STEVE GIGUERE - CLOUD NATIVE SECURITY ADVOCATE AND ...

# LAY8R CAK3

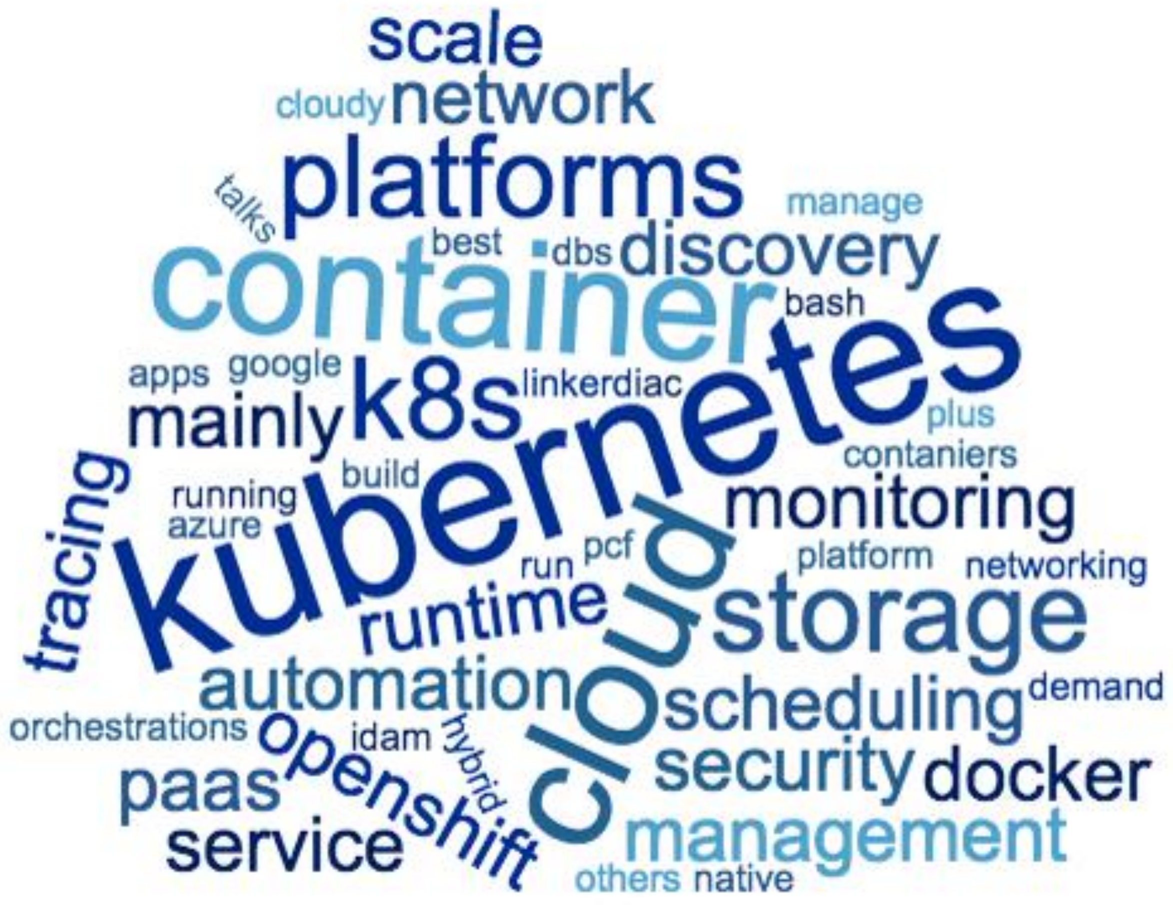
Moving security to deterministic over probabilistic

# INTRO

- Steve Giguere - StackRox Director of Tech Stuff EMEA
  - Twit: @_SteveGiguere_
  - https://www.linkedin.com/in/stevegiguere
- Podcast: The Continuous Security Podcast
  - https://cosecast.com
- Twitch: KubeNative Security
  - https://www.twitch.tv/kubenativesecurity
- Beer
  - Untappd: stevegiguere
  - Youtube: BeerNativeTV

# KUBERNETES! (FOR SECURITY)

- <insert blatant nautical theme image>
- <add photo of shipping container>
- <add word cloud>
- <talk about change>



Stolen from storageOS

# SECURITY (A.K.A. INFOSEC)

- Noisy vs Dangerous
- Primitive vs Expensive
- Reactive vs proactive
- Probability versus certainty
- Risk = Likelihood * Impact

# EVERYTHING AS CODE

# EVERYTHING AS CODE

- WE DID IT!...?
- STANDARDS
- SO MANY STANDARDS!
- CLOUD FORMATION, TERRAFORM, PULUMI, CLOUDIFY, CLUSTERAPI, YAML, CDK8s, JSON, HELM, ANSIBLE, XML

# WHY CODE?

- IMPERATIVE COMMANDS = MISTAKES = NON-REPEATABLE RESULTS = NO CHANGE CONTROL
- ...and many other reasons... like HUMANS

AMERICA GREAT!
2020
For President Of The United States
BILLA ONG
COVID-19
IS A LIE
DARWIN AWARDS
Never a shortage of candidates

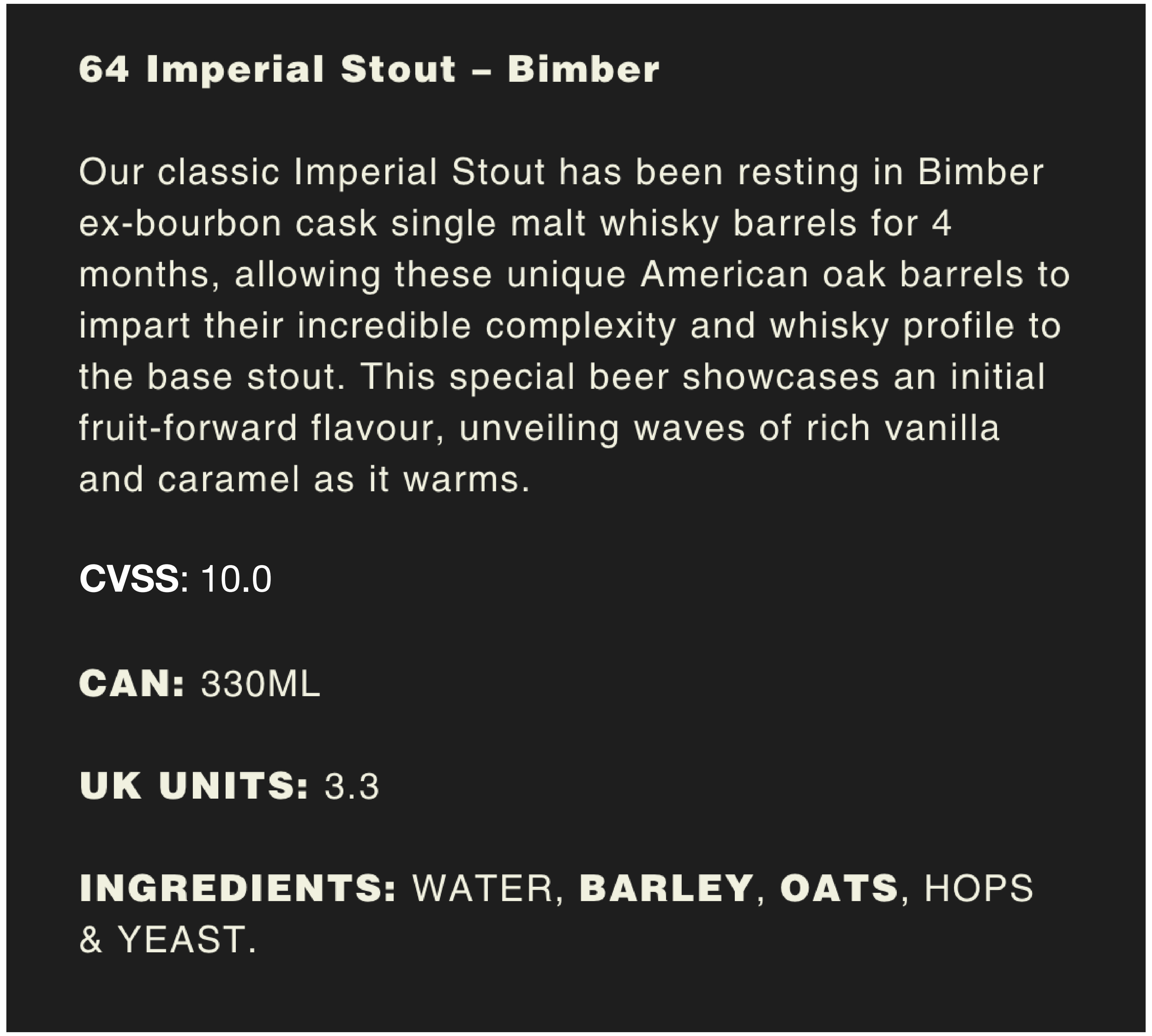
# CHANGE THE BATTLEFIELD

- Reduce Likelihood
- Employ checks earlier and often
- Scan for IaC misconfiguration as code

Use IaC context to our advantage when determining risk

- Reduce Impact
- Leverage Chaos Engineering
- Cattle versus Pets

# THE IMPORTANCE OF CONTEXT



**64 Imperial Stout – Bimber**

Our classic Imperial Stout has been resting in Bimber ex-bourbon cask single malt whisky barrels for 4 months, allowing these unique American oak barrels to impart their incredible complexity and whisky profile to the base stout. This special beer showcases an initial fruit-forward flavour, unveiling waves of rich vanilla and caramel as it warms.

**CVSS**: 10.0

**CAN:** 330ML

**UK UNITS:** 3.3

**INGREDIENTS:** WATER, **BARLEY**, **OATS**, HOPS & YEAST.

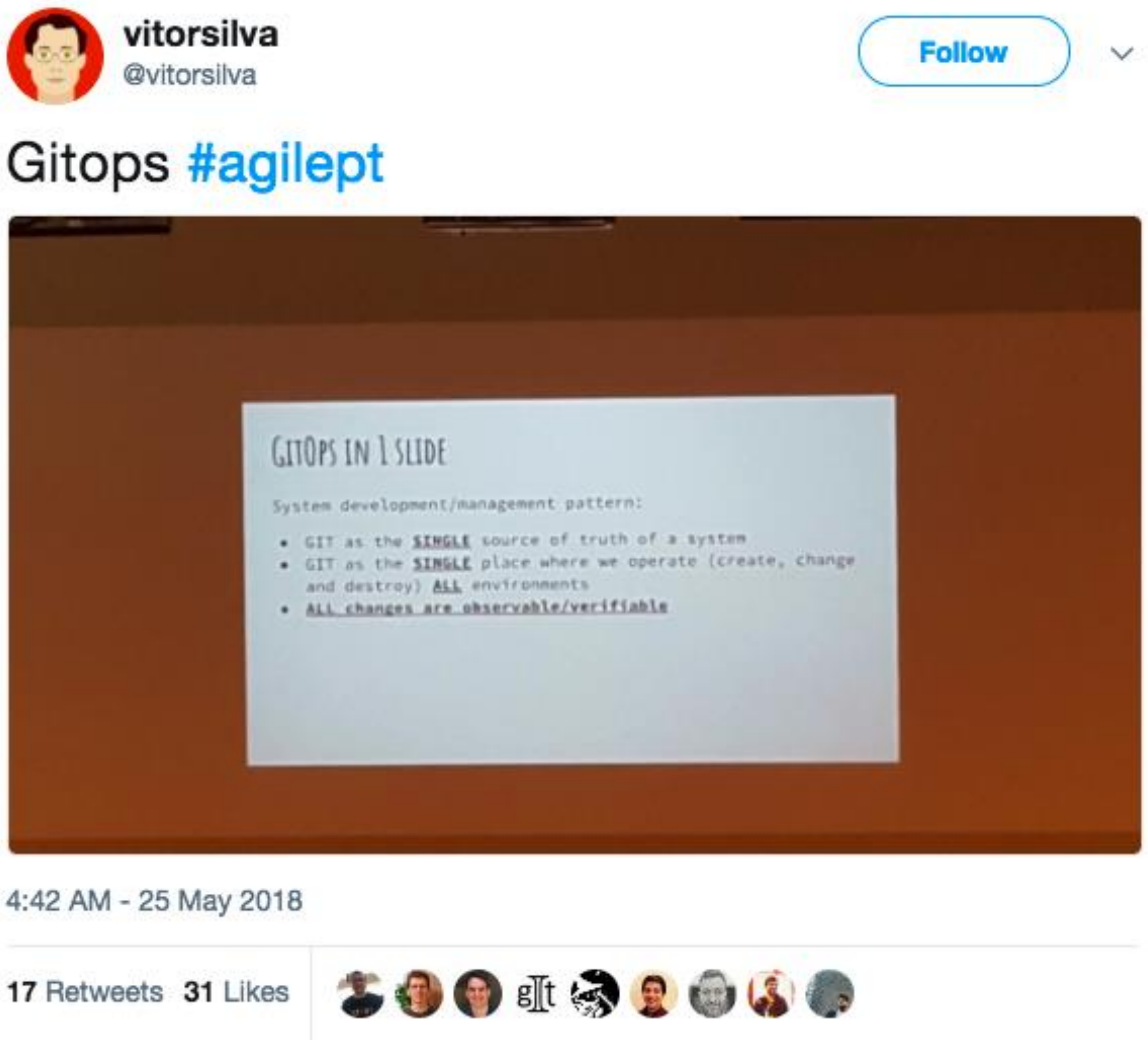
**evening-drinks.yaml**

```
kind: deployment
meta-data:
  name: bimber-ba
  type: imperial-stout
  labels:
    dinner: salad
spec:
  replicas: 3
  service: amazon-prime
  - tv:
    show: american-gods
      duration: 45m
    env:
    - name: dry-january
      value: yes
```

# GET STARTED WITH GITOPS

- Making Code Great Again
- “Opinionated and prescriptive best practices”
    - Weaveworks

- GIT as the SINGLE source of truth
- GIT as the SINGLE place where we operate
- OBSERVABLE and VERIFIABLE



Gitops #agilept

GITOPS IN 1 SLIDE

System development/management pattern:

- GIT as the **SINGLE** source of truth of a system
- GIT as the **SINGLE** place where we operate (create, change and destroy) **ALL** environments
- **ALL changes are observable/verifiable**

4:42 AM - 25 May 2018

17 Retweets 31 Likes

Stole from Vitor Silva via Weaveworks

# THE LAYERS

- LAYER 0 - THE CLOUD
- LAYER 1 - THE PIPELINE
- LAYER 2 - THE APPLICATION
- LAYER 3 - THE APPLICATION’S FRIENDS
- LAYER 4 - THE IMAGE
- LAYER 5 - THE DEPLOYMENT (CONTEXT)
- LAYER 6 - THE RUNTIME (THE ICING)

# LAYER 0: SECURE THE BASE

- What
  - IaC Scanners for pre-flight checks
    - checkov by Bridgecrew, kics by CheckMarX, terrascan by Accurics
  - CSPM (Cloud Security Posture Management) for maintaining a seCure state
    - Free: OpenCSPM or Commercial:  Accurics, Wave (Aqua), Dome9
- Why
  - Humans are creating the code so verification prior to use should be standard

terrascan
by accurics

OpenCSPM

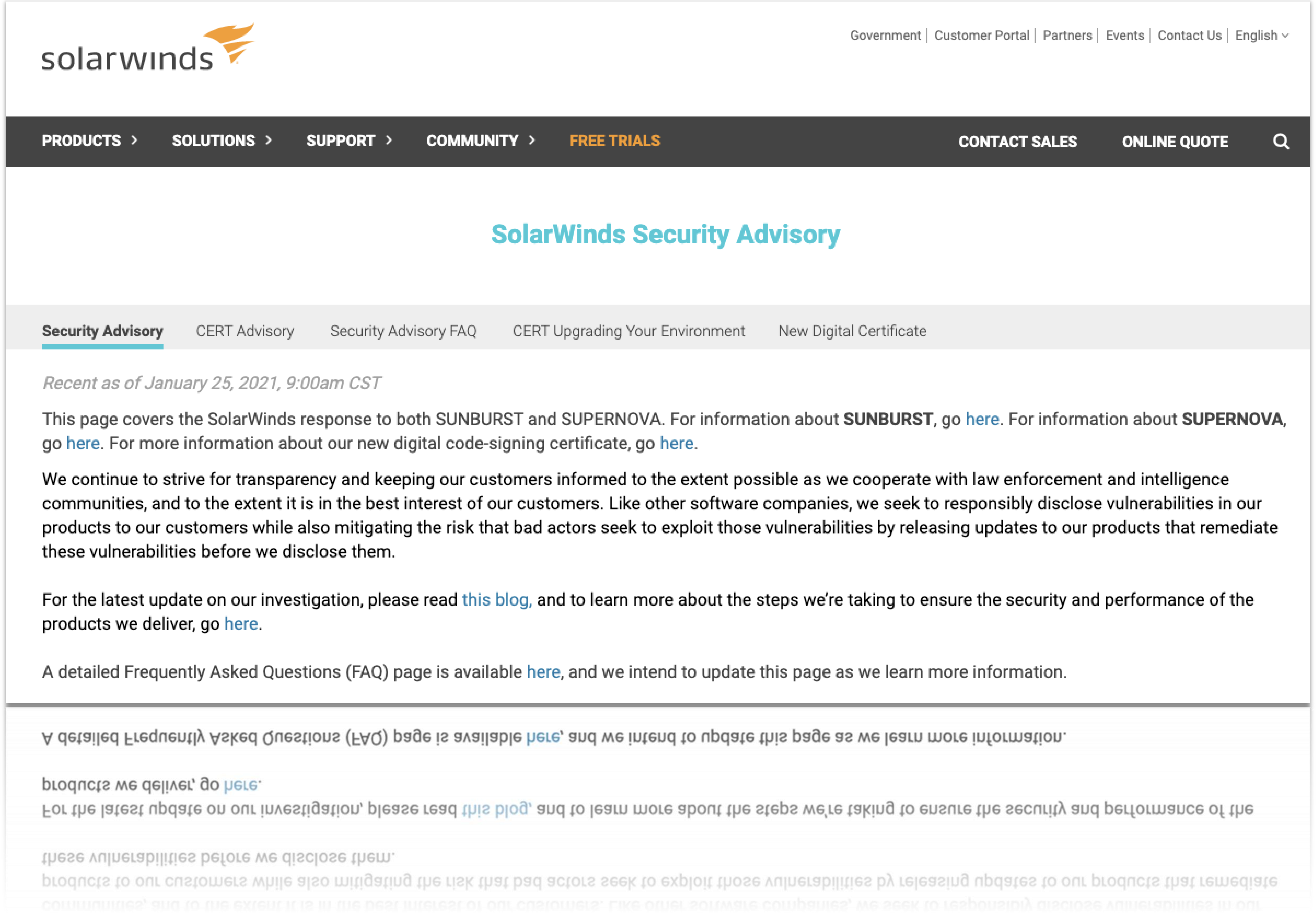
```
        Description    :       Ensure that your RDS database has IAM Authentication enabled.
        File           :       ec2-database.tf
        Line           :       19
        Severity       :       HIGH
        -----------------------------------------------------------------------

        Description    :       http port open to internet
        File           :       security.tf
        Line           :       25
        Severity       :       HIGH
        -----------------------------------------------------------------------

        Description    :       EC2 instances should disable IMDS or require IMDSv2 as this can be related to the weaponization phase of kill chain
        File           :       ec2-database.tf
        Line           :       34
        Severity       :       MEDIUM
        -----------------------------------------------------------------------

        Description    :       Ensure that your RDS database instances encrypt the underlying storage. Encrypted RDS instances use the industry standard AES-2
56 encryption algorithm to encrypt data on the server that hosts RDS DB instances. After data is encrypted, RDS handles authentication of access and descriptio
n of data transparently with minimal impact on performance.
        File           :       ec2-database.tf
        Line           :       19
        Severity       :       HIGH
        -----------------------------------------------------------------------


Scan Summary -

        File/Folder         :   /Users/stephengiguere/code/terraform-aws-wordpress
        IaC Type            :   terraform
        Scanned At          :   2021-02-17 11:16:13.214249 +0000 UTC
        Policies Validated  :   562
        Violated Policies   :   5
        Low                 :   0
        Medium              :   2
        High                :   3
stephengiguere@Stephens-MacBook-Pro terraform-aws-wordpress %
```

# LAYER 0: SECURE THE BASE

- Pros
  - Infrastructure as Code, controlled and observed change workflow (gitops)
  - Chaos Engineering friendly
  - Reduction of dependence on tribal knowledge
- Cons
  - IaC from scratch (amalgamation of StackOverflow and Github) with insecure defaults
  - Template squatting (eg WordPress) with changes to trust boundaries
  - Can age over time
  - Less attentive to updates

# LAYER 1: SECURING THE PIPELINE

- What
  - Software supply chain integrity/provenance
- Why
  - Solarwinds
  - Development CI/CD systems often have the keys to the kingdom



## SolarWinds Security Advisory

*Recent as of January 25, 2021, 9:00am CST*

This page covers the SolarWinds response to both SUNBURST and SUPERNOVA. For information about **SUNBURST**, go here. For information about **SUPERNOVA**, go here. For more information about our new digital code-signing certificate, go here.

We continue to strive for transparency and keeping our customers informed to the extent possible as we cooperate with law enforcement and intelligence communities, and to the extent it is in the best interest of our customers. Like other software companies, we seek to responsibly disclose vulnerabilities in our products to our customers while also mitigating the risk that bad actors seek to exploit those vulnerabilities by releasing updates to our products that remediate these vulnerabilities before we disclose them.

For the latest update on our investigation, please read this blog, and to learn more about the steps we're taking to ensure the security and performance of the products we deliver, go here.

A detailed Frequently Asked Questions (FAQ) page is available here, and we intend to update this page as we learn more information.

# LAYER 1: SECURING THE PIPELINE

- Pros
  - Ensure your code is still your code!  E.g. InToto, Rekor, Grafaes
  - Prevent MITM supply chain attacks
- Cons
  - Difficult to deploy at scale
  - No commercial solutions (TIKO)

in-toto

https://in-toto.io

# LAYER 2: SECURING THE CODE

## (THE ACTUAL APPLICATION SOURCE CODE)

- What
  - SAST (Static Application Security Testing) tools in the code pipeline
  - IDE SAST
- Why
  - Humans are involved
  - 1 in every 1000 lines of code contains a bug or insecurity

# LAYER 2: SECURING THE CODE

## (THE ACTUAL APPLICATION SOURCE CODE)

- Pros
  - Many open source and commercial offerings (Guardrails, Coverity, CheckMarX)
  - Cost Benefits.  Shift Left vs Pen Testing
  - Traverses all (including untested) code paths
- Cons
  - Slow and potentially disruptive
  - False positives
  - Implementation difficult:
  - Tech stack in the hands of developers / security choices are not
  - Few good FOSS IDE integrated (eg.  EsLint)
  - Limited reach in next gen languages (Golang : gosec, Rust: clippy)

Repositories | GuardRails.io
Introduction · GuardRails
dashboard.guardrails.io/gh/eurogig
Apps
Inbox
Calendar
K8S-BMKS
StackRox
How-To K8s
RPi
CoseCast
Thought Leadership
License Server | C...
AWS Event
Security Tools
Imported
eurogig
Steve G
Repositories
Vulnerabilities
Findings
Scans
Insights
Settings
Documentation
Help
Organization
eurogig
Manage Repositories
Filter by repository nan
Private: All
Languages: All
Enabled: All

# Repositories

| Repository Name ↑ | Language | Last Scan | Vulnerabilities | Enabled |
|---|---|---|---|---|
| aqua-py A fantastic Python 3 client for Aqua's CSP Platf... | Python | -- | Not analyzed | ○ |
| black-duck-radar | JavaScript | -- | Not analyzed | ○ |
| docker-minimal-nginx Minimal docker image for nginx ... | Dockerfile | -- | Not analyzed | ○ |
| docker-vulnerable-dvwa Damn Vulnerable Web Appli... | PHP | -- | Not analyzed | ○ |
| gin Gin is a HTTP web framework written in Go (Golang). ... | | -- | Not analyzed | ○ |
| govwa | | -- | Not analyzed | ○ |
| hands-on-trivy-to-tracee A hands on guided lesson f... | Python | -- | Not analyzed | ○ |
| jenkins-docker Docker image for jenkins with docker su... | Dockerfile | -- | Not analyzed | ○ |
| juice-shop OWASP Juice Shop: Probably the most mod... | JavaScript | 17 Feb 2021 | 6 ↑ 100% Increase | ● |
| k3s Deploy Rancher on DigitalOcean | | 10 Dec 2020 | ✓ | ● |

# LAYER 3: SECURING THE (OSS) SUPPLY CHAIN

- What
  - SCA (Software Composition Analysis) / Dependency Checkers
    - FOSS : OWASP dependency checker / npm
    - Commercial: Black Duck / Sonatype
- Why
  - Because most (80%ish) software is open source
  - Open source vulnerabilities are known to the bad guys.  Even the script kids.

# LAYER 3: SECURING THE (OSS) SUPPLY CHAIN

- Pros
  - Finds known vulnerabilities in dependencies!  Woop!
  - Can locate low hanging fruit in security vulnerabilities
- Cons
  - Difficult to prioritise
    - Are the dependencies used and in what context.  Is it real risk?
  - Noisy (15000 CVES disclosed per year)
    - False positives.

# LAYER 4: SECURING THE IMAGE

- What
  - Find known vulnerabilities in base image AND dependencies (via package managers a la SCA)
  - Check for Dockerfile best practices
    - Eg.  Use ADD instead of COPY  /  Run as a non-root user /  many more
- Why
  - Defaults can be dangerously insecure (e.g. default user as root)
  - Images can introduce user space OS dependencies with critical vulnerabilities

# LAYER 4: SECURING THE IMAGE

- Pros
  - Finds known CVEs in CI and developer desktop
  - Teaches best practice
  - Does some SCA as well
  - Plenty of open source free tools e.g. Clair, Trivy, Hadolint
- Cons
  - Can become security theatre
  - Further confuses the vulnerability management debt

```
|                  |                     |          |                           |               | -->avd.aquasec.com/nvd/cve-2019-19882                         |
+                  +---------------------+          +                           +---------------+---------------------------------------------------------------+
|                  | TEMP-0628843-DBAD28 |          |                           |               | -->security-tracker.debian.org/tracker/TEMP-0628843-DBAD28    |
+------------------+---------------------+----------+---------------------------+---------------+---------------------------------------------------------------+
| perl-base        | CVE-2011-4116       |          | 5.28.1-6+deb10u1          |               | perl: File::Temp insecure                                     |
|                  |                     |          |                           |               | temporary file handling                                       |
|                  |                     |          |                           |               | -->avd.aquasec.com/nvd/cve-2011-4116                          |
+------------------+---------------------+----------+---------------------------+---------------+---------------------------------------------------------------+
| sysvinit-utils   | TEMP-0517018-A83CE6 |          | 2.93-8                    |               | -->security-tracker.debian.org/tracker/TEMP-0517018-A83CE6    |
+------------------+---------------------+----------+---------------------------+---------------+---------------------------------------------------------------+
| tar              | CVE-2005-2541       |          | 1.30+dfsg-6               |               | Tar 1.15.1 does not                                           |
|                  |                     |          |                           |               | properly warn the user when                                   |
|                  |                     |          |                           |               | extracting setuid or...                                       |
|                  |                     |          |                           |               | -->avd.aquasec.com/nvd/cve-2005-2541                          |
+                  +---------------------+          +                           +---------------+---------------------------------------------------------------+
|                  | CVE-2019-9923       |          |                           |               | tar: null-pointer dereference                                 |
|                  |                     |          |                           |               | in pax_decode_header in sparse.c                              |
|                  |                     |          |                           |               | -->avd.aquasec.com/nvd/cve-2019-9923                          |
+                  +---------------------+          +                           +---------------+---------------------------------------------------------------+
|                  | CVE-2021-20193      |          |                           |               | tar: Memory leak in                                           |
|                  |                     |          |                           |               | read_header() in list.c                                       |
|                  |                     |          |                           |               | -->avd.aquasec.com/nvd/cve-2021-20193                         |
+                  +---------------------+          +                           +---------------+---------------------------------------------------------------+
|                  | TEMP-0290435-0B57B5 |          |                           |               | -->security-tracker.debian.org/tracker/TEMP-0290435-0B57B5    |
+------------------+---------------------+----------+---------------------------+---------------+---------------------------------------------------------------+
stephengiguere@Stephens-MacBook-Pro ~ % trivy image nginx:latest | head
2021-02-17T11:39:10.849Z        WARN    You should avoid using the :latest tag as it is cached. You need to specify '--clear-cache' option when :latest image is changed
2021-02-17T11:39:18.574Z        INFO    Detecting Debian vulnerabilities...
2021-02-17T11:39:18.596Z        INFO    Trivy skips scanning programming language libraries because no supported file was detected

nginx:latest (debian 10.8)
==========================
Total: 155 (UNKNOWN: 4, LOW: 108, MEDIUM: 9, HIGH: 33, CRITICAL: 1)

+------------------+---------------------+----------+---------------------------+---------------+---------------------------------------------------------------+
|     LIBRARY      |  VULNERABILITY ID   | SEVERITY |     INSTALLED VERSION     | FIXED VERSION |                             TITLE                             |
stephengiguere@Stephens-MacBook-Pro ~ %
stephengiguere@Stephens-MacBook-Pro ~ %
stephengiguere@Stephens-MacBook-Pro ~ %
stephengiguere@Stephens-MacBook-Pro ~ %
```

# RISK CONTEXT

# LAYER 5: SECURING THE DEPLOYMENT

- What
  - Best Practices for Kubernetes Objects
  - Operational risk / Security risk
- Why
  - Bring essential context to image deployment and vulnerability management
  - Defaults can be dangerously insecure

# LAYER 5: SECURING THE DEPLOYMENT

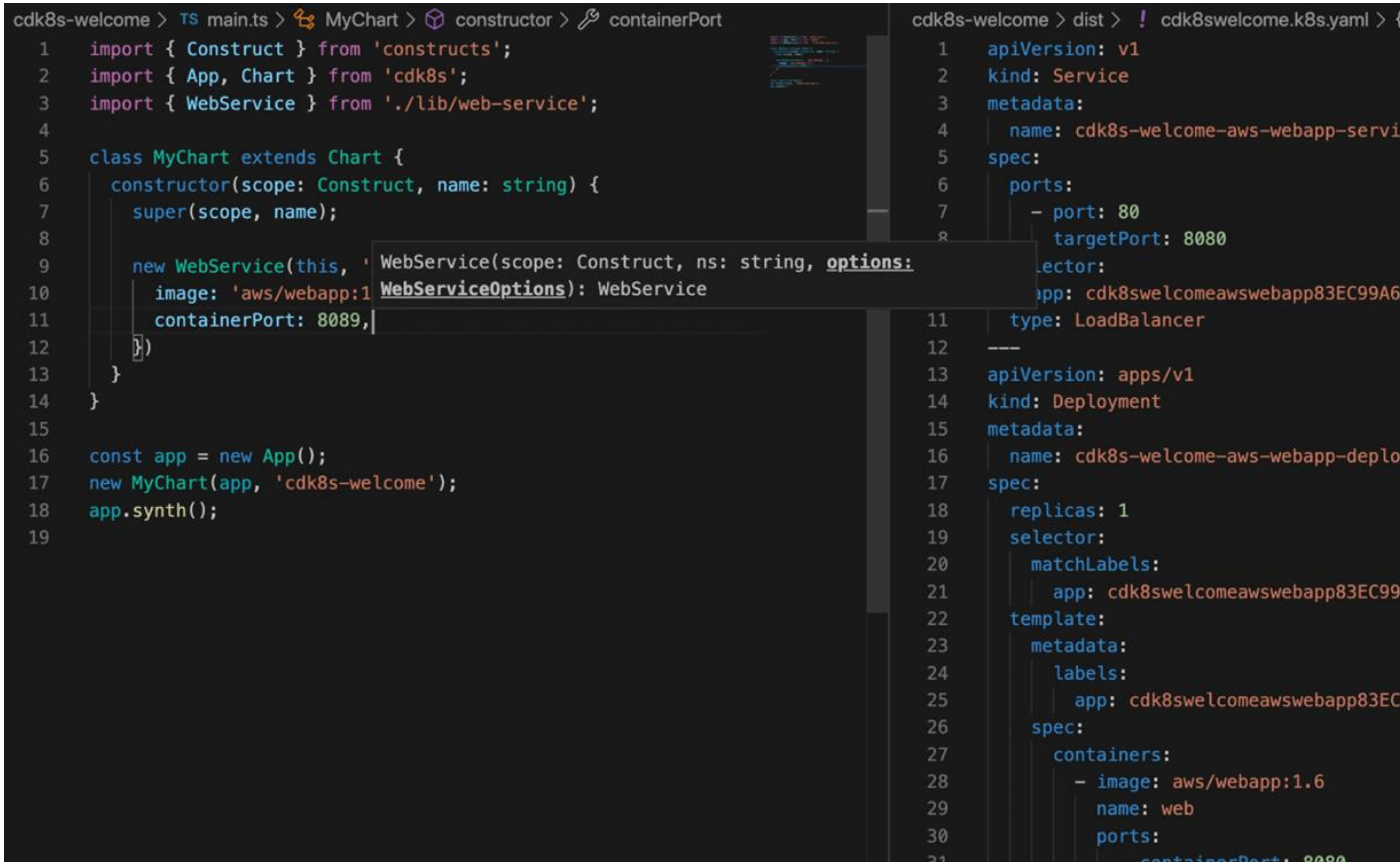
- Pros
  - Many open source tools
  - Kube-linter / kube-score / checkov / kics / kubescan

KubeLinter

```
stephengiguere@Stephens-MacBook-Pro kubernetes-manifests % kube-linter lint emailservice-rushed.yaml
emailservice-rushed.yaml: (object: <no namespace>/emailservice apps/v1, Kind=Deployment) container "server" does not have a read-only root file system (check:
no-read-only-root-fs, remediation: Set readOnlyRootFilesystem to true in your container's securityContext.)

emailservice-rushed.yaml: (object: <no namespace>/emailservice apps/v1, Kind=Deployment) container "server" is not set to runAsNonRoot (check: run-as-non-root,
 remediation: Set runAsUser to a non-zero number, and runAsNonRoot to true, in your pod or container securityContext. See https://kubernetes.io/docs/tasks/conf
igure-pod-container/security-context/ for more details.)

emailservice-rushed.yaml: (object: <no namespace>/emailservice apps/v1, Kind=Deployment) container "server" has cpu request 0 (check: unset-cpu-requirements, r
emediation: Set your container's CPU requests and limits depending on its requirements. See https://kubernetes.io/docs/concepts/configuration/manage-resources-
containers/#requests-and-limits for more details.)

emailservice-rushed.yaml: (object: <no namespace>/emailservice apps/v1, Kind=Deployment) container "server" has cpu limit 0 (check: unset-cpu-requirements, rem
ediation: Set your container's CPU requests and limits depending on its requirements. See https://kubernetes.io/docs/concepts/configuration/manage-resources-co
ntainers/#requests-and-limits for more details.)

emailservice-rushed.yaml: (object: <no namespace>/emailservice apps/v1, Kind=Deployment) container "server" has memory request 0 (check: unset-memory-requireme
nts, remediation: Set your container's memory requests and limits depending on its requirements. See https://kubernetes.io/docs/concepts/configuration/manage-r
esources-containers/#requests-and-limits for more details.)

emailservice-rushed.yaml: (object: <no namespace>/emailservice apps/v1, Kind=Deployment) container "server" has memory limit 0 (check: unset-memory-requirement
s, remediation: Set your container's memory requests and limits depending on its requirements. See https://kubernetes.io/docs/concepts/configuration/manage-res
ources-containers/#requests-and-limits for more details.)

Error: found 6 lint errors
stephengiguere@Stephens-MacBook-Pro kubernetes-manifests %
```

# LAYER 5: SECURING THE DEPLOYMENT
## (CAVEAT)

- Abstractions
    - Eg. CDK8S
- Pros
    - Time to market
- Cons
    - Difficult to secure misconfigurations in generated yaml back to source code



cdk8s-welcome > TS main.ts > MyChart > constructor > containerPort

```
import { Construct } from 'constructs';
import { App, Chart } from 'cdk8s';
import { WebService } from './lib/web-service';

class MyChart extends Chart {
  constructor(scope: Construct, name: string) {
    super(scope, name);

    new WebService(this, '
      image: 'aws/webapp:1
      containerPort: 8089,
    })
  }
}

const app = new App();
new MyChart(app, 'cdk8s-welcome');
app.synth();
```

```
WebService(scope: Construct, ns: string, options:
WebServiceOptions): WebService
```

cdk8s-welcome > dist > ! cdk8swelcome.k8s.yaml >

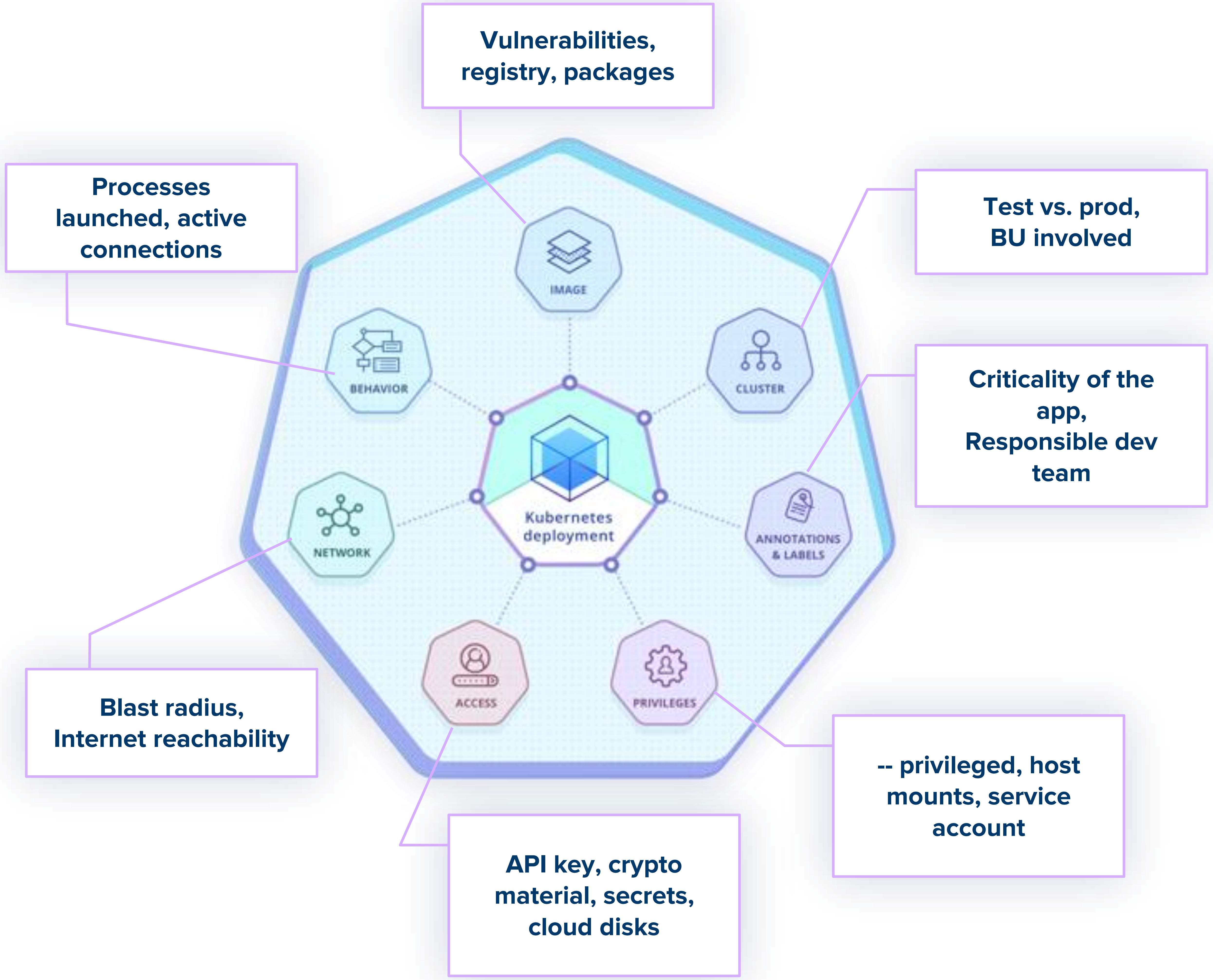
```
apiVersion: v1
kind: Service
metadata:
  name: cdk8s-welcome-aws-webapp-servi
spec:
  ports:
    - port: 80
      targetPort: 8080
   .ector:
     pp: cdk8swelcomeawswebapp83EC99A6
  type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: cdk8s-welcome-aws-webapp-deplo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: cdk8swelcomeawswebapp83EC99
  template:
    metadata:
      labels:
        app: cdk8swelcomeawswebapp83EC
    spec:
      containers:
        - image: aws/webapp:1.6
          name: web
          ports:
```

# LAYER 5: SECURING THE DEPLOYMENT

- Pros
  - Many open source tools
  - Kube-linter / kube-score / checkov / kics
- Cons

**Few (if any) open source tools combine image vulnerabilities with deployment context**

# LAYER 5: THE VALUE OF CONTEXT



Vulnerabilities, registry, packages

Processes launched, active connections

Test vs. prod, BU involved

Criticality of the app, Responsible dev team

Blast radius, Internet reachability

-- privileged, host mounts, service account

API key, crypto material, secrets, cloud disks

# LAYER 5: THE VALUE OF CONTEXT

- Be able to prioritise a container with a CVE with CVSS 9.8
  - in an backend service
  - no external connectivity
  - not running as privileged
  - a recorded baseline of process activity

- Against a container with a CVE with CVSS 7.6
  - in multiple frontend services
  - behind a load balancer
  - exposed 22
  - Complex base image and behaviour
  - Tools present like
    - curl
    - wget
    - nmap

# LAYER 6: THE RUNTIME
## MAINTAIN THE STATE (OF SECURITY!)

- What
  - eBPF
    - Falco - Sysdig, Tracee - Aqua
  - Anomalies as k8s
  - Prevention by admission controllers
  - Security as Policy
    - OPA, Kyverno
- Why
  - We can only find so much in layers 0-5
  - 0 day exploits of new attack vectors

# LAYER 6: THE RUNTIME

## MAINTAIN THE STATE (OF SECURITY!)

- Pros
  - InfoSec people understand EDR and IDS
  - Zero day / anomaly detection
  - Safety net
- Cons
  - Reactionary, probabilistic
  - Labour intensive
  - Expensive
  - Horse bolted, door closed
  - Still required

# KEY TAKEAWAYS

- SHIFT LEFT ( is hard work filled with cons )
  - The more people you need to buy in the easier it needs to be
- SHIFT MIDDLE (or Everywhere)?
  - Simpler checks but more often throughout the pipeline
- CONTEXT is huge advantage
  - Technical debt will be overwhelming without context
- Everything as Code (EaC)
  - Reduces imperative intervention but creates more traditional security challenges
- GitOps + Kubernetes
  - Stateful at rest and runtime
- Declarative = More Deterministic = Less Probabilistic

DETERMINISTIC SECURITY

# KUBERNETES

# THANKS!

StackRox

redhat

- Steve Giguere - StackRox Director of Tech Stuff EMEA
  - Twit: @_SteveGiguere_
  - https://www.linkedin.com/in/stevegiguere
- Podcast: The Continuous Security Podcast
  - https://cosecast.com
- Twitch: KubeNative Security
  - https://www.twitch.tv/kubenativesecurity
- Beer
  - Untappd: stevegiguere
  - Youtube: BeerNativeTV

COSECAST
powered by StackRox