



KubeCon



CloudNativeCon

North America 2017

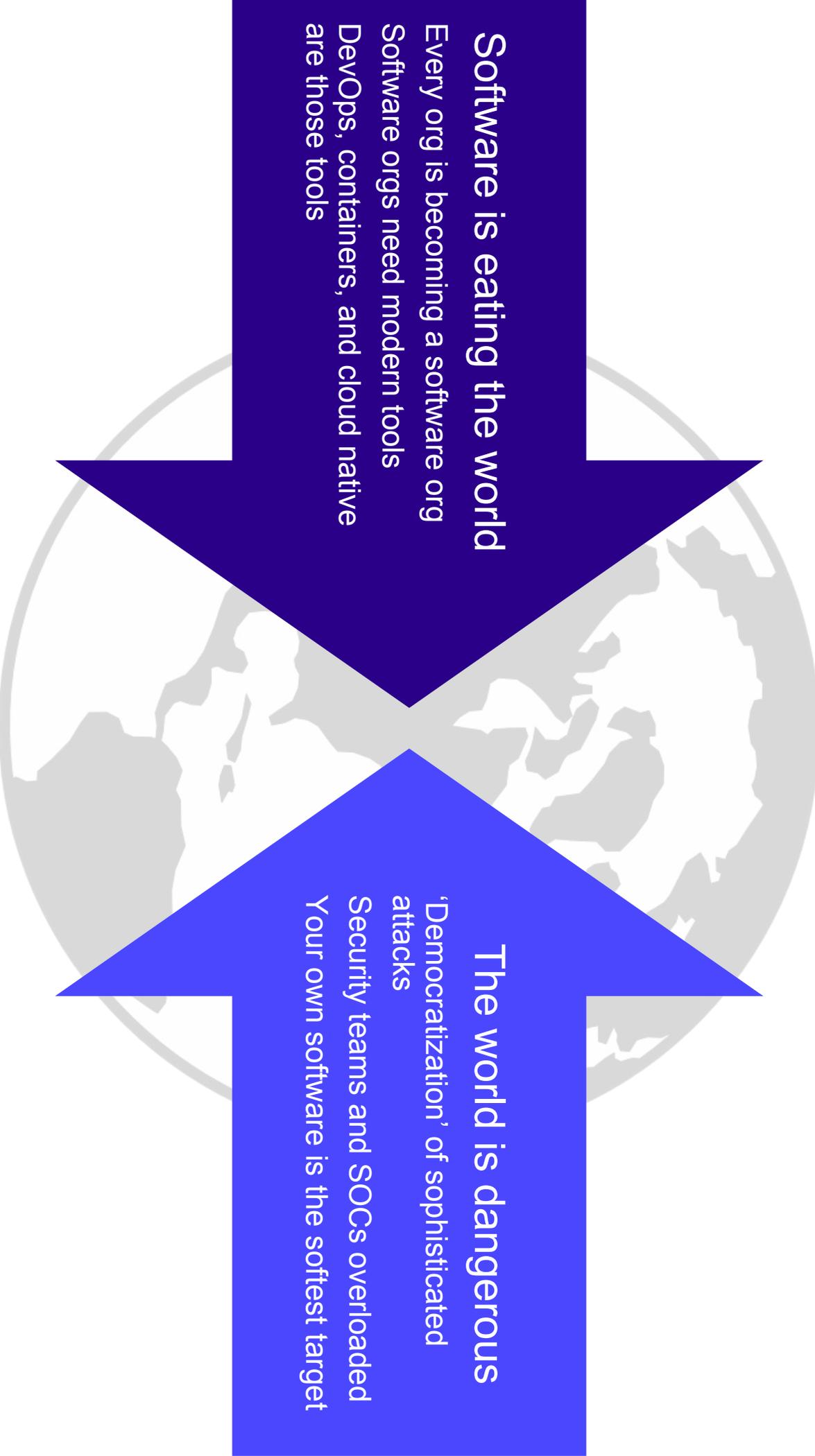


**Twistlock**<sup>™</sup>

John Morello

CTO

[john@twistlock.com](mailto:john@twistlock.com)  
[@morellonet](mailto:@morellonet)



## Software is eating the world

Every org is becoming a software org  
Software orgs need modern tools  
DevOps, containers, and cloud native  
are those tools

## The world is dangerous

'Democratization' of sophisticated  
attacks  
Security teams and SOCs overloaded  
Your own software is the softest target

# Containers improve security

# Old World Security

Developers manually describe their apps to security teams

Security teams manually create policies in multiple tools

As the app evolves, the rules rot



WordPress runs on Apache 2.2 and needs port 80

MySQL listens on port 3306 and gets requests from WordPress

```
firewall: allow tcp/80 on 10.0.20.12
firewall allow tcp/3306 on 10.0.20.16
ids: allow httpd 2.2.31
ids: allow mysql 5.7.9
```



I'm upgrading Apache and need to run MySQL on a different server

```
allow any/any
```



# Security Cake

You can have DevOps and  
containers or you can keep  
your old world security  
approach...

**But you can't have  
both**



# An Opportunity for Better Defense

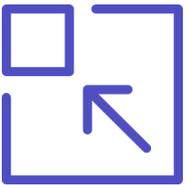


The nature of containers allows for a new approach to security

Apply machine learning to understand actual runtime behavior

Build models of what containers ***should*** do to detect and prevent what they ***shouldn't***

# Container Characteristics



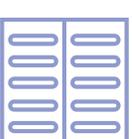
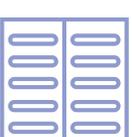
## Minimal

Typically single  
process entities



## Declarative

Built from  
images that are  
machine  
readable



## Predictable

Do exactly the  
same thing from  
**run** to **kill**

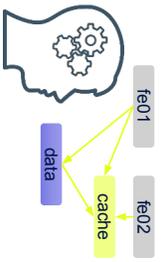
# Autonomous Defense

```

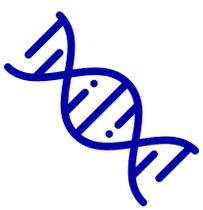
1:~/code$ git checkout master
2:~/code$ git pull
3:~/code$ git push
4:~/code$ git push
5:~/code$ git push
6:~/code$ git push
7:~/code$ git push
8:~/code$ git push
9:~/code$ git push
10:~/code$ git push
11:~/code$ git push
12:~/code$ git push
13:~/code$ git push
14:~/code$ git push
15:~/code$ git push
16:~/code$ git push
17:~/code$ git push
18:~/code$ git push
19:~/code$ git push
20:~/code$ git push
21:~/code$ git push
22:~/code$ git push
23:~/code$ git push
24:~/code$ git push
25:~/code$ git push
26:~/code$ git push
27:~/code$ git push
28:~/code$ git push
29:~/code$ git push
30:~/code$ git push
31:~/code$ git push
32:~/code$ git push
33:~/code$ git push
34:~/code$ git push
35:~/code$ git push
36:~/code$ git push
37:~/code$ git push
38:~/code$ git push
39:~/code$ git push
40:~/code$ git push
41:~/code$ git push
42:~/code$ git push
43:~/code$ git push
44:~/code$ git push
45:~/code$ git push
46:~/code$ git push
47:~/code$ git push
48:~/code$ git push
49:~/code$ git push
50:~/code$ git push
51:~/code$ git push
52:~/code$ git push
53:~/code$ git push
54:~/code$ git push
55:~/code$ git push
56:~/code$ git push
57:~/code$ git push
58:~/code$ git push
59:~/code$ git push
60:~/code$ git push
61:~/code$ git push
62:~/code$ git push
63:~/code$ git push
64:~/code$ git push
65:~/code$ git push
66:~/code$ git push
67:~/code$ git push
68:~/code$ git push
69:~/code$ git push
70:~/code$ git push
71:~/code$ git push
72:~/code$ git push
73:~/code$ git push
74:~/code$ git push
75:~/code$ git push
76:~/code$ git push
77:~/code$ git push
78:~/code$ git push
79:~/code$ git push
80:~/code$ git push
81:~/code$ git push
82:~/code$ git push
83:~/code$ git push
84:~/code$ git push
85:~/code$ git push
86:~/code$ git push
87:~/code$ git push
88:~/code$ git push
89:~/code$ git push
90:~/code$ git push
91:~/code$ git push
92:~/code$ git push
93:~/code$ git push
94:~/code$ git push
95:~/code$ git push
96:~/code$ git push
97:~/code$ git push
98:~/code$ git push
99:~/code$ git push
100:~/code$ git push

```

+



=



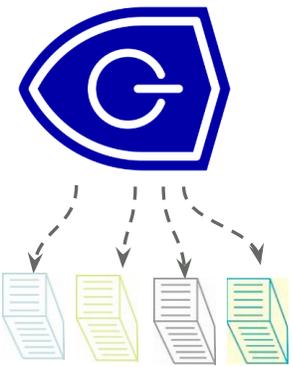
+

```

ip, category, score, first_seen,
last_seen, ports
74, 88, 8.7, 31, 65, 2016-04-16, 2016-
04-16,
23, 16, 9.49, 35, 125, 2016-04-11, 201
6-04-20, 80
82, 16, 9.65, 35, 127, 2016-04-09, 201
6-04-21, 80

```

=



Static  
analysis

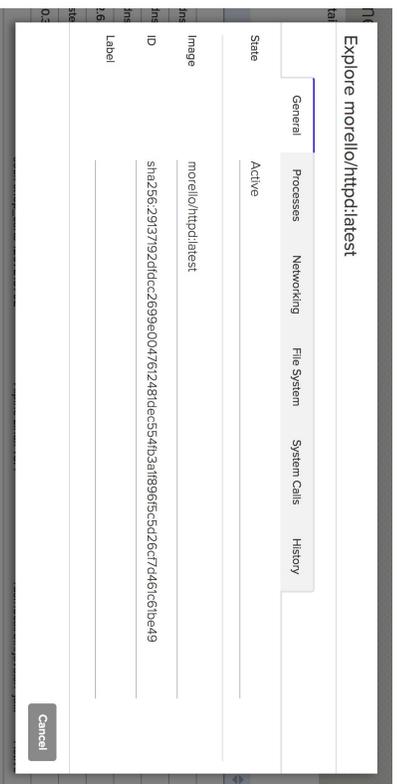
Machine  
learning

Predictive  
model

Threat  
intelligence

Automated  
defense

# What's a Model?



Automatically learned across 4 dimensions

Correlated to unique image digest

Models are explicit allow lists



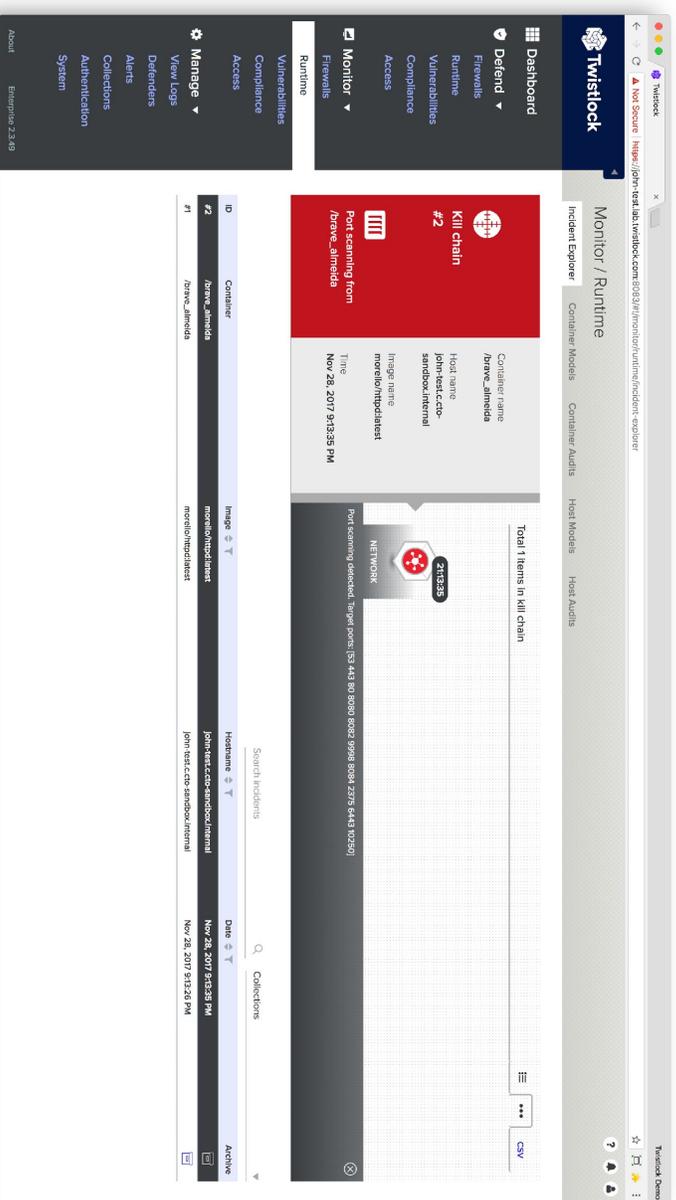
© 2017

```
root@john-test:/home/john# curl -k -u user:pass
https://localhost:8083/api/v1/profiles/container" | jq . | grep httpd -A 20 -B 5
[...
{
  "state": "active",
  "remainingLearningDurationSec": 0,
  "_id": "sha256:29137192dfdcc2699e0047612481dec554fb3a1f696f5c5d26cf7d461c61be49",
  "image": "morello/httpd:latest",
  "version": "2.3.49",
  "os": "Debian GNU/Linux 8 (jessie)",
  "archived": false,
  "entrypoint": "httpd-foreground",
  "infra": false,
  "processes": {
    "static": [
      {
        "path": "/usr/local/apache2/bin/httpd",
        "path": "",
        "md5": "f9ff43be045e42222f08bb6dede293af3"
      },
      {
        "path": "/bin/bash",
        "path": "",
        "md5": "33135f5a1fd45f5dff91sec1193c0dc7"
      }
    ],
    "network": {
      "static": {
        "listeningPorts": [
          {
            "app": "/usr/local/apache2/bin/httpd",
            "ports": [
              80
            ],
            "allPorts": false
          }
        ]
      },
      "behavioral": {
        "outboundPorts": [],
        "listeningPorts": [],
        "dnsQueries": []
      }
    }
  }
}
```

# Automatic Prevention

Models enable automatically detecting anomalies at scale

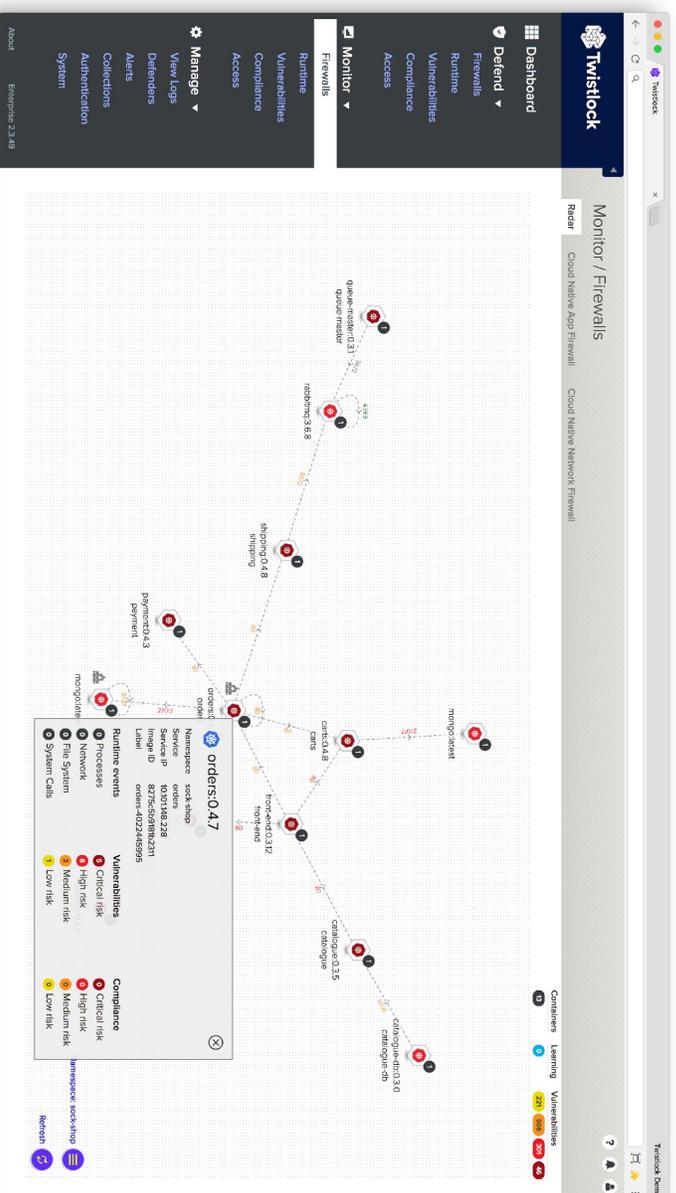
Correlate data from multiple sensors to summarize multilayered attack patterns



# Supermodels

Correlate models from multiple microservices to create supermodels

Visualize and compartmentalize intra-namespace traffic flows



# New World Security

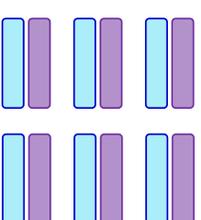
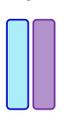
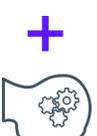
Modeling integrated into development and deployment

Custom tailored policy for each app

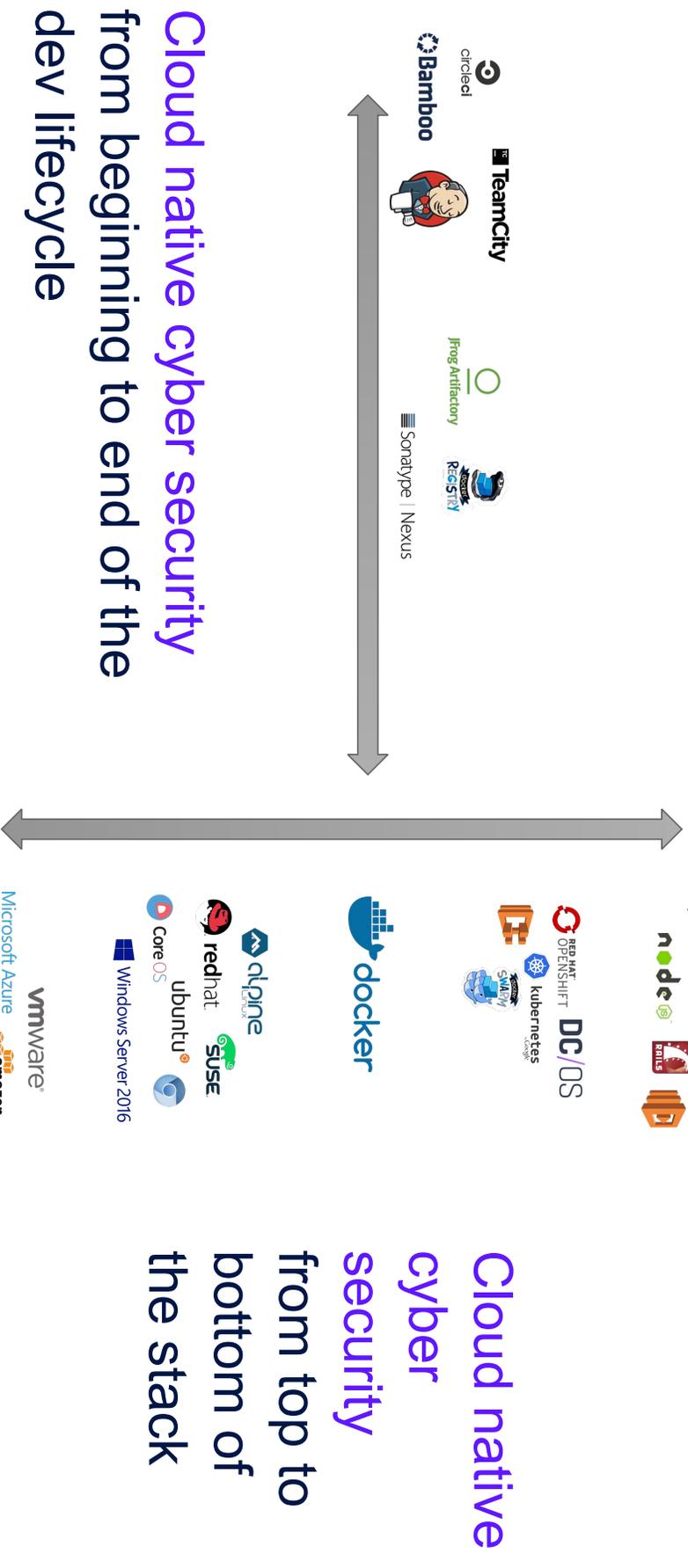
Security scales with the app, everywhere it runs



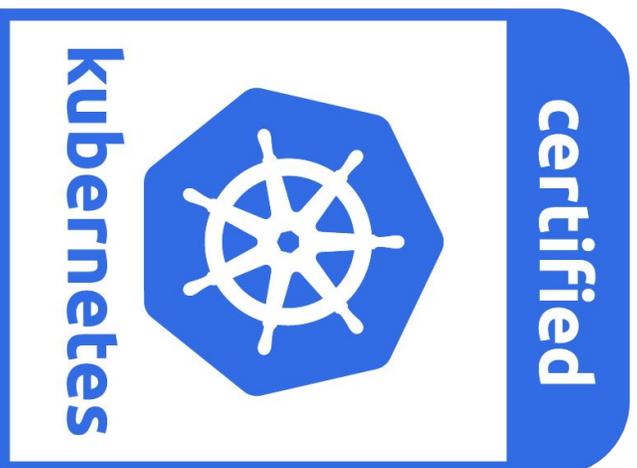
```
var express =  
require('express'); var  
app = express();  
  
app.listen(3000, function () {  
  console.log('Example app  
  listening on port 3000!'); });
```



# What is Twistlock?



# Kubernetes Certified



<https://github.com/cncf/k8s-conformance/pull/94>

Deploy Console as a  
Replication Controller

Deploy Defender to every node  
as a Daemon Set

# Who Are We?

## Technology pioneer and innovator

- Started in early 2015 as the first ever purpose-built solution for containers and cloud native security
- 12 major releases shipped, including >250 customer requested features to date
- 13 patents pending
- 4 container related 0-days discovered by our research team

## Market leader

- >100 customers around the world
- Enterprise grade global support with 24/7/365 SLA
- >\$30M raised to date

## Ecosystem leader

- We built the authorization framework in Docker and OpenShift and secrets management in Docker Swarm
- Lead author of NIST SP 800-190, the Container Security Guide
- The launch partners for Amazon, Google, and Microsoft's container services

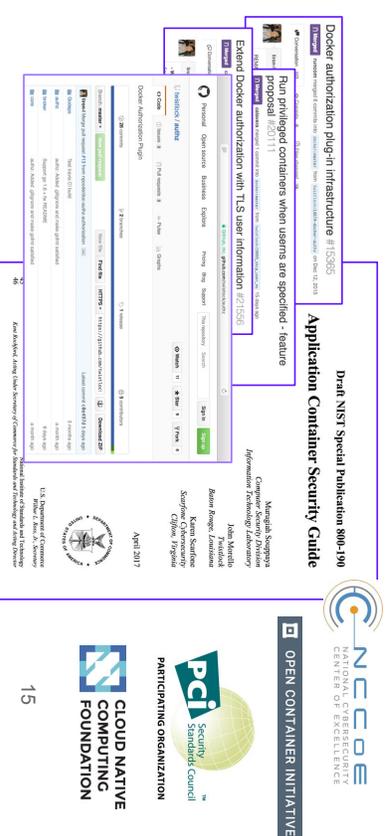
Awards & press



Partnerships



Open source and standards work



© 2017

twistlock.com

@TwistlockTeam

[sales@twistlock.com](mailto:sales@twistlock.com)

Booth G28

