
KubeCon
CloudNativeCon
North America 2018
How we survived our first PCI/HIPPA compliant check with Kubernetes


Nav

Hi

Travis Jeppson

Director of Engineering at Nav

Twitter: @stmpy

Keybase: @stmpy

LinkedIn: linkedin.com/in/stmpy

Should I stay and listen?

The Goal: Help others understand the change associated with adopting a Kubernetes workflow, and still abiding to regulations

Setting up some common language

Regulations: an authoritative rule dealing with details or procedure

Compliance: the act or process of complying to a desire, demand, proposal, or regimen or to coercion

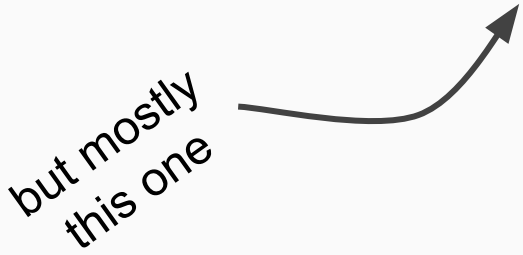
Standard: something established by authority, custom, or general consent as a model or example

Classification: systematic arrangement in groups or categories according to established criteria

Which Regulations?

HIPPA

PCI-DSS



*Or a regulation that deals with protecting data

I'm not a compliance expert

PCI-DSS 12 Requirements

Build and Maintain a Secure Network

1. Install and maintain a firewall configuration to protect cardholder data
2. Do not use vendor-supplied defaults for system passwords and other security parameters

Protect Cardholder Data

3. Protect stored cardholder data
4. Encrypt transmission of cardholder data across open, public networks

Maintain a Vulnerability Management Program

5. Use and regularly update anti-virus software or programs
6. Develop and maintain secure systems and applications

Implement Strong Access Control Measures

7. Restrict access to cardholder data by business need-to-know
8. Assign a unique ID to each person with computer access
9. Restrict physical access to cardholder data

Regularly Monitor and Test Networks

10. Track and monitor all access to network resources and cardholder data
11. Regularly test security systems and processes

Maintain an Information Security Policy

12. Maintain a policy that addresses information security for employees and contractors

PCI-DSS 12 Requirements (that need addressed with )

3. Protect stored cardholder data

5. Use and regularly update anti-virus software or programs

7. Restrict access to cardholder data by business need-to-know

8. Assign a unique ID to each person with computer access

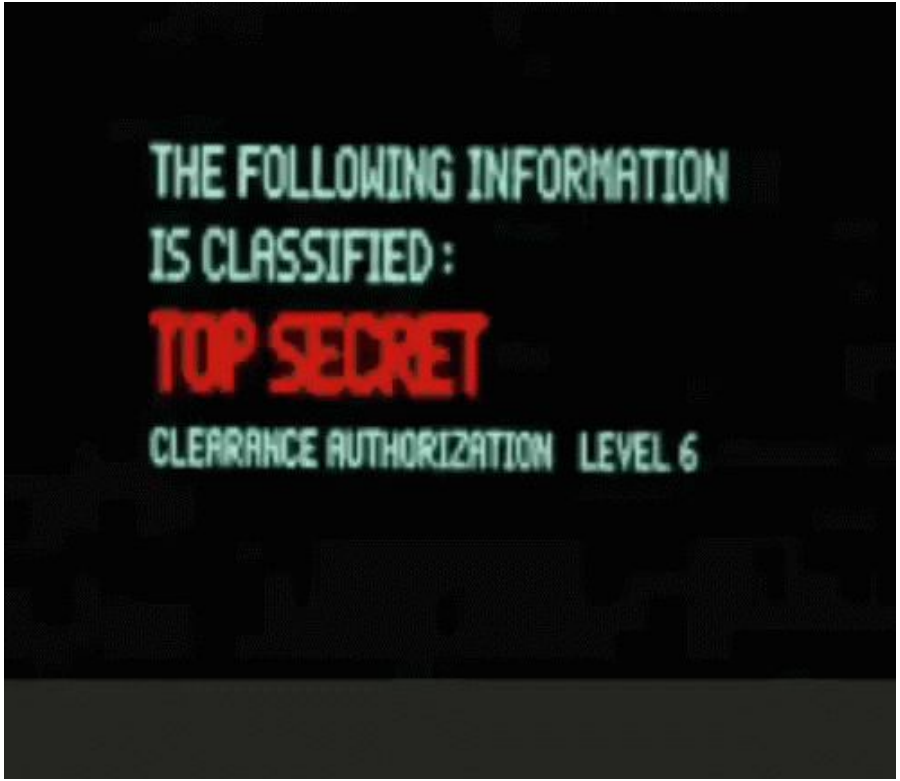
10. Track and monitor all access to network resources and cardholder data

11. Regularly test security systems and processes

Data Protection

Requirement 3: Protect stored cardholder data

Start with Classification!



Data Classification (RED)

Social Security Number

Passport Details

Credit Card Information

Drivers License Number

Personal Credit Data

Medical Records

Anything **directly controlled** or that can be used to **individualize** a person

Data Classification (YELLOW)

Birthday

Age

Part of an address

Gender association

Full (Legal) Name

Full Address

The city you were born in

Your favorite school teacher

Where you met your spouse

Your mother's maiden name

The make and model of your first car

Your favorite childhood friend's name

...

Anything that can be used to individualize a person given multiple data points

Data Classification (GREEN)

Unique ID

Encrypted Data

Binary information

Email address

Username

Site interactive data

Basically impossible to individualize a person, or not regulated (public) data

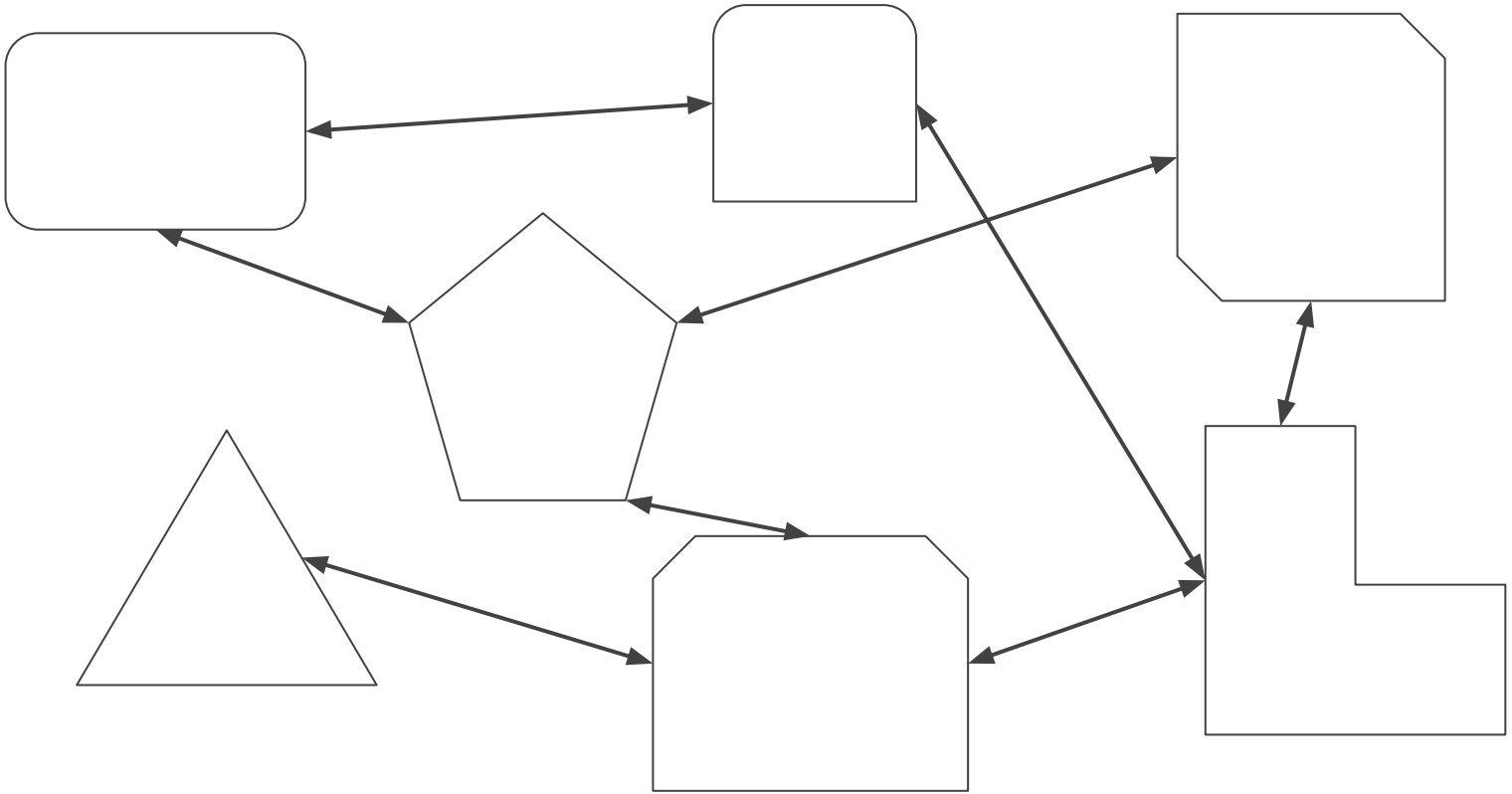
Service Classification follows Data

Service Classification == Data Classification

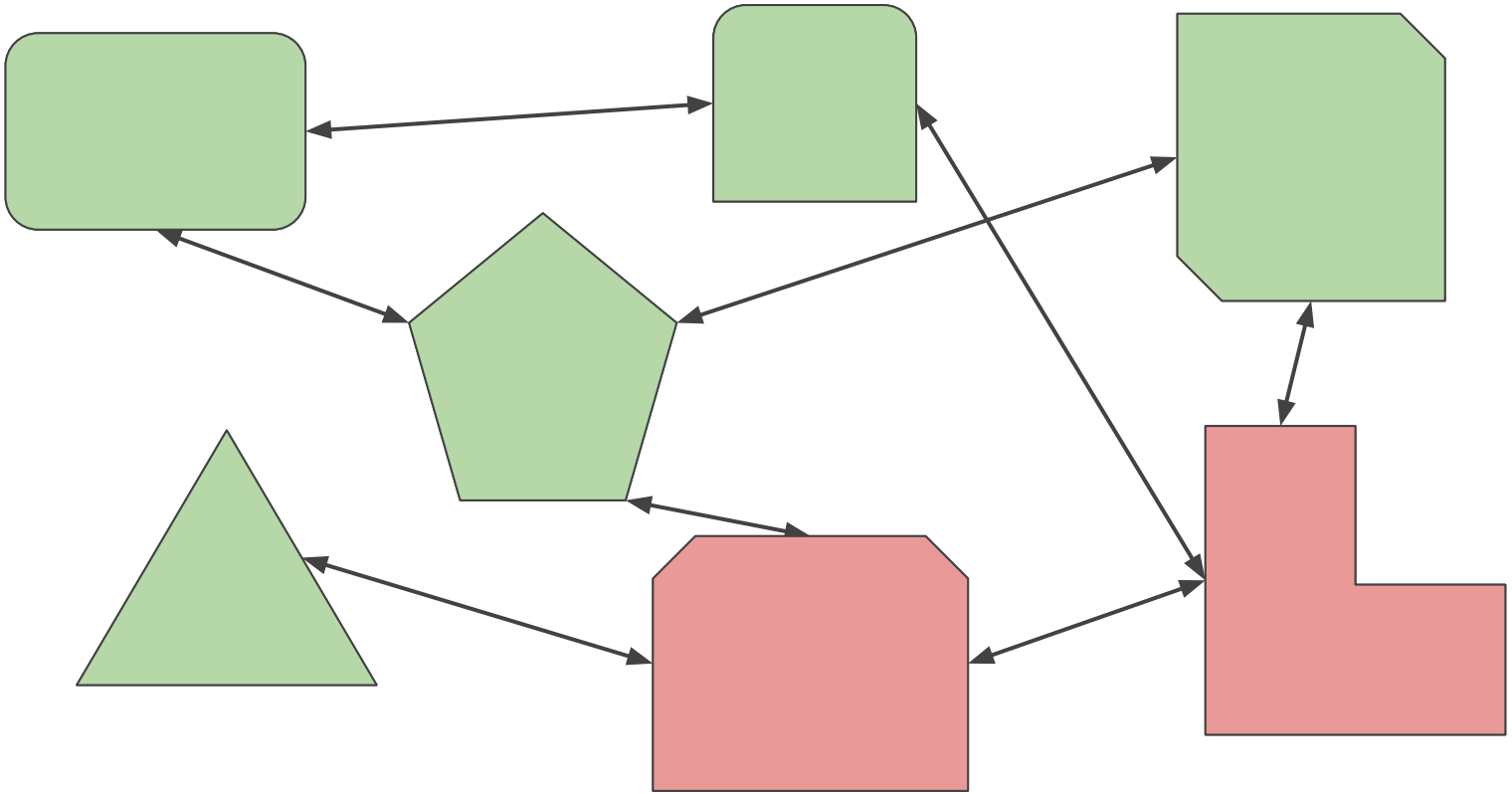
RED == RED

GREEN == GREEN

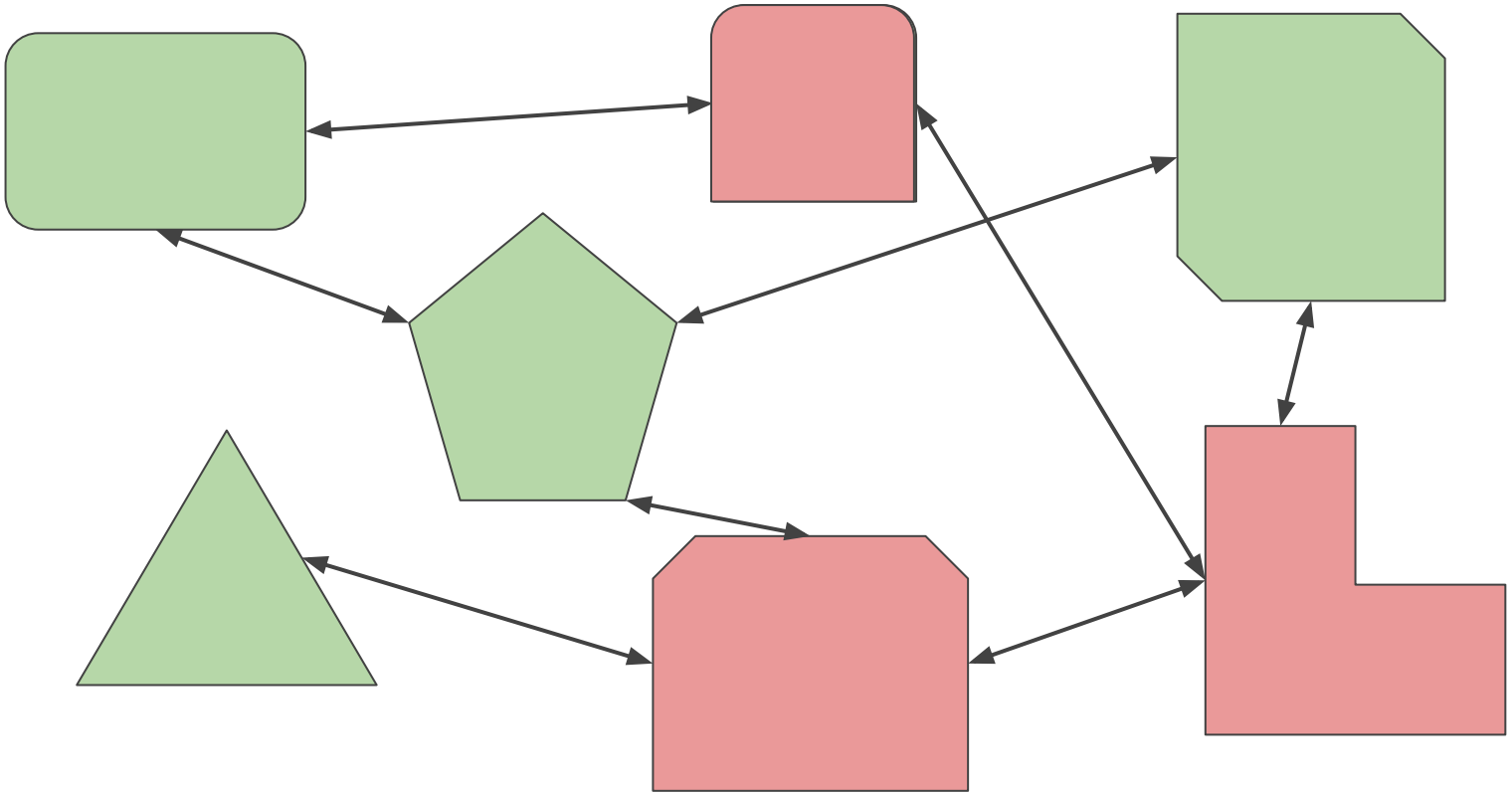
Least Common Denominator



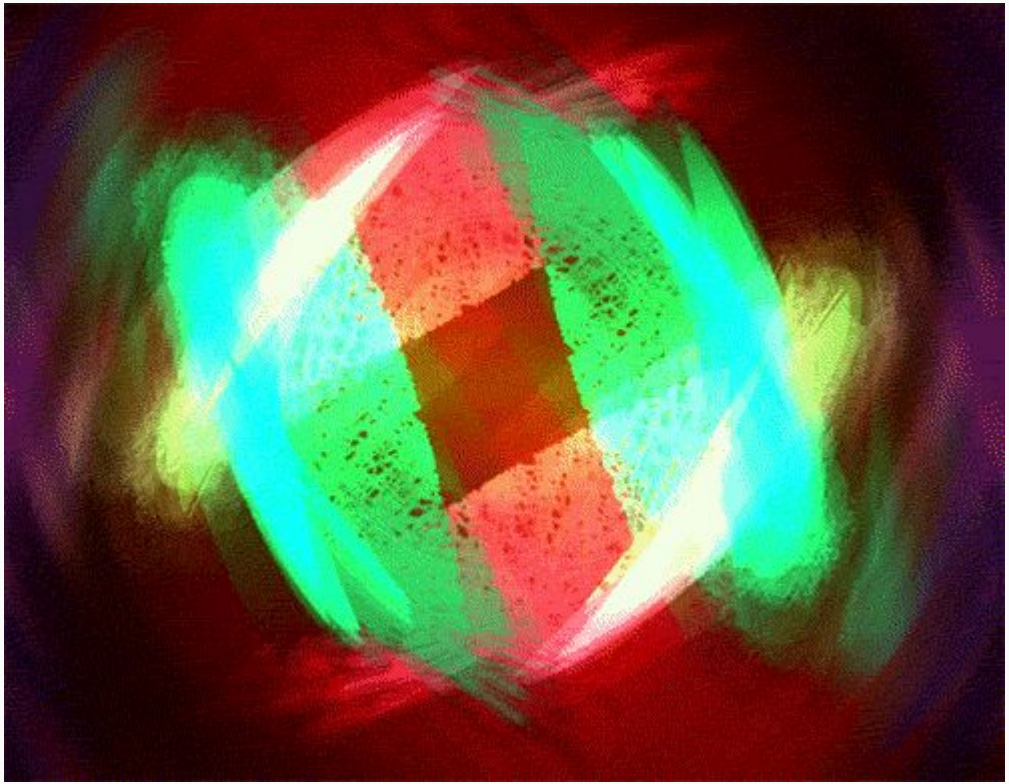
Least Common Denominator



Least Common Denominator



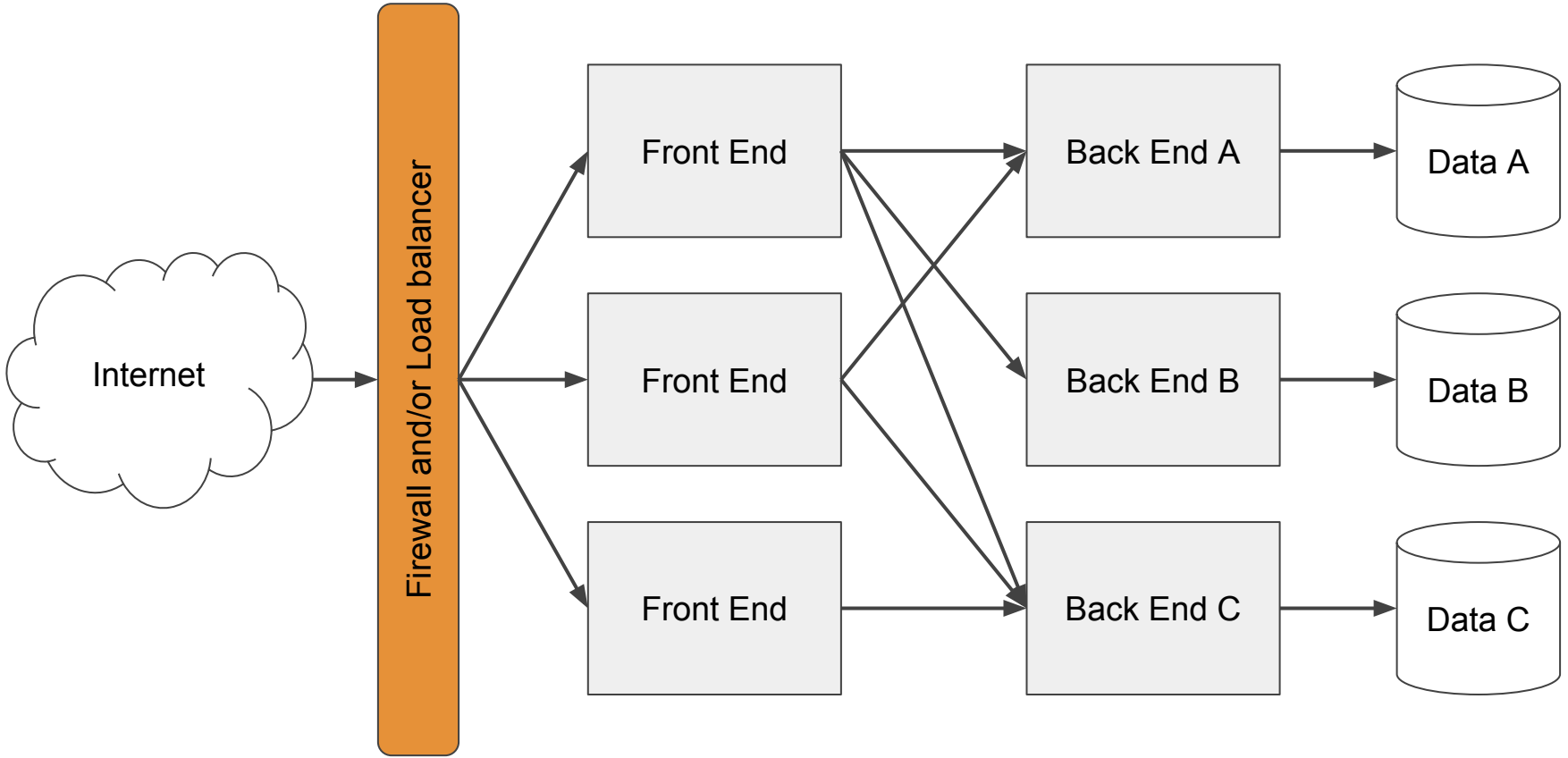
Mixing Green and Red Data



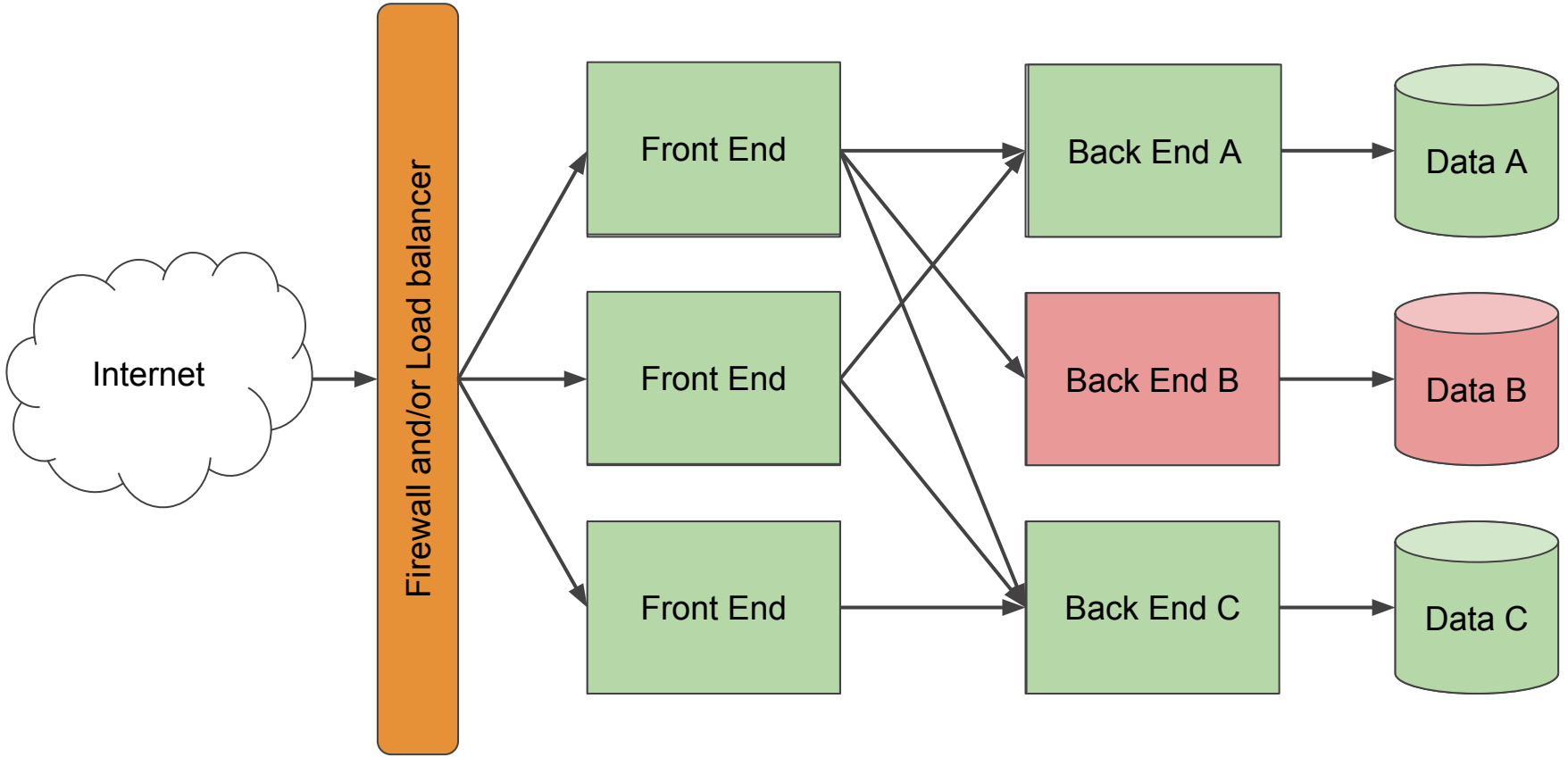
Environment Comparison

Traditional vs Distributed

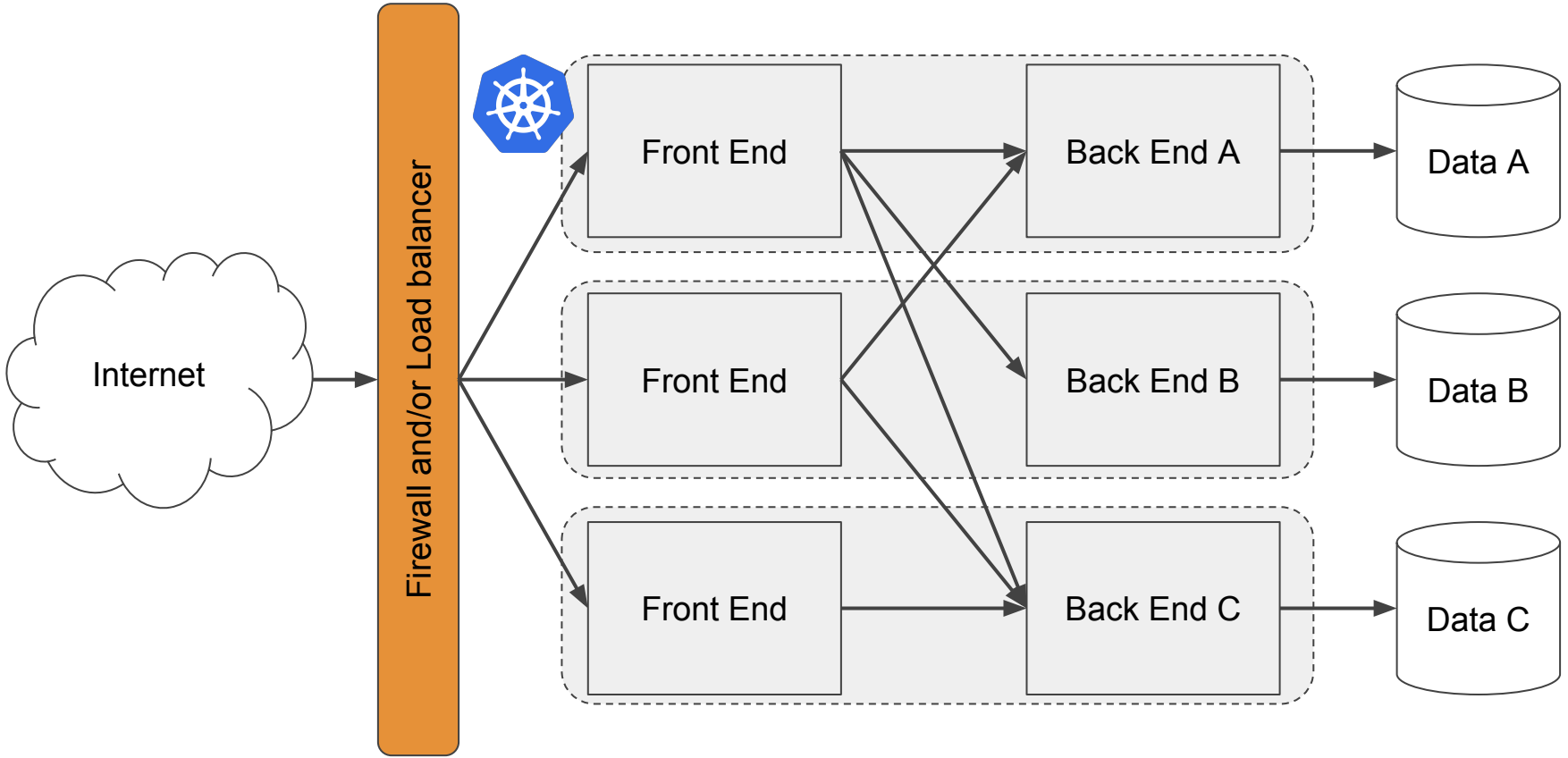
"Traditional" 1:1 - service per machine (vm)



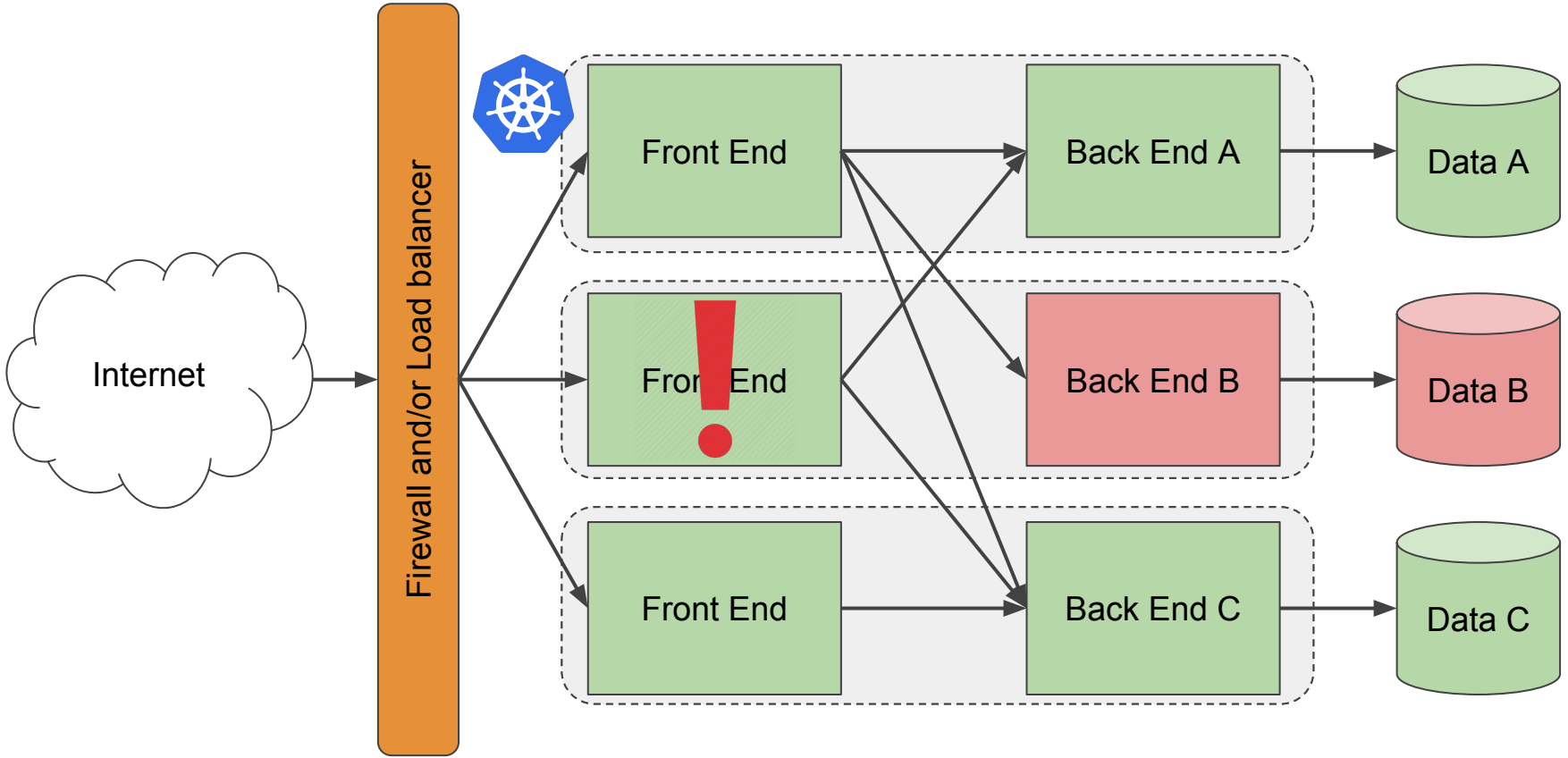
"Traditional" 1:1 - Data Classification



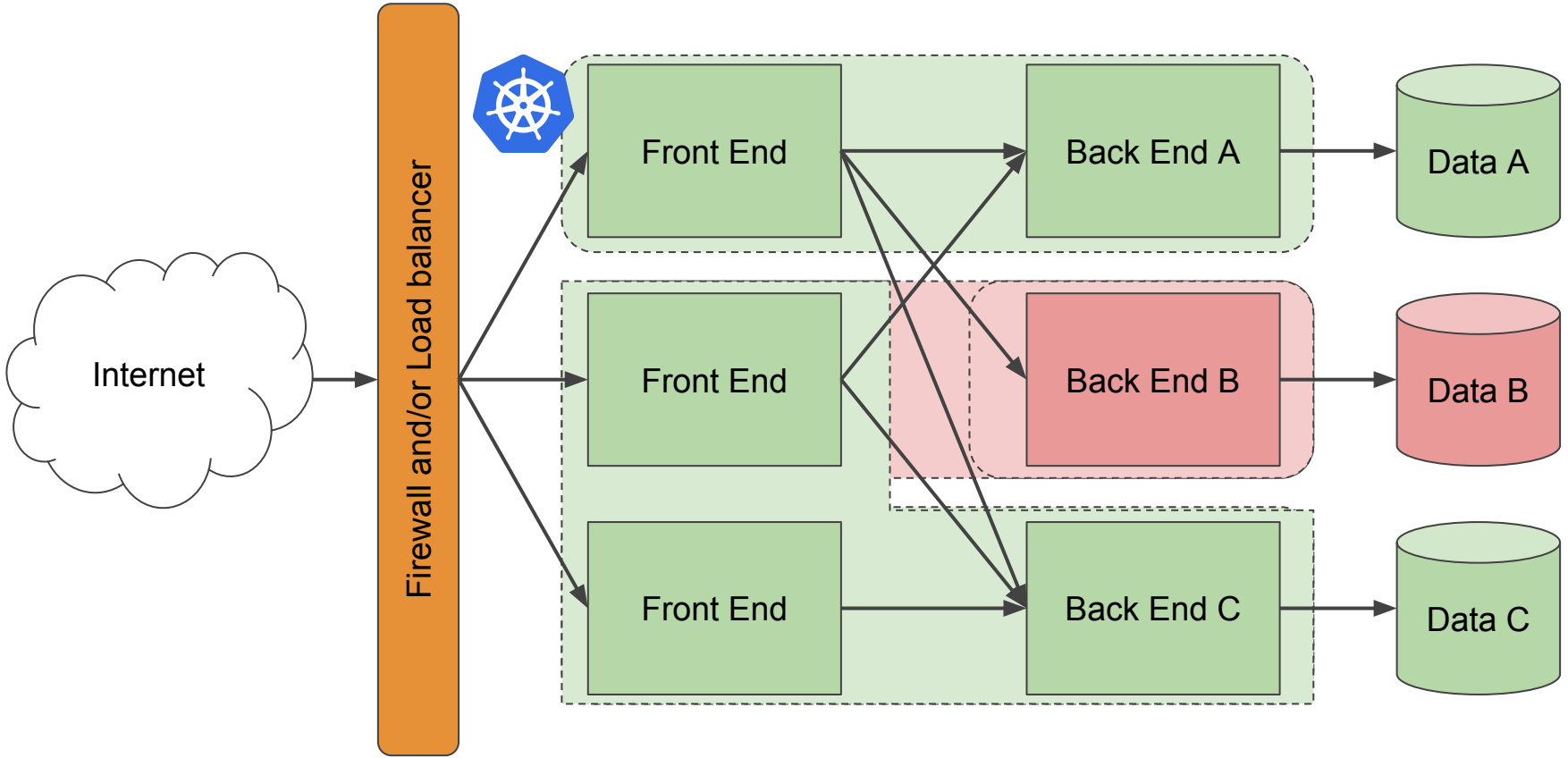
"Distributed" - multiple services per machine



"Distributed" - multiple services per machine



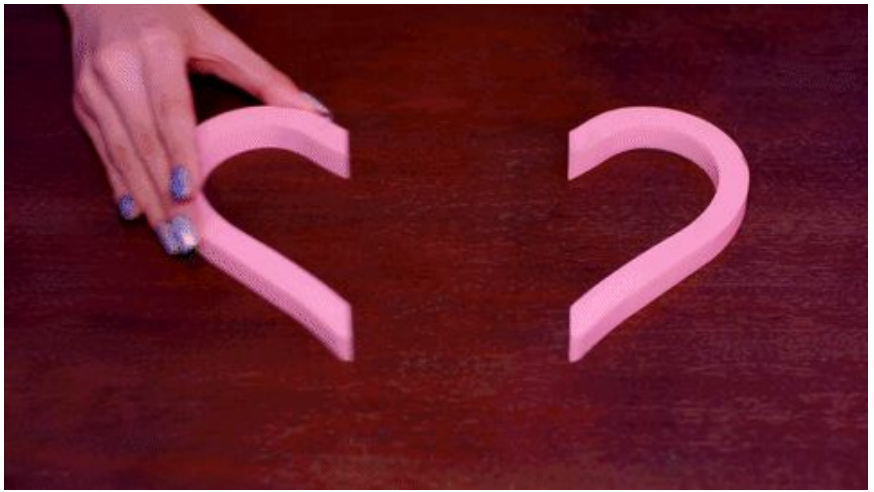
"Classified Distributed" - multiple services per machine



Logical Separations

Taints and Tolerances

https://kubernetes.io/docs/concepts/configuration/taint-and-toleration/



Taints

Applies to a whole node.

```
kubectl taint nodes node1 key=value:NoSchedule
```

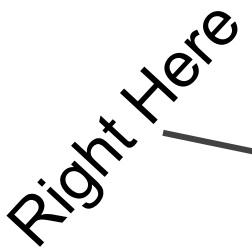
places a taint on node `node1`. The taint has key `key`, value `value`, and taint effect `NoSchedule`. This means that no pod will be able to schedule onto `node1` unless it has a matching toleration.

Tolerance

You specify a toleration for a pod in the PodSpec. Both of the following tolerations “match” the taint created by the `kubectl taint`, and thus a pod with either toleration would be able to schedule onto `node1`:

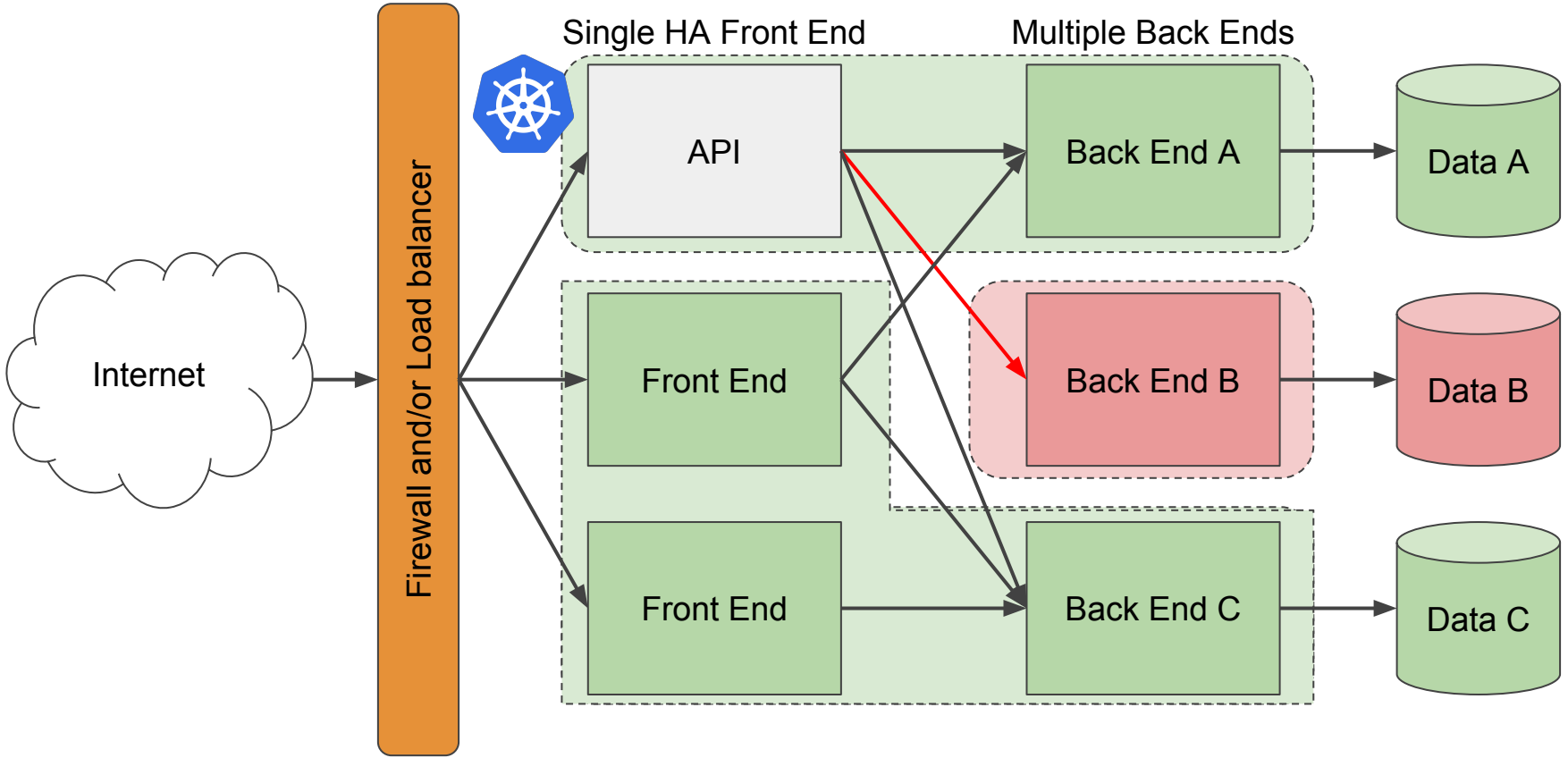
```
tolerations:
- key: "key"
  operator: "Equal"
  value: "value"
  effect: "NoSchedule"
```

```
tolerations:
- key: "key"
  operator: "Exists"
  effect: "NoSchedule"
```




```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', 'echo Hello Kubernetes! && sleep 3600']
```

Pod-Pod Network Communication



Pod-Pod Network Restrictions

There are multiple ways to handle this, but none of which are natively built into Kubernetes.

NetworkPolicy: with a supporting CNI layer

Cilium - https://cilium.io/ (also includes layer 7 security controls)

Calico - https://www.projectcalico.org/

Weave.net - https://www.weave.works/oss/net/

CNI Networking layers are difficult to replace, you will probably want to work with the vendor to make sure you don't run into any issues

Pod-Pod Network Restrictions

Service Mesh:

Linkerd - https://linkerd.io/

Istio - https://istio.io/

Aspen Mesh - https://aspenmesh.io/

Envoy - https://www.envoyproxy.io/

Service Meshes aren't always a "drop in" solution

They do come with more than just network regulations:

- TLS everywhere
- Pod failover
- Lots of metrics
- etc ..

Pod-Pod Network Restrictions

Cloud Native Firewall:

https://www.twistlock.com/

https://www.aquasec.com/

Deployable with containers and `DaemonSets`

Virus Protection

Requirement 5: Use and regularly update anti-virus software or programs

Traditional virus protection with Cloud Native

Antivi

Additional su

From the Docker web si

When antivirus software

One way to reduce these

to the antivirus's exclusi

ers, or volumes are not c

schedule a recurring tas

antivirus, s

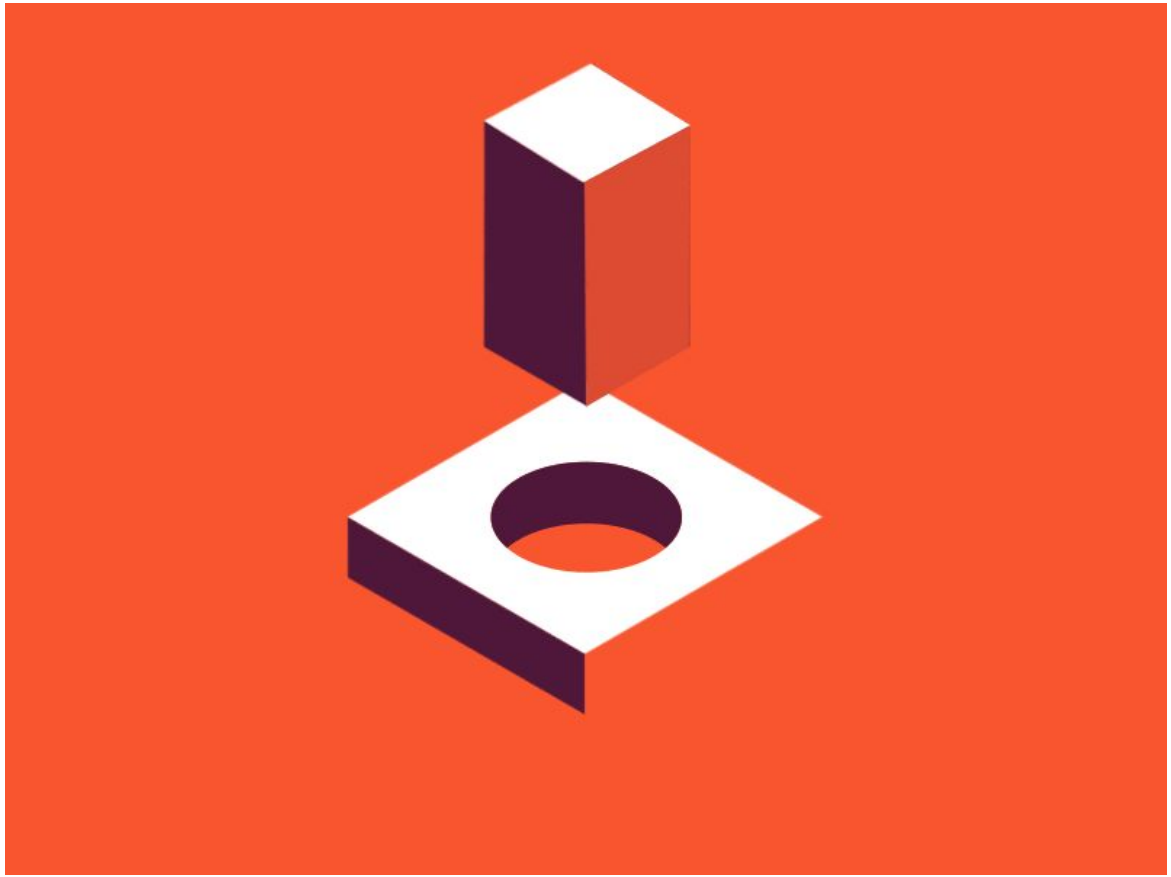
nds to hang.

ata on Windows Server)

ritable layers of contain-

ing, you may want to

Traditional virus protection with Cloud Native



Cloud Native Virus Protection

Virus Protection in Containers

(1) static (during build)
(2) dynamic (during runtime) scanning are a must!!!!

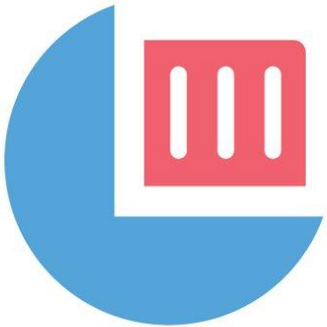
There are a lot of options here: http://lmgtfy.com/?q=container+scanning

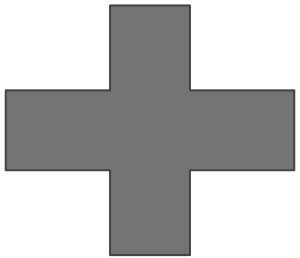
- CoreOS Claire: https://github.com/coreos/clair
- Docker Bench Security: https://github.com/docker/docker-bench-security
- Qualys: https://www.qualys.com/solutions/devsecops/
- Anchore: https://anchore.com/
- Twistlock: https://twistlock.com/
- Sysdig: https://sysdig.com/products/secure/

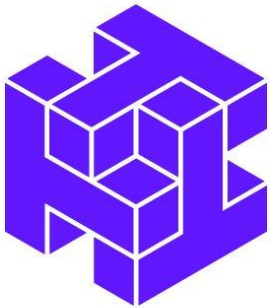
Most of these are platforms help protect both areas

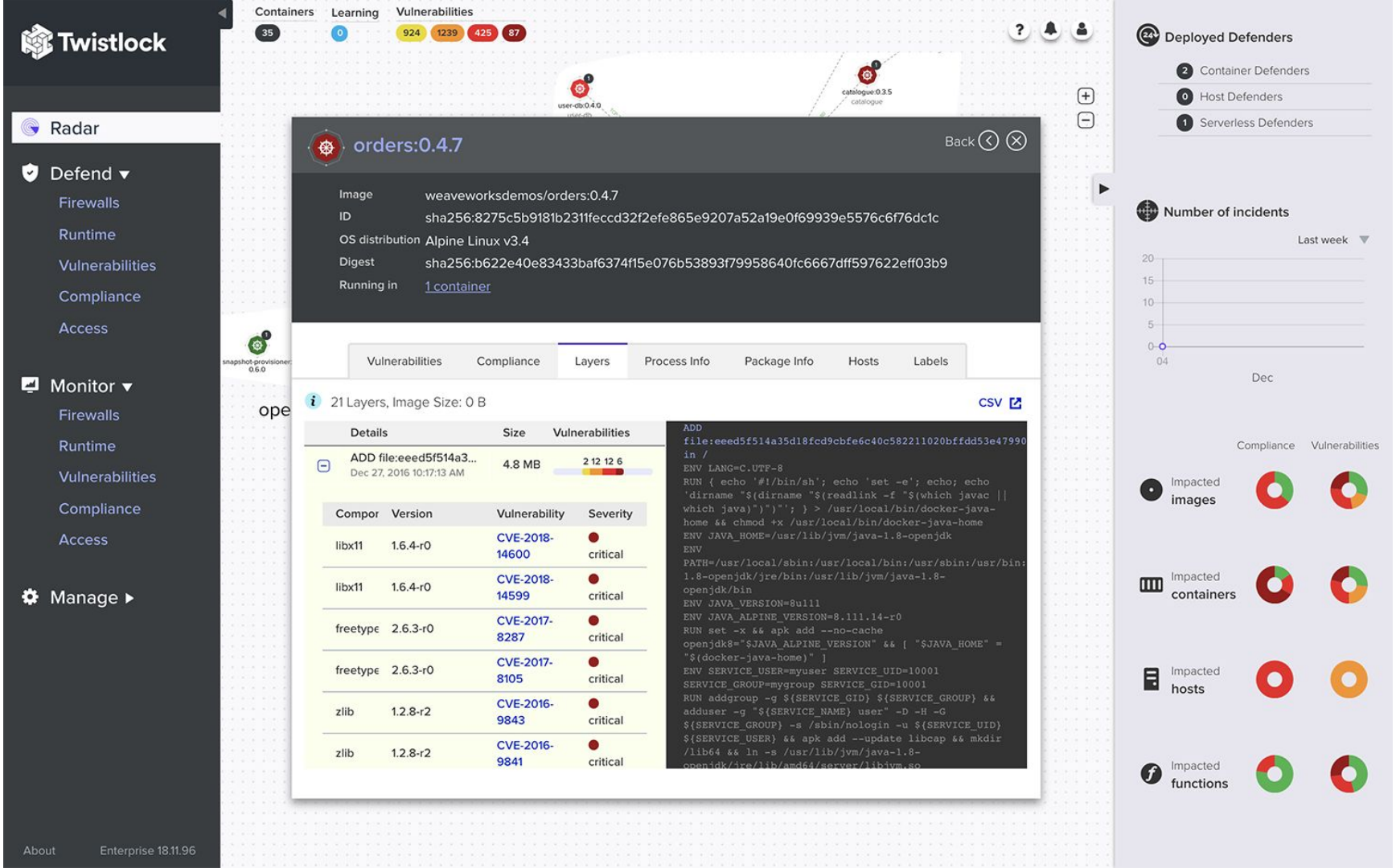
Virus Protection on Node

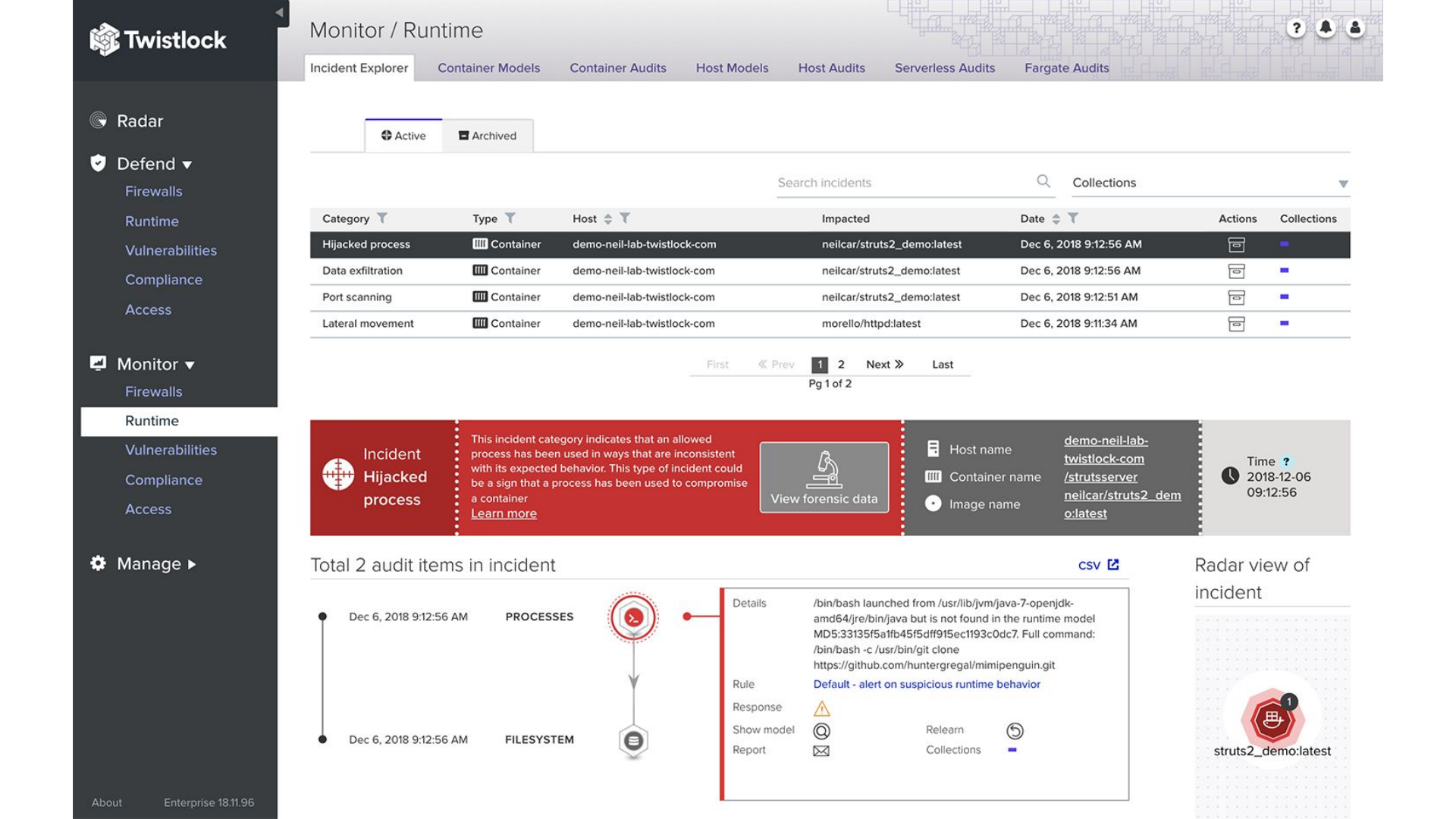
This was pretty easy to achieve with Container Linux + container security platform (from previous slide), for us it was Twistlock








Twistlock
Containers 35
Learning 0
Vulnerabilities 924 1239 425 87
Radar
Defend
Firewalls
Runtime
Vulnerabilities
Compliance
Access
Monitor
Firewalls
Runtime
Vulnerabilities
Compliance
Access
Manage
About
Enterprise 18.11.96
user-db:0.4.0
catalogue:0.3.5
catalogue
orders:0.4.7
Back
Image weaveworksdemos/orders:0.4.7
ID sha256:8275c5b9181b2311feccd32f2efe865e9207a52a19e0f69939e5576c6f76dc1c
OS distribution Alpine Linux v3.4
Digest sha256:b622e40e83433baf6374f15e076b53893f79958640fc6667dff597622eff03b9
Running in 1 container
snapshot-provisioner:0.6.0
Vulnerabilities
Compliance
Layers
Process Info
Package Info
Hosts
Labels
21 Layers, Image Size: 0 B
CSV
Details
Size
Vulnerabilities
ADD file:eeed5f514a3...
Dec 27, 2016 10:17:13 AM
4.8 MB
2 12 12 6
Compor
Version
Vulnerability
Severity
libx11 1.6.4-r0 CVE-2018-14600 critical
libx11 1.6.4-r0 CVE-2018-14599 critical
freetype 2.6.3-r0 CVE-2017-8287 critical
freetype 2.6.3-r0 CVE-2017-8105 critical
zlib 1.2.8-r2 CVE-2016-9843 critical
zlib 1.2.8-r2 CVE-2016-9841 critical
ADD file:eeed5f514a35d18fcd9cbfe6c40c582211020bffdd53e47990 in /
ENV LANG=C.UTF-8
RUN { echo '#!/bin/sh'; echo 'set -e'; echo; echo 'dirname "$(dirname "$(readlink -f "$(which javac || which java)")")"'; } > /usr/local/bin/docker-java-home && chmod +x /usr/local/bin/docker-java-home
ENV JAVA_HOME=/usr/lib/jvm/java-1.8-openjdk
ENV PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/lib/jvm/java-1.8-openjdk/jre/bin:/usr/lib/jvm/java-1.8-openjdk/bin
ENV JAVA_VERSION=8u111
ENV JAVA_ALPINE_VERSION=8.111.14-r0
RUN set -x && apk add --no-cache openjdk8="$JAVA_ALPINE_VERSION" && ["$JAVA_HOME" = "$(docker-java-home)"]
ENV SERVICE_USER=myuser SERVICE_UID=10001 SERVICE_GROUP=mygroup SERVICE_GID=10001
RUN addgroup -g ${SERVICE_GID} ${SERVICE_GROUP} && adduser -g "${SERVICE_NAME} user" -D -H -G ${SERVICE_GROUP} -s /sbin/nologin -u ${SERVICE_UID} ${SERVICE_USER} && apk add --update libcap && mkdir /lib64 && ln -s /usr/lib/jvm/java-1.8-openjdk/jre/lib/amd64/server/libjvm.so
Deployed Defenders
2 Container Defenders
0 Host Defenders
1 Serverless Defenders
Number of incidents
Last week
20
15
10
5
0
04
Dec
Compliance
Vulnerabilities
Impacted images
Impacted containers
Impacted hosts
Impacted functions


Twistlock
Monitor / Runtime
Incident Explorer
Container Models
Container Audits
Host Models
Host Audits
Serverless Audits
Fargate Audits
Radar
Defend
Firewalls
Runtime
Vulnerabilities
Compliance
Access
Monitor
Firewalls
Runtime
Vulnerabilities
Compliance
Access
Manage
About
Enterprise 18.11.96
Active
Archived
Search incidents
Collections
Category
Type
Host
Impacted
Date
Actions
Collections
Hijacked process
Container
demo-neil-lab-twistlock-com
neilcar/struts2_demo:latest
Dec 6, 2018 9:12:56 AM
Data exfiltration
Container
demo-neil-lab-twistlock-com
neilcar/struts2_demo:latest
Dec 6, 2018 9:12:56 AM
Port scanning
Container
demo-neil-lab-twistlock-com
neilcar/struts2_demo:latest
Dec 6, 2018 9:12:51 AM
Lateral movement
Container
demo-neil-lab-twistlock-com
morello/httpd:latest
Dec 6, 2018 9:11:34 AM
First
Prev
1
2
Next
Last
Pg 1 of 2
Incident
Hijacked process
This incident category indicates that an allowed process has been used in ways that are inconsistent with its expected behavior. This type of incident could be a sign that a process has been used to compromise a container
Learn more
View forensic data
Host name
demo-neil-lab-twistlock-com
Container name
/strutsserver
Image name
neilcar/struts2_demo:latest
Time
2018-12-06
09:12:56
Total 2 audit items in incident
CSV
Dec 6, 2018 9:12:56 AM
PROCESSES
Dec 6, 2018 9:12:56 AM
FILESYSTEM
Details
/bin/bash launched from /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java but is not found in the runtime model MD5:33135f5a1fb45f5dff915ec1193c0dc7. Full command: /bin/bash -c /usr/bin/git clone https://github.com/huntergregal/mimipenguin.git
Rule
Default - alert on suspicious runtime behavior
Response
Show model
Relearn
Report
Collections
Radar view of incident
1
struts2_demo:latest

Authorization & Authentication

Requirement 7: Restrict access to cardholder data by business need-to-know

Requirement 8: Assign a unique ID to each person with computer access

Kubernetes RBAC FTW!

https://kubernetes.io/docs/reference/access-authn-authz/rbac/

```yaml
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""] # "" indicates the core API group
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

Google Authentication + Kubernetes Open ID = https://github.com/micahhausler/k8s-oidc-helper

```yaml
# This role binding allows "jane" to read pods in the "default" namespace.
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: User
  name: jane # Name is case sensitive
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role #this must be Role or ClusterRole
  name: pod-reader # this must match the name of the Role or ClusterRole you wish to bind to
  apiGroup: rbac.authorization.k8s.io
```

Track and Monitor

Requirement 10: Track and monitor all access to network resources and cardholder data

3 Core areas to address

Auditing

Logging

Monitoring

Auditing - Know when changes happen

Kubernetes Audit:
https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

- Fine grain control over what gets logged
- Multiple backend storage locations

Logging & Monitoring

Things to consider:

- Easy for developers (they despise things that don't relate to BBQ'ing or the latest topic of functional vs OO programming)
  - ✕ Additional Code dependencies
  - ✓ Something they are already familiar with … their code

- Easy for operations (they despise things that don't relate to Star Wars or how to eject their phone out a window at 3AM when they get paged)
  - ✕ Adding additional process to workflow
  - ✓ Cloud Native - DaemonSets, CRDs, etc …

Vulnerabilities

Requirement 11. Regularly test security systems and processes

Pipeline Practices

Things that have helped us stay secure

Static Container Image Scanning

```
No CA cert was specified, using insecure connection
Vulnerabilities
---------------
Image    ID    CVE    Package    Version    Severity    Status
-----    --    ---    -------    -------    --------    ------
Vulnerability threshold check results: PASS

Compliance
----------
Image                                          ID                Severity    Description
-----                                          --                --------    -----------
registry.nav.engineering/goldmaster/alpine:latest    921e67f2a023fca3    high        Image should be created with a user
```

Set a threshold on the vulnerability/compliance scanning to fail builds if surpassed.

Scan anything that builds and pushes into your container registry.

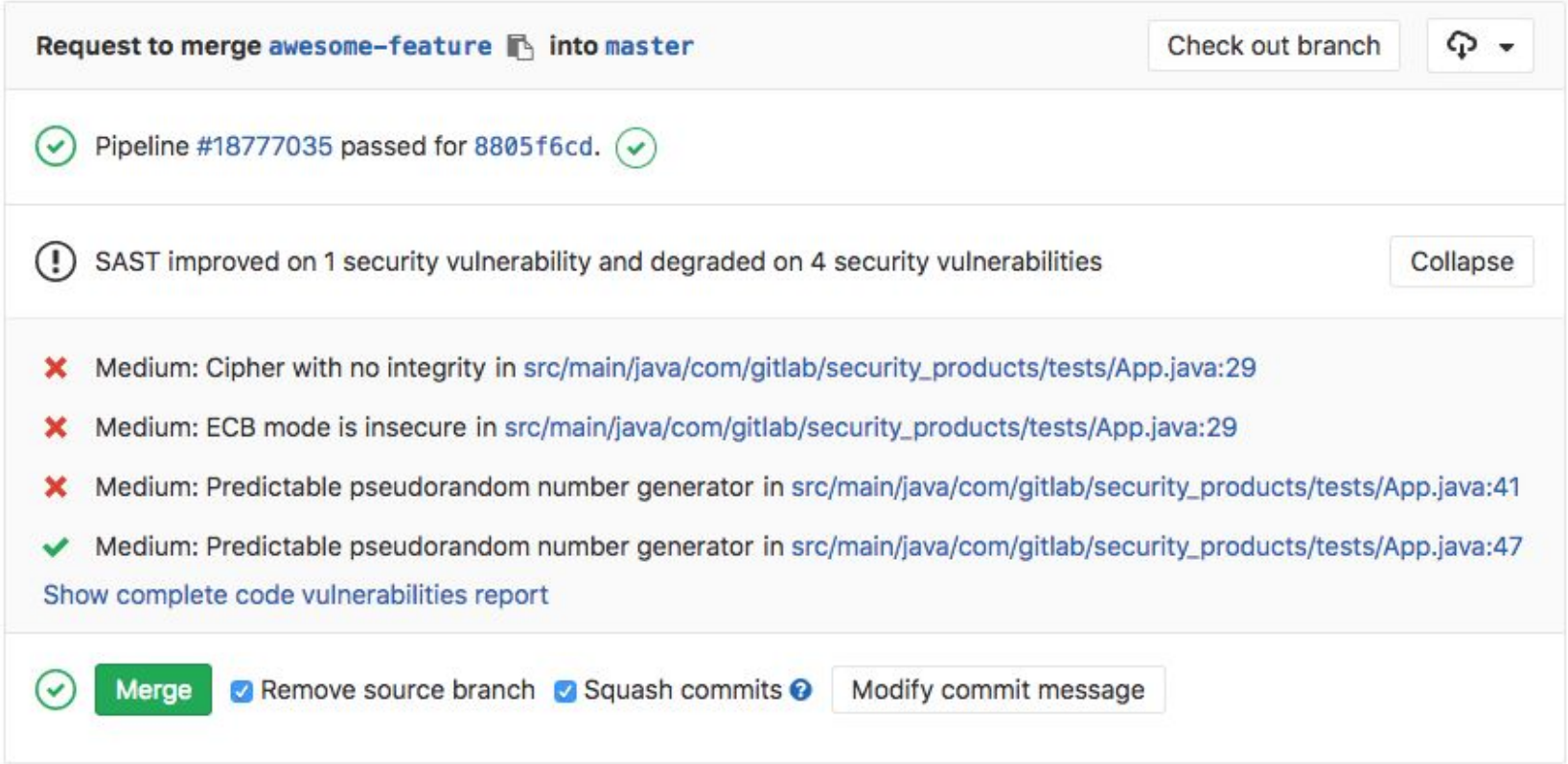
"Gold master" Base Image

```
1  FROM alpine:3.8
2
3  RUN apk --no-cache upgrade
4
5  RUN apk --no-cache add \
6        curl \
7        ca-certificates \
8        bash \
9        shadow \
10       jq
11
12 COPY *.crt /usr/local/share/ca-certificates/
13
14 RUN update-ca-certificates
15
16 COPY entrypoint.sh /entrypoint.sh
17
18 COPY alpine.gitlog /alpine.gitlog
19
20 ENTRYPOINT ["/entrypoint.sh"]
```

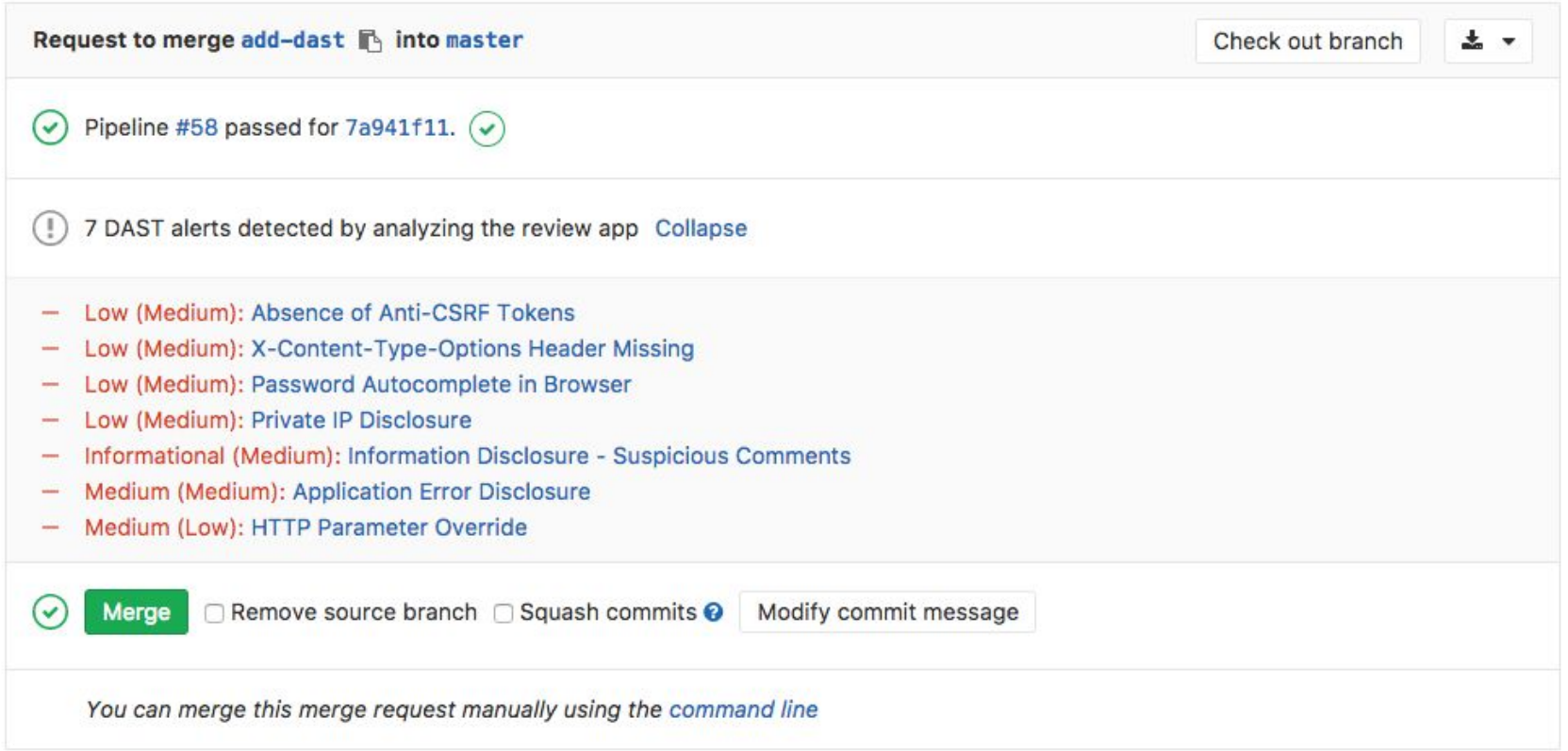
Base all other projects off of this image (as much as possible anyway).

This helps immensely when trying to push updates and vulnerability fixes out.

SAST - Static Application Security Testing



DAST - Dynamic Application Security Testing



Nav

Questions?