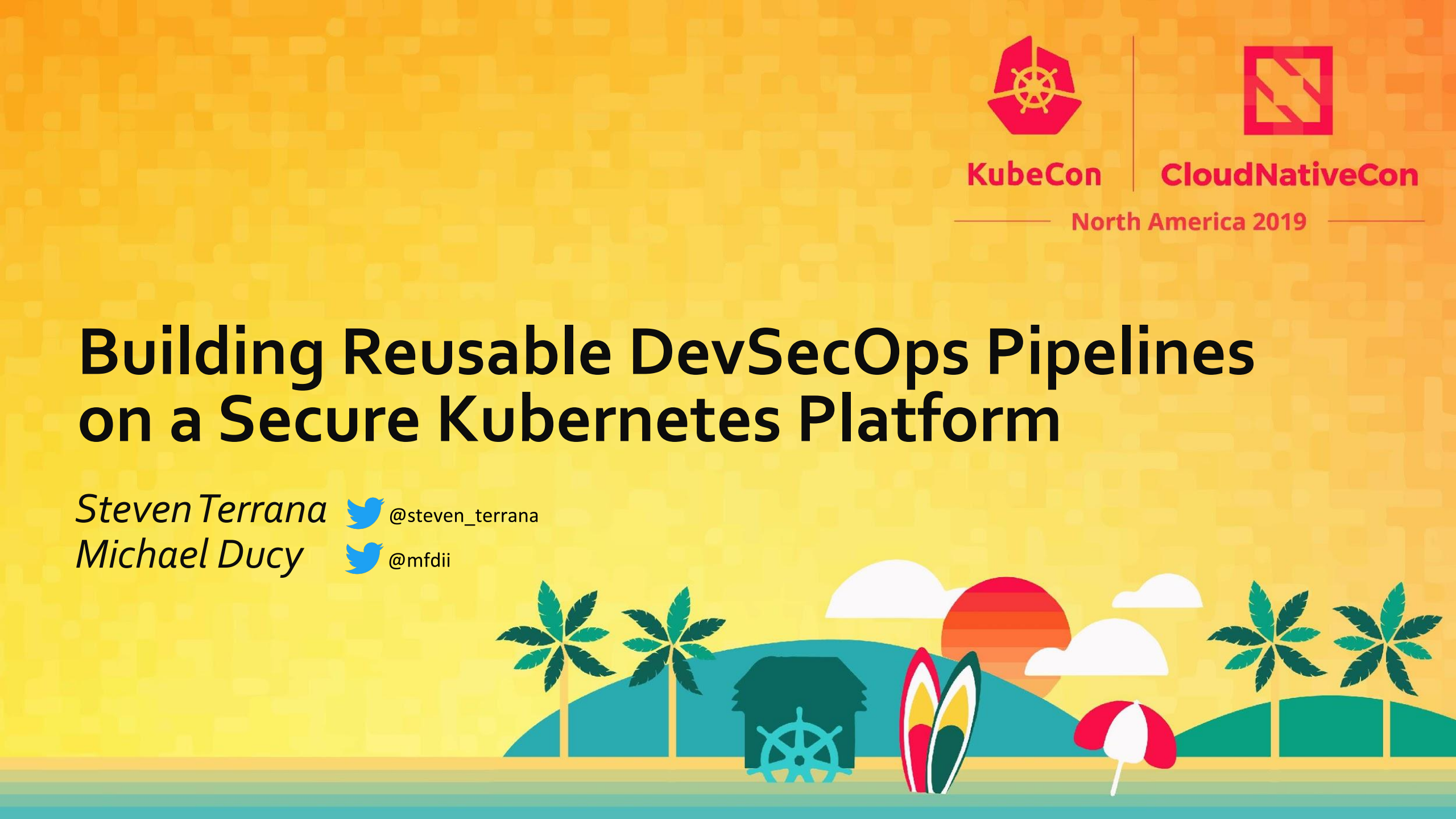

KubeCon
CloudNativeCon
North America 2019



Building Reusable DevSecOps Pipelines on a Secure Kubernetes Platform

Steven Terrana @steven_terrana
Michael Ducy @mfdii

About Us

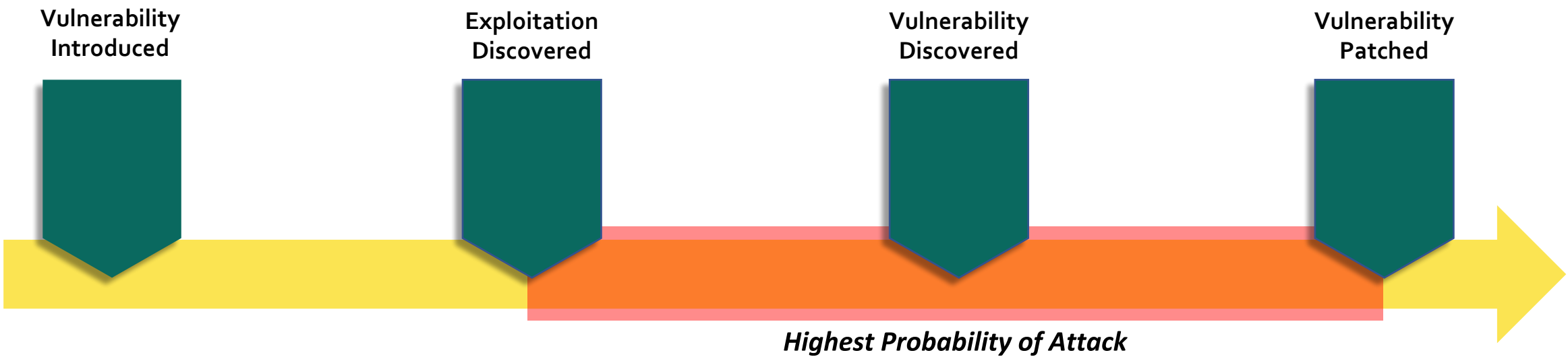
Michael Ducy

Sysdig
Falco Maintainer

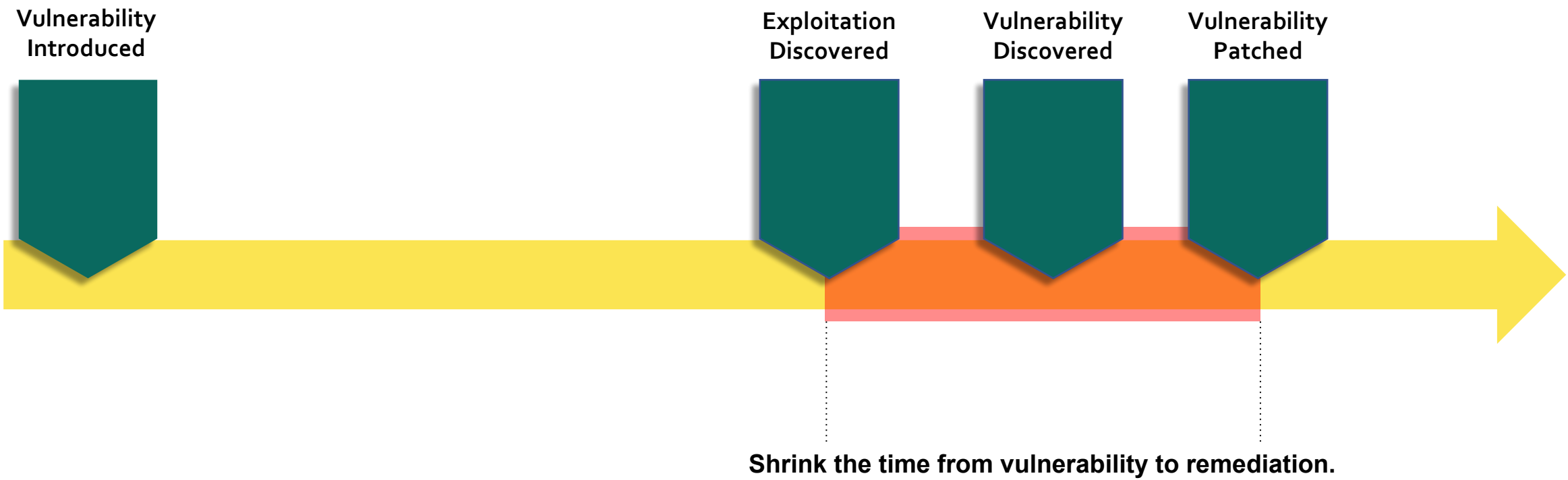
Steven Terrana

Booz Allen Hamilton
Jenkins Templating Engine Maintainer

Security Posture Lifecycle: Current State



Security Posture Lifecycle: Shift Left Security

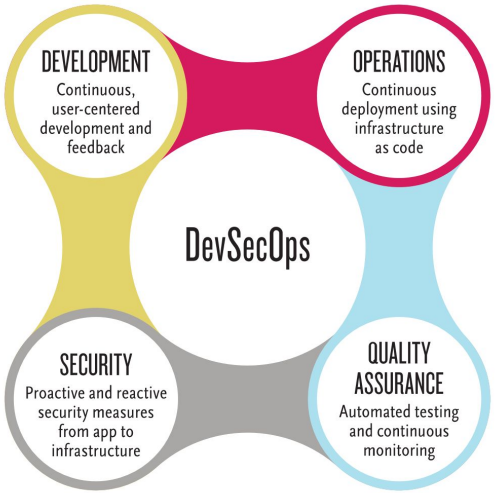


What is DevSecOps?

CONTINUOUS SECURITY & COMPLIANCE IS PERVASIVE IN OUR DEVOPS APPROACH. IT CROSS-CUTS EVERY PRACTICE AREA

Security and compliance are indicative of the same software delivery sins that spawned the DevOps movement. Work piles up because it is tedious, foreign, or difficult. Security pros are alienated and left to burn down the pile in isolation, as an afterthought. True concerns then become hugely disruptive, which breeds further discontent within the team.

AS WITH QUALITY ASSURANCE, SECURITY ASSURANCE AND COMPLIANCE CAN BE INTEGRATED INTO YOUR SOFTWARE DEVELOPMENT LIFECYCLE



- **Shift-left** many security and compliance activities as a **shared responsibility** of the whole team
- **Educate and automate** security vigilance to establish **early detection, confidence, and trust** required for Continuous Delivery
- Perform vulnerability and compliance **inspection** of dependencies, code, container images, and running applications

DEPENDENCIES

Prevent introduction of vulnerabilities from the outside. Scan libraries in dependency repos, source code repos, and on disk for known vulnerabilities.

STATIC CODE ANALYSIS

Analyze the code written by developers for inadvertent technical and logical flaws that make it vulnerable.

DYNAMIC APPLICATION SECURITY TESTING

Perform automated penetration testing to see how your application will withstand common attacks at runtime.

IMAGE SCANNING

Unpack and scan dependencies and configuration of the image to be used at runtime for vulnerabilities, out-of-date patching, and to ensure a trusted pedigree.

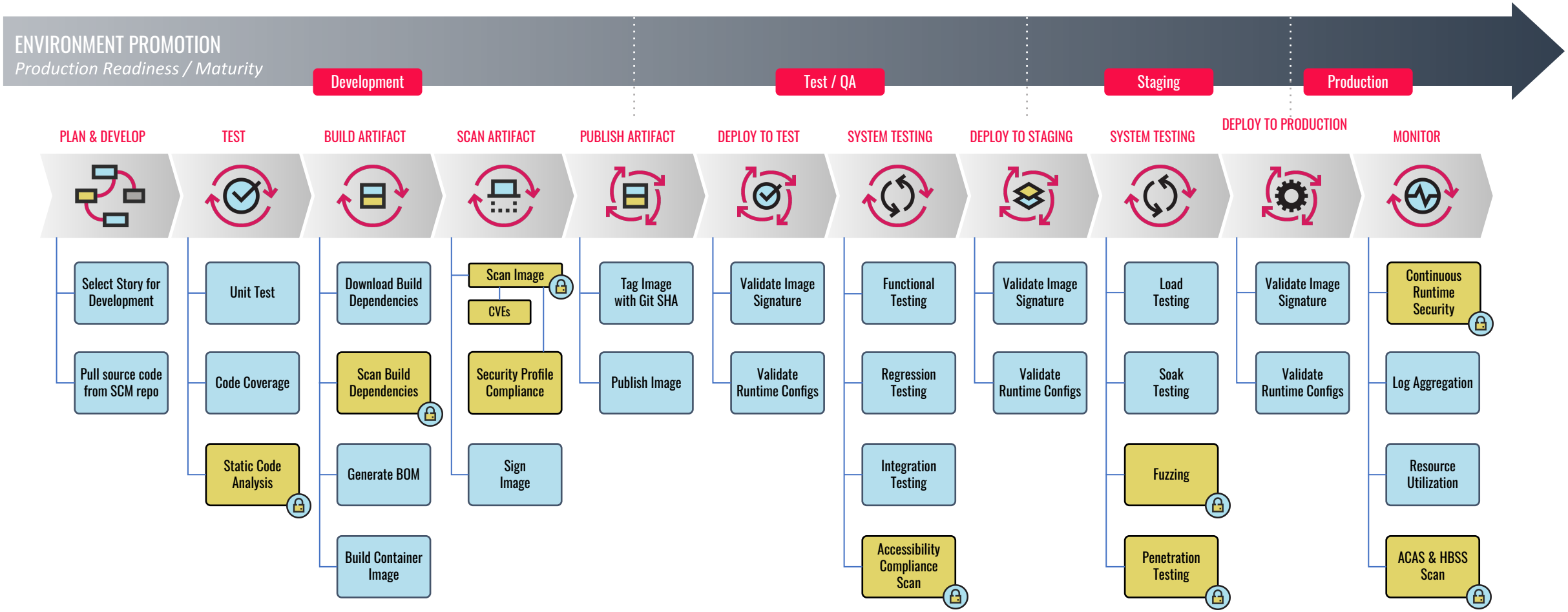
CONTINUOUS COMPLIANCE

Routinely scan the configuration of hosts or containers in their packaged image state or at runtime for compliance with security policy groups (NIST, CIS, FISMA, STIG, etc.), for required patches, or for configuration drift.

ACCESSIBILITY ASSURANCE

Crawl web pages for compliance with section 508 standards to give developers early warning and opportunity to improve the site while accelerating manual 508 testing.

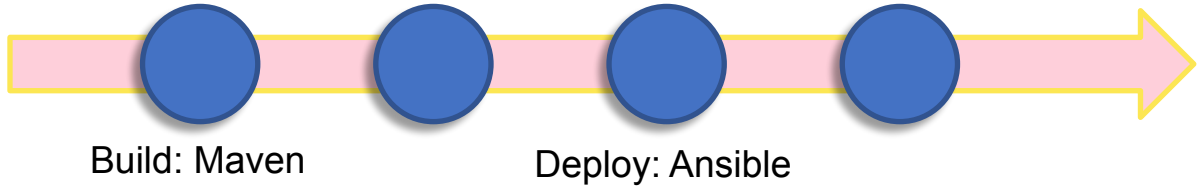
Trusted Software Supply Chain



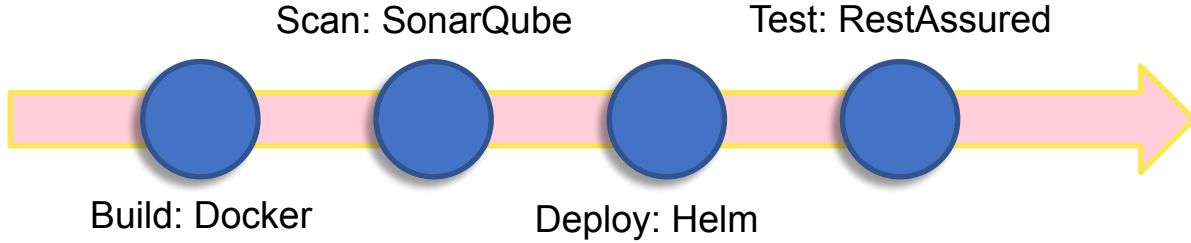
Jenkins Templating Engine

FROM

Source Code Repository



Source Code Repository



Jenkins Templating Engine

Project 1

TO

Project 2

Source Code Repository

Central DevOps Team defines organizational standards.

SecOps Engineer

DevOps Engineer

QA Engineer

Scan

Test

Build

Deploy

Pipeline Template

Source Code Repository

CONFIG

CONFIG

Scan: Fortify

Test: Selenium

Build: Maven

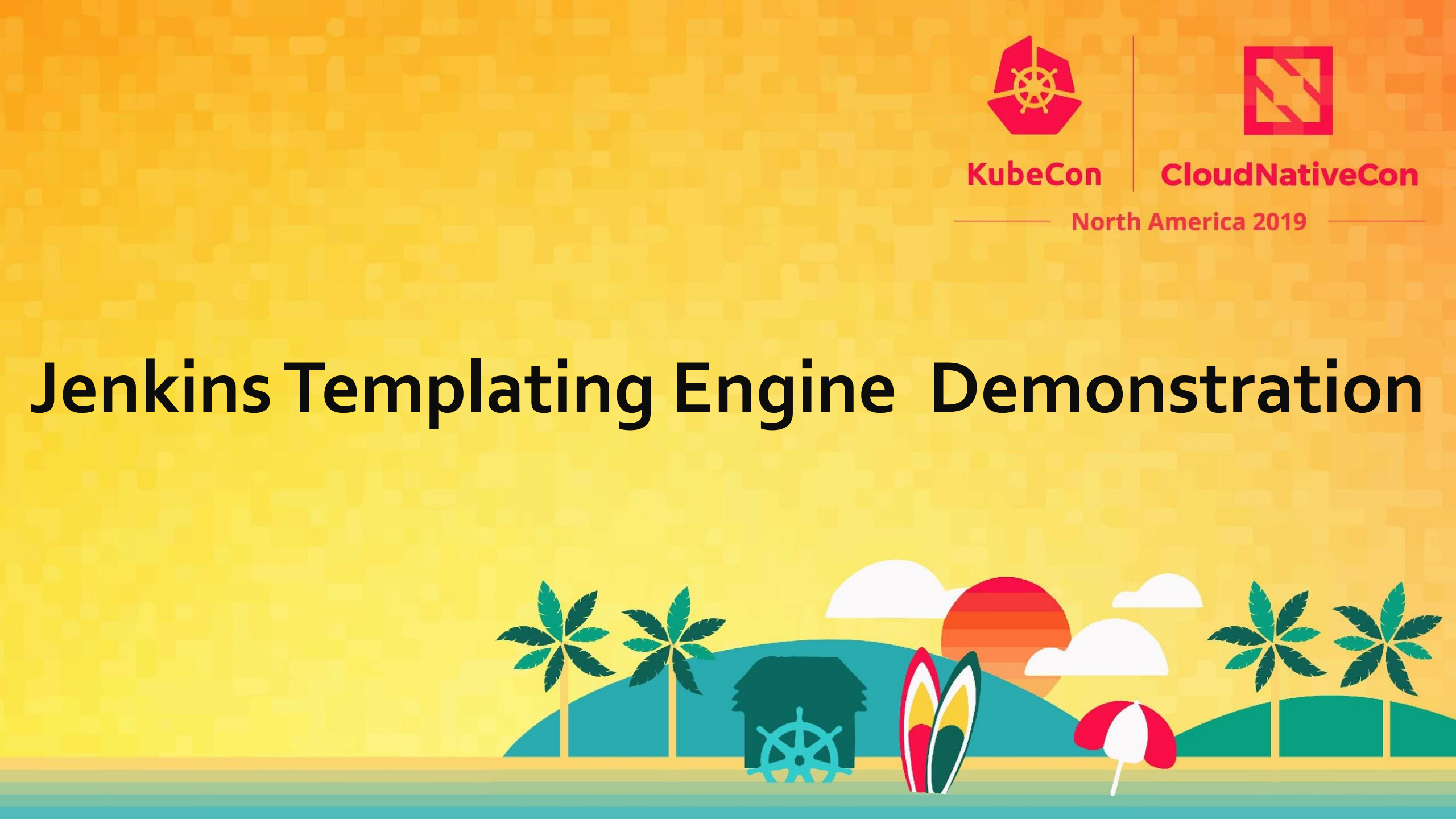
Deploy: Ansible

Scan: SonarQube

Test: RestAssured

Build: Docker

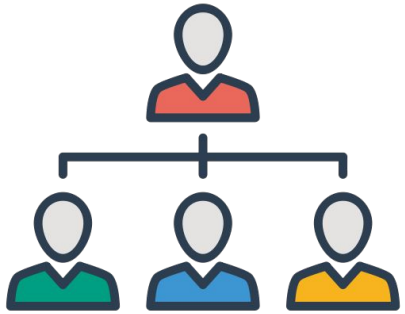
Deploy: Helm



Jenkins Templating Engine Demonstration

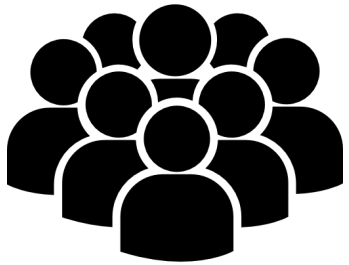
Benefits of JTE

Apply Organizational Governance



Standardize software delivery processes

Optimize Pipeline Code Reuse

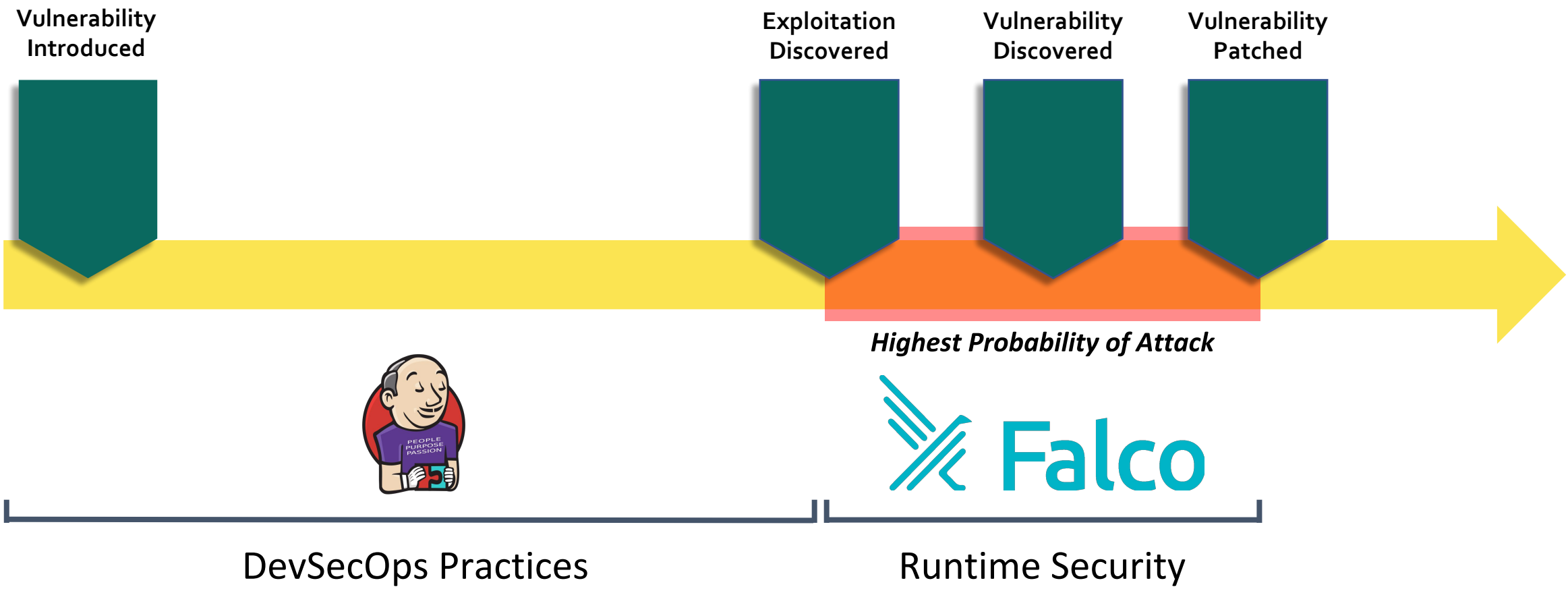


Crowd source quality through common open source tool integrations

Simplify Pipeline Maintainability

Manage centralized pipeline templates over individualized application-specific pipelines

Security Posture Lifecycle: Continuous Monitoring




KubeCon
CloudNativeCon
North America 2019
Falco

Falco.

A behavioral activity monitor

- Detects suspicious activity defined by a set of rules
- Uses sysdig's flexible and powerful filtering expressions

With full support for containers/orchestration

- Utilizes sysdig's container & orchestrator support

And flexible notification methods

- Alert to files, standard output, syslog, programs

Open Source

- Anyone can contribute rules or improvements

@mfdii

Falco: a CNCF sandbox project.

Runtime security for cloud native platforms

- Detect abnormal behavior in applications, containers, and hosts.
- Audit orchestrator activity.

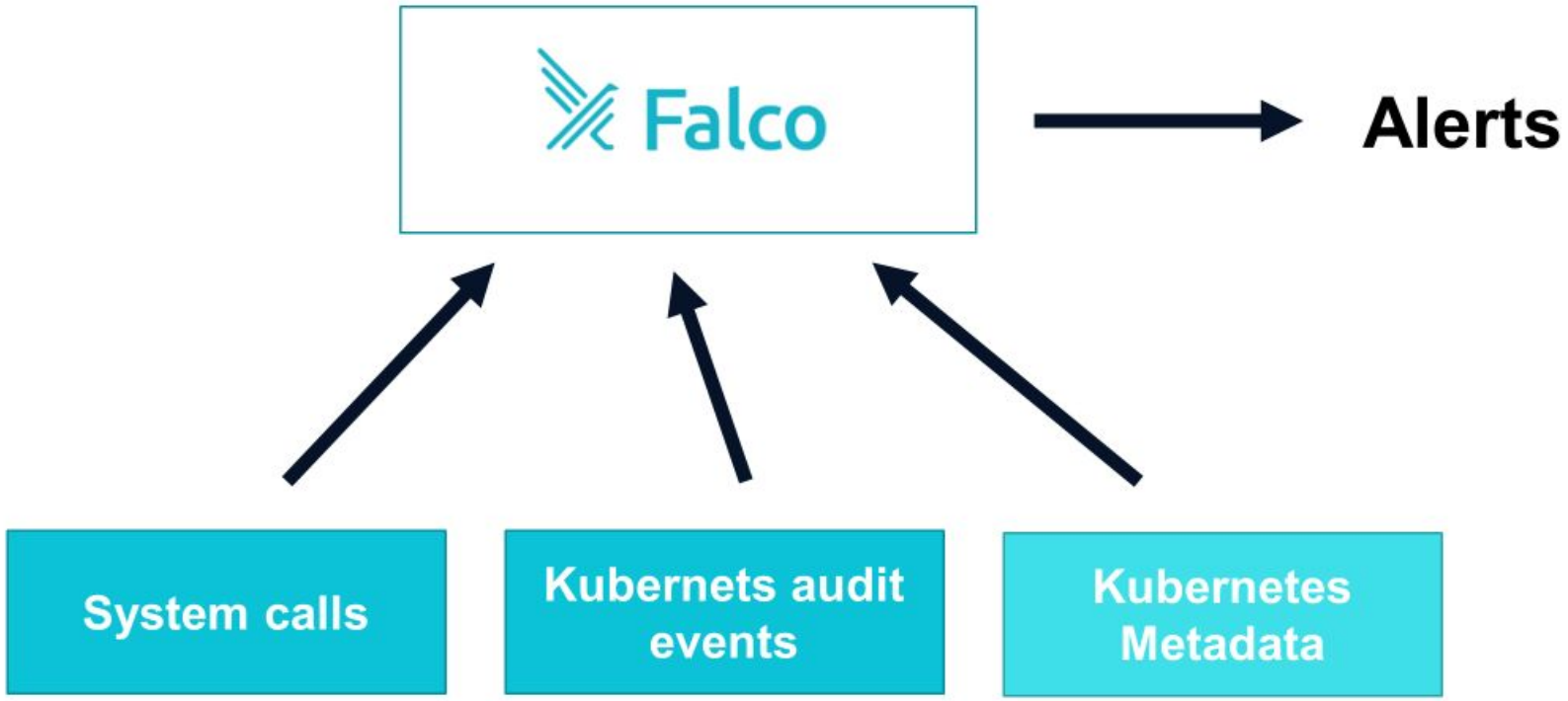
Cloud Native Computing Foundation (CNCF)

- Sandbox level project
- Incubation Proposal: https://github.com/cncf/toc/pull/307

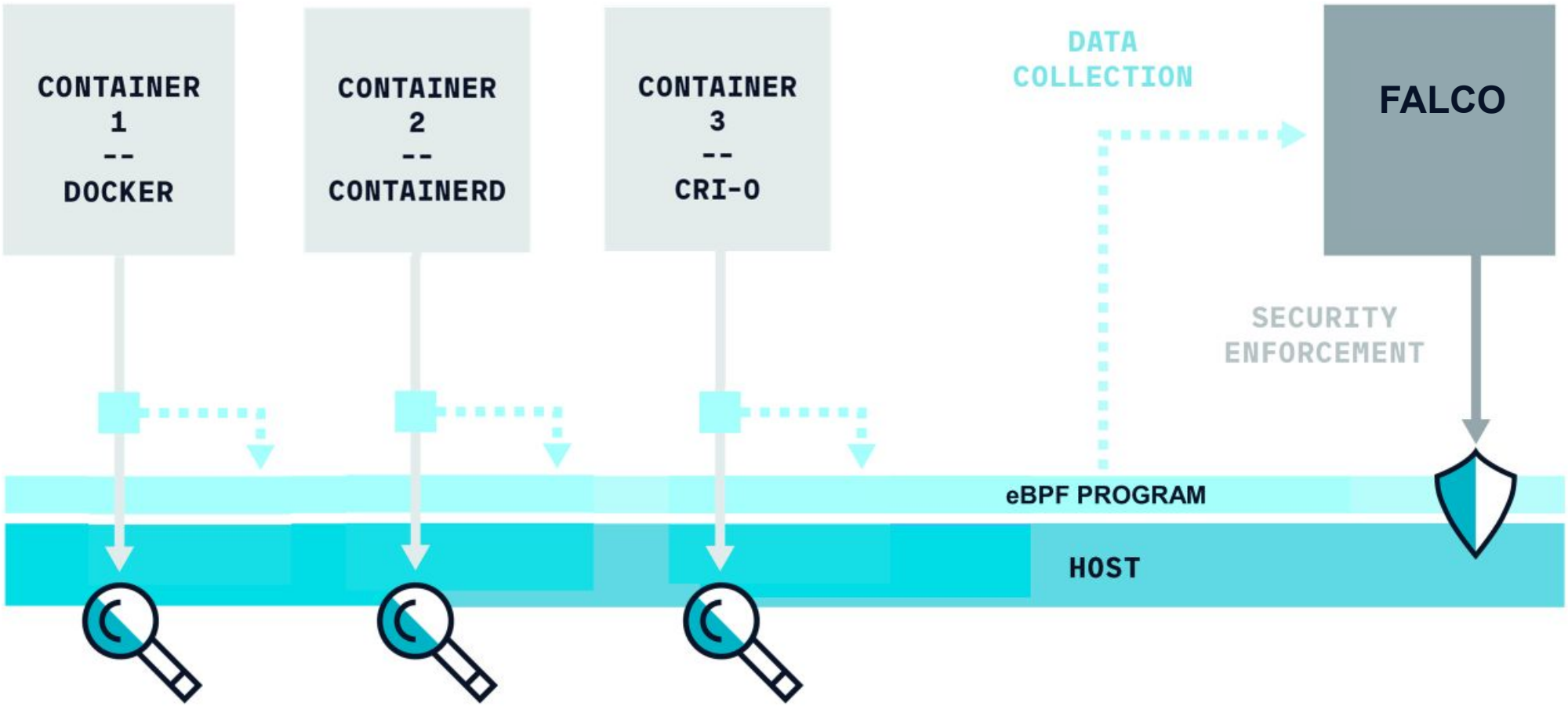
Anomaly detection.

- Containers are **isolated** processes.
- Processes are **scoped** as to what's expected.
- Container images are **immutable, runtime environments** often **aren't.**
- How do you **detect abnormal behavior?**

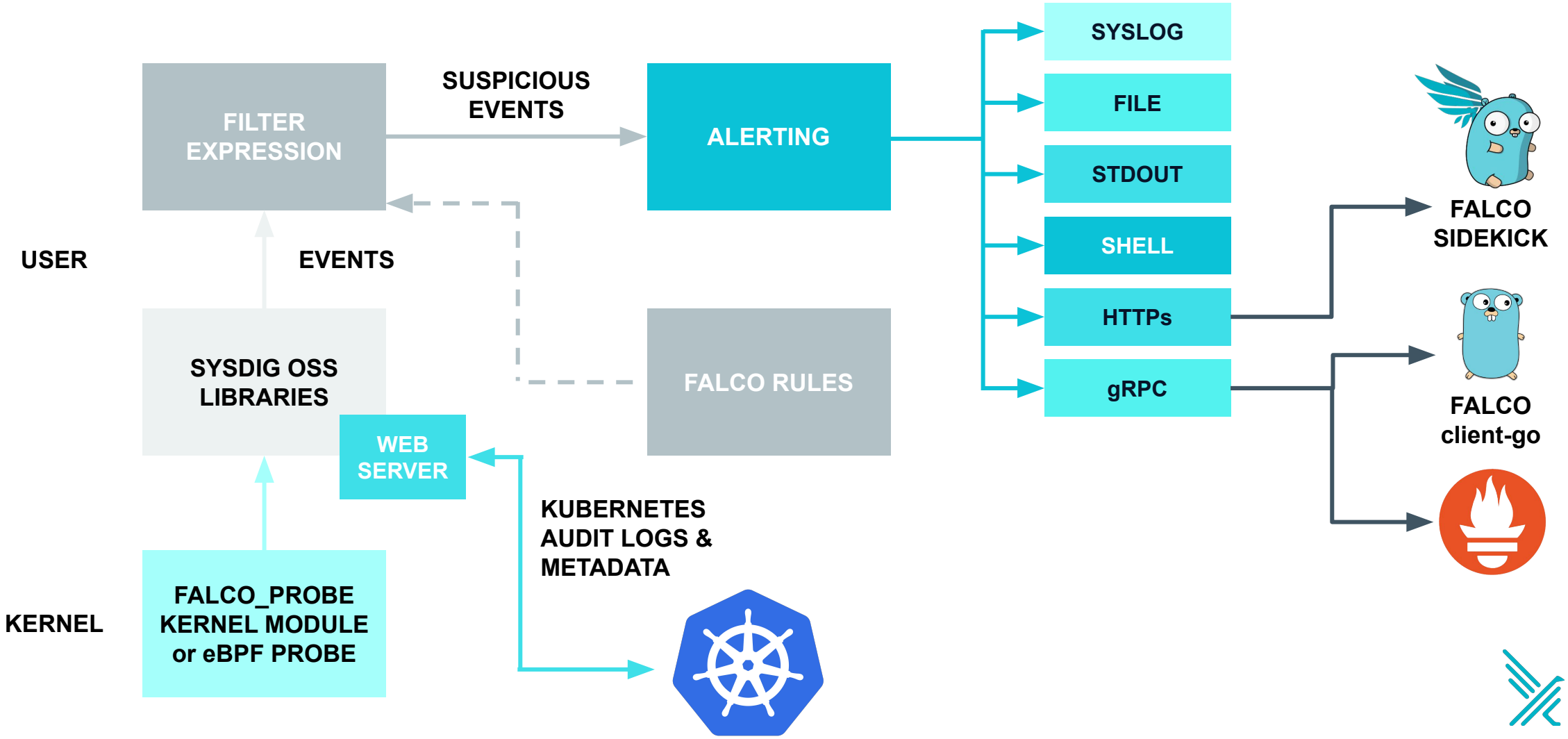
Architecture.


Falco
Alerts
System calls
Kubernets audit events
Kubernetes Metadata

Falco instrumentation



Falco Architecture



@falco_org

Falco Rules

Falco rules.

yaml file containing Macros, Lists, and Rules

```
- list: bin_dirs
  items: [/bin, /sbin, /usr/bin, /usr/sbin]
- macro: bin_dir
  condition: fd.directory in (bin_dirs)
- rule: write_binary_dir
  desc: an attempt to write to any file below a set of binary directories
  condition: bin_dir and evt.dir = < and open_write and not
package_mgmt_procs
  output: "File below a known binary directory opened for writing
    (user=%user.name command=%proc.cmdline file=%fd.name)"
  priority: WARNING
```


Conditions and Sysdig Filter Expressions.

Based on “Field Classes”. Supported classes include:

fd - File Descriptors

process - Processes

evt - System events

user - Users

group - Groups

syslog - Syslog messages

container - Container metadata

fdlist - FD poll events

k8s - Kubernetes metadata

ka - Kubernetes Audit Logs

mesos - Mesos metadata

A custom Falco rule.

```
- rule: Node Container Runs Node
  desc: Detect a process that's not node started in a Node container.
  condition: evt.type=execve and container.image startswith node and proc.name!=node
  output: Node container started other process (user=%user.name
                command=%proc.cmdline %container.info)
  priority: INFO
  tags: [container, apps]
```

Something is executing a program

In a container based on the Node image

And the process name isn't node

Kubernetes Audit Log Events

- New in K8s v1.11
- Provides chronological set of records documenting changes to cluster
- Each record is a JSON object
- Audit policy controls which events are included in event log
- Log backend controls where events are sent
  - Log file
  - Webhook
  - AuditSink (alpha as of 1.13)

Kubernetes Audit Events.

```
{
    "kind": "Event",
    "timestamp": "2018-10-26T13:00:25Z",
    "stage": "ResponseComplete",
    "verb": "delete",
    "requestURI": "/api/v1/namespaces/foo",
    "user": { "username": "minikube-user" },
    "responseStatus": { "code": 200 },
    "objectRef": { "resource": “namespaces", "namespace": “foo” },
    "level": "Request",
    "auditID": "693f4726-2430-450a-83e1-123c050fde98",
    "annotations": { "authorization.k8s.io/decision": "allow" }
}
```


Kubernetes Audit Event Fields.

- **`jevt.value[<json_pointer>]`**
  - Access any field from json object
- **`jevt.time`**
  - Access event timestamp
- **`ka.verb, ka.uri, ka.user.name, ka.target.resource, …`**
  - Access specific values from object
  - Implemented as macros:
    - ka.verb -> jevt.value[/verb]
    - ka.target.resource -> jevt.value[/objectRef/resource]
  - Full list: `falco –list=k8s_audit`

K8s audit log rule example.

```
- macro: contains_private_credentials
  condition: >
    (ka.req.configmap.obj contains "aws_access_key_id" or
     ka.req.configmap.obj contains "aws_s3_access_key_id" or
     ka.req.configmap.obj contains "password")

- macro: configmap
  condition: ka.target.resource=configmaps

- macro: modify
  condition: (ka.verb in (create,update,patch))

- rule: Create/Modify Configmap With Private Credentials
  desc: Detect creating/modifying a configmap containing a private credential
    (aws key, password, etc.)
  condition: configmap and modify and contains_private_credentials
  output: K8s configmap with private credential (user=%ka.user.name
          verb=%ka.verb name=%ka.req.configmap.name
          configmap=%ka.req.configmap.name config=%ka.req.configmap.obj)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```


Installing and Integrations

Installing Falco.

- **Debian Package**
  - apt-get -y install falco
- **Redhat Package**
  - yum -y install falco
- **Installation Script**
  - curl -s s3.amazonaws.com/download.draios.com/stable/install-falco | sudo bash
- **Docker container**
  - docker pull sysdig/falco
- **Full instructions**
  - github.com/draios/falco/wiki/How-to-Install-Falco-for-Linux

Installing Falco on kubernetes.

- **Use Helm**
  - $ helm install --name sysdig-falco-1 stable/falco
  - https://sysdig.com/blog/falco-helm-chart/

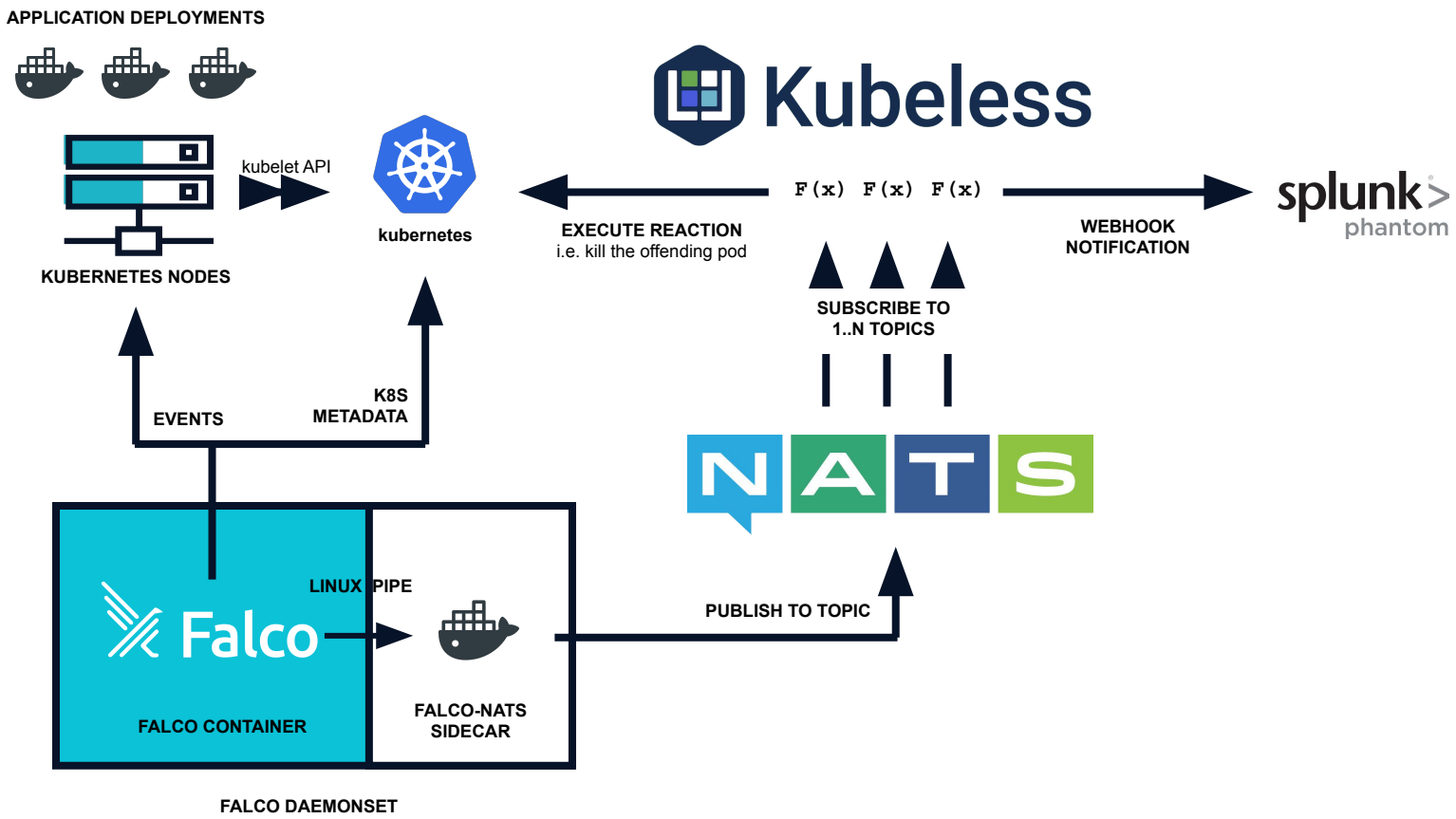
- **Install Falco as Kubernetes Daemonset**
  - https://github.com/draios/falco/tree/dev/examples/k8s-using-daemonset
  - Configuration stored in Kubernetes ConfigMaps
  - Conditions in a Falco Rule can leverage Kubernetes metadata to trigger events
  - Falco events can include Kubernetes metadata to give notification context:
    - name, id, labels for Pods, ReplicationController, Service, Namespace, ReplicaSet, and Deployment

How can you use Falco?

Response engine & security playbooks.

- **Detect abnormal events** with Falco
- **Publish alerts** to Pub/Sub service (NATS.io)
- Subscribers can **subscribe to various FALCO topics** to receive alerts:
  - FALCO.* - All alerts
  - FALCO.Notice - Alerts of priority “Notice” only
  - FALCO.Critical - Alerts of priority “Critical” only
- Subscribers can **take action** on alerts:
  - Kill offending Pod
  - Taint Nodes to prevent scheduling
  - Isolate Pod with Networking Policy
  - Send notification via Slack

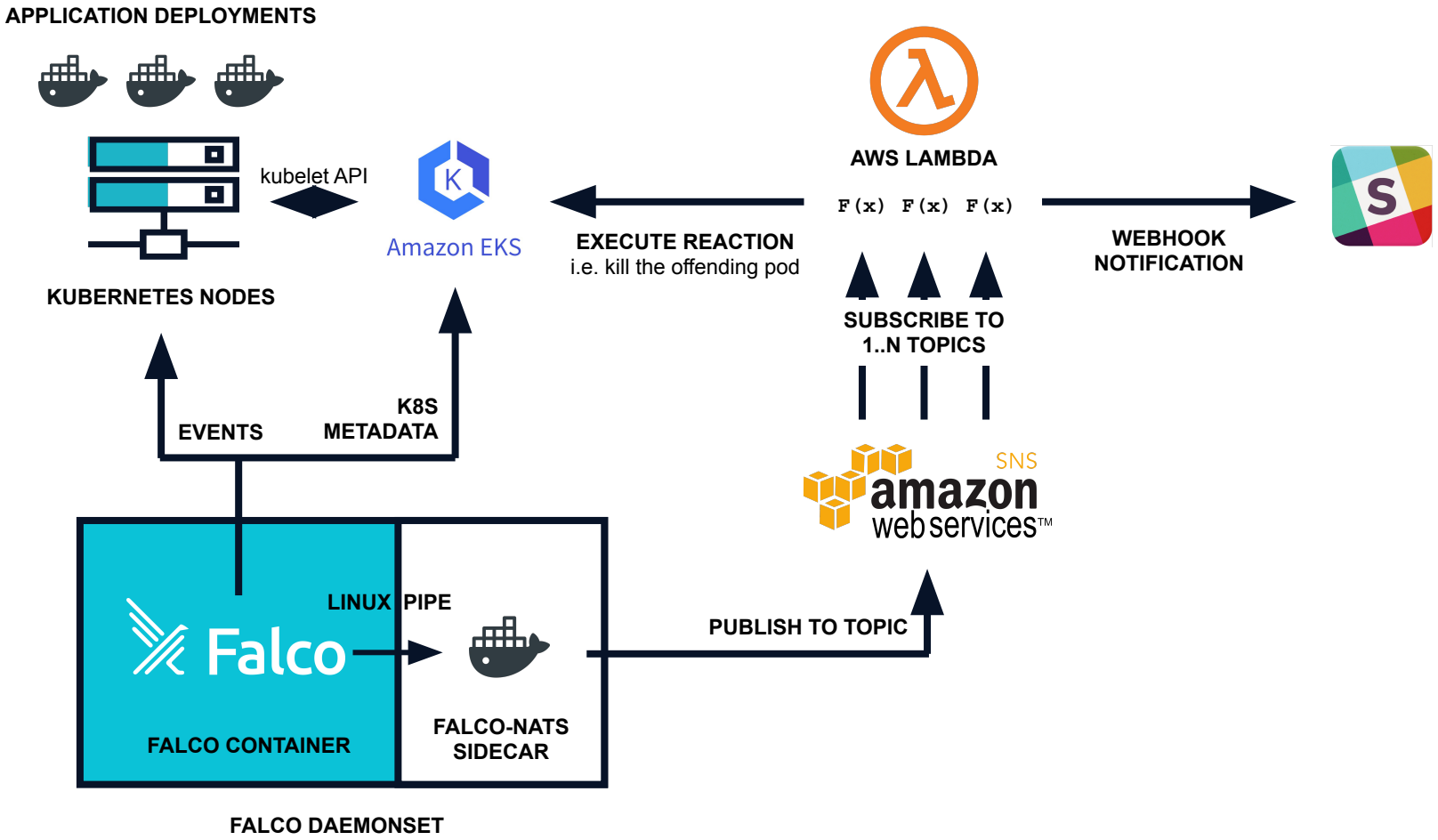
Response engine & security playbooks.



https://sysdig.com/blog/container-security-orchestration-falco-splunk-phantom/

@mfdii

Response engine & security playbooks.



https://aws.amazon.com/blogs/opensource/securing-amazon-eks-lambda-falco/

@mfdii

SIEM with EFK.

Detects abnormal event, Publishes alert to stdout

Fluentd ships alerts to Elasticsearch

Kibana dashboards can be used to aggregate, filter, and report on alerts.

Join the community.

Website

- https://falco.org

Public Slack

- http://slack.sysdig.com/
- https://sysdig.slack.com/messages/falco

Blog

- https://falco.org/blog/

Github

- https://github.com/falcosecurity/falco/

Documentation

- https://falco.org/docs

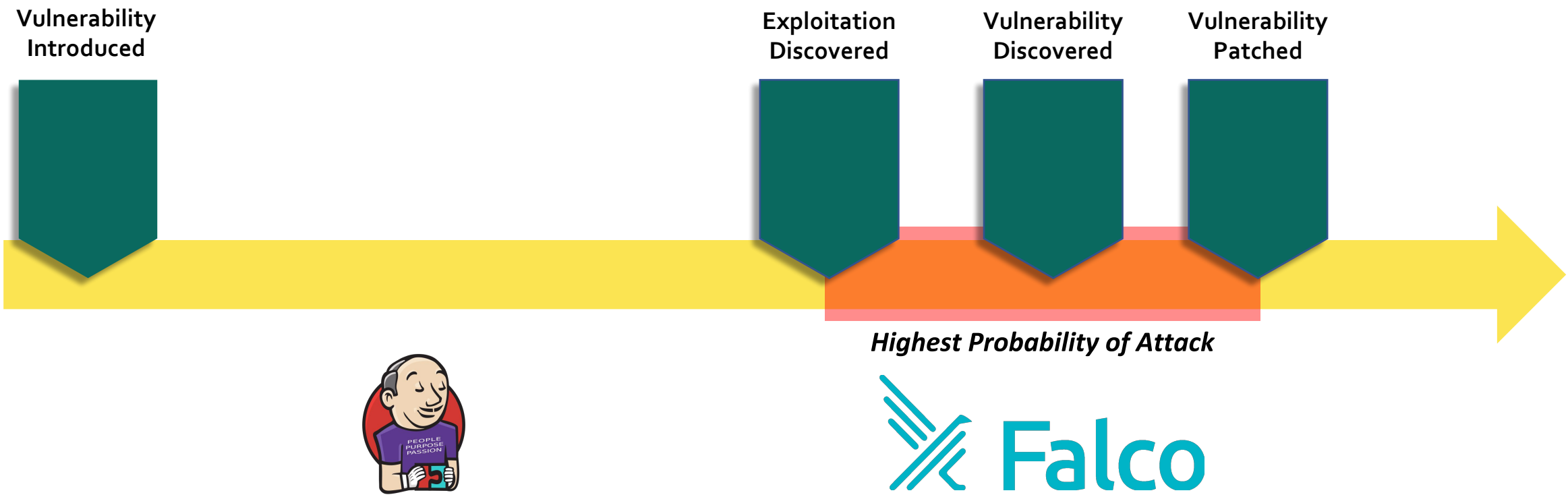
Docker Hub

- https://hub.docker.com/r/falcosecurity/falco/

@mfdii

Falco Demonstration

Security Posture Lifecycle: Continuous Monitoring



Learn More!

Falco