostPath”: A hostPath volume mounts a file or directory from the kubernetes’ host node’s filesystem into your pod. This is not something that most Pods will need, but it offers a powerful escape hatch for some very very rare applications.
Shiny Problem 6: Single Replica

I admit this is the most controversial one!

Some people just don't need failover.
K8S Clusters
Problem Before
Facing Black Friday

Efficiency
Security
Stability
Availability
A Kubernetes Prepared for Black Friday!
Keep The Lights On Even in Disaster!
Solution

Inspired By

The-Twelve-Factor
I. Codebase  II. Dependencies  III. Config  IV. Backing services  V. Build, release, run  VI. Stateless Processes  VII. Port binding  VIII. Concurrency  IX. Disposability  X. Dev/prod parity  XI. Logs  XII. Admin processes
Solution 1 -  WRONG!

Wake up when with a pager’s alert and figure out why k8s cluster is slow or down in the middle of the night.

Set up 1-1 meeting with app teams, preach for them the twelve factor app design and best practices. Sending long email explaining back and forth to each app team’s which does a mistake.
Solution 2 - Scan/Action/Report Automatically

An Automated way to keep the K8S clusters form violations.

- Detect developer’s violations,
- Notify and warn them of their violation
- Take action on them on the violations
- Generate Metrics and Searchable reports for audits.
## First Let's Defining Violation Types

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>Efficiency</td>
<td>5 GB image size</td>
</tr>
<tr>
<td>Image Repo</td>
<td>Security</td>
<td>Downloading image from a shady repo</td>
</tr>
<tr>
<td>Extra Capabilities</td>
<td>Security</td>
<td>Setting UID/GUID</td>
</tr>
<tr>
<td>Privileged Mode</td>
<td>Security</td>
<td>Root containers</td>
</tr>
<tr>
<td>Single Replica</td>
<td>Availability</td>
<td>Not 12-factor app</td>
</tr>
<tr>
<td>Invalid Ingress</td>
<td>Security/Stability</td>
<td>Having &quot;*&quot; in ingress</td>
</tr>
<tr>
<td>Mount Host Vols</td>
<td>Security/Stability</td>
<td>Mounting kubernetes system files</td>
</tr>
<tr>
<td>No Owner</td>
<td>Security</td>
<td>No owner annotation for namespace</td>
</tr>
<tr>
<td>Required Entity</td>
<td>Security/Stability</td>
<td>Required pod not deployed</td>
</tr>
<tr>
<td>Required Annotation</td>
<td>Security/Stability</td>
<td>Required annotation for</td>
</tr>
<tr>
<td>Required Labels</td>
<td>Security/Stability</td>
<td>Required label</td>
</tr>
</tbody>
</table>
Enforce your policy on K8S entities

1. Deployments,
2. Pods
3. Jobs/CronJobs
4. Daemonset
5. Ingresses
6. Namespaces
K8Guard Discover

- Discover service, when in messaging mode, finds violations and puts them on a kafka topic. and also discover API mode, is able to serve without depending on kafka. you can hit the end points to get JSON response.
K8Guard Action

WHAT KIND OF ACTIONS DOES IT TAKE?

- Notifies the namespace owner (email, hipchat, ...).
- After X amount of notifications, it will do a hard action such as:
  - Scale bad deployments down to zero.
  - Suspend bad jobs.
  - Delete bad ingress.

Note that there is a safe mode – which only notifies and does not do hard actions.
Kubernetes Violation!

donotreply.kubernetes@target.com <donotreply.kubernetes@target.com>
Wednesday, May 24, 2017 at 10:30 PM
To: [redacted]

Violation in namespace [redacted] in deployment [redacted]:
- Deployment Name: planning-deployment
- Violation: Invalid Image Size
- Source: merchandizing-planning-deployment:1.0.8
- Warning Count: 1
K8Guard Report

Report service will generate a human readable and searchable report of all the past violation actions.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Source</th>
<th>Violation Source</th>
<th>Violation Type</th>
<th>Action Type</th>
<th>Action Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>minikube</td>
<td>k8guard-report-deployment</td>
<td>local/k8guard-report</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-memcached</td>
<td>k8guard-memcached</td>
<td>SINGLE_REPLICA</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-memcached</td>
<td>memcached:alpine</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-kafka</td>
<td>k8guard-kafka</td>
<td>SINGLE_REPLICA</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-kafka</td>
<td>wurstmeister/kafka:0.10.2.0</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-kafka</td>
<td>wurstmeister/zookeeper</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-discover-deployment</td>
<td>local/k8guard-discover</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-cassandra</td>
<td>k8guard-cassandra</td>
<td>SINGLE_REPLICA</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-cassandra</td>
<td>bitnami/cassandra</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:12 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-action-deployment</td>
<td>k8guard-action-deployment</td>
<td>SINGLE_REPLICA</td>
<td>notify</td>
<td>May 23, 2017 20:27:11 CST</td>
</tr>
<tr>
<td>minikube</td>
<td>k8guard-action-deployment</td>
<td>local/k8guard-action</td>
<td>IMAGE_REPO</td>
<td>notify</td>
<td>May 23, 2017 20:27:11 CST</td>
</tr>
</tbody>
</table>
Discover API generates 19 metrics which are accessible at /metrics, and you can hook a monitoring system like Prometheus to collect them and then generate pretty Grafana dashboards.

Violation metric examples:

- The number of all deployments.
- The number of bad deployments.
- The number of all pods.
- The number of bad pods.

Performance metrics examples:

- The number of seconds took to return all images from Kubernetes api.
- The number of seconds took to return all deployments from Kubernetes api.
Example Grafana Dashboard

Percent of Deployments that have violations

2017-06-24 14:55:13

Percent:
Why Golang?

- Statically linked libraries
- Ship Binary, and run it on Docker Scratch!
- Small language, New libraries,
- Strongly encourages good practices (error handling, formatting, ...)
- Kubernetes is written in Go itself.
Batteries Included! Developer Friendly!

How many times you tried to run a github project and built failed?

Some projects not only don’t give you the batteries, they won’t even give you an idea what kind of batteries you have to buy. Just push the code and good luck!
Makefile!

make help
make build-deploy-minikube
Open Sourcing K8S

Target embraces open source.

second open source project at Target. After winnaker
«Только все знают всё»
Thank You Open source Contributions

- Daemonsets
- HelmCharts
- Bug fixes
- Reddis and minimal technology stack
Open Sourcing Challenges!

There are no bad pull requests! There are only a bad merges!
Future of K8Guard

- Engine Rule.
- Plugin System.
- Cloud Messaging.
- Delegate notification
- Consolidate Technology Stack
Keep in touch, Make your first PR!

- https://github.com/k8guard
- https://k8guard.github.io
- Twitter: @medyadaily