



# Aqua Container Security Platform

Security and Compliance in the era of micro-services

# About Aqua

## TEAM

80 passionate, experienced innovators coming from:



## STRATEGIC PARTNERSHIPS



## BACKED BY

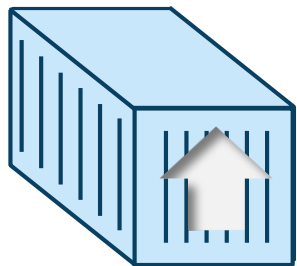
Microsoft Ventures | Lightspeed | Shlomo Kramer | TLV Partners

📍 Boston

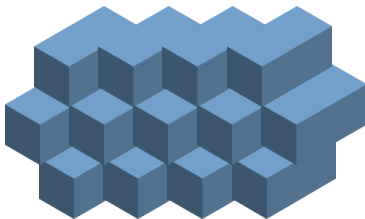
📍 San Francisco

📍 Tel Aviv

# Why containers



Up in seconds



Massive scale



Run Anywhere

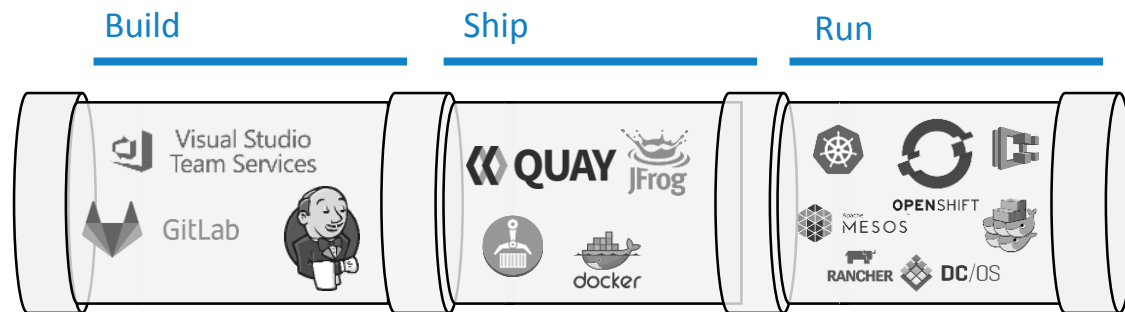


No finger pointing



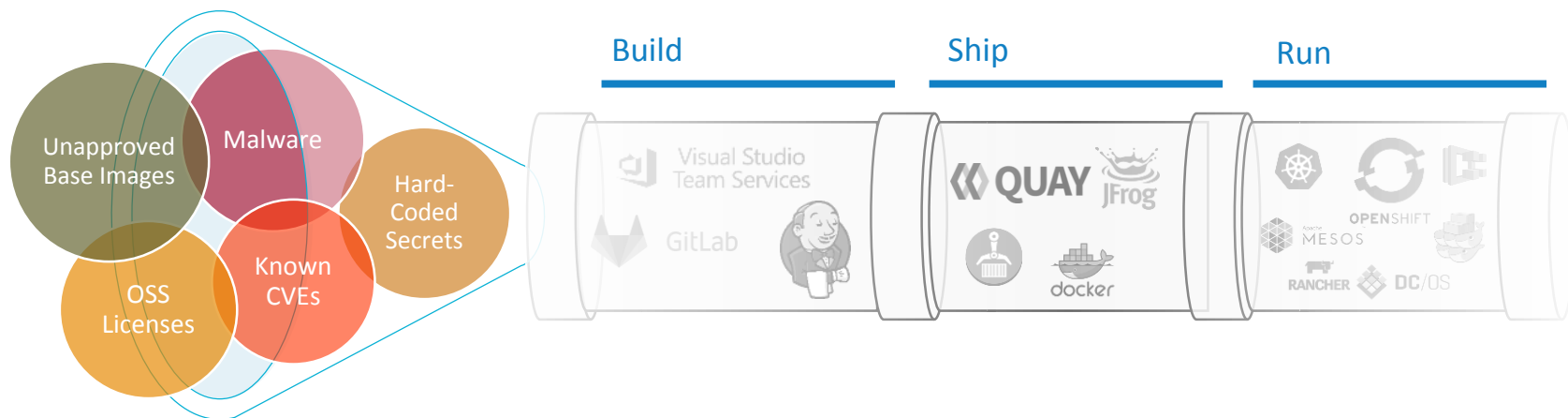
# How is that possible?

- Images with everything preconfigured
- Automated build, distribution and orchestration
- Small footprint per workload without VM overhead



# The view from Security

- Development is making infrastructure decisions
- Code moves too fast for risk analysis
- Thousands of containers with limited visibility or control



# How Aqua can help



Automate  
DevSecOps



Focus on  
Prevention



Portable  
Controls

# It starts with building an image



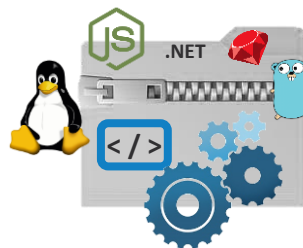
Base Image

```
/
├── bin
├── etc
├── proc
├── root
├── run
├── sys
├── usr
└── var
```



Builder Image

```
/
├── bin
├── etc
├── lib
├── proc
├── root
├── run
├── sys
├── usr
├── var
└── opt
```

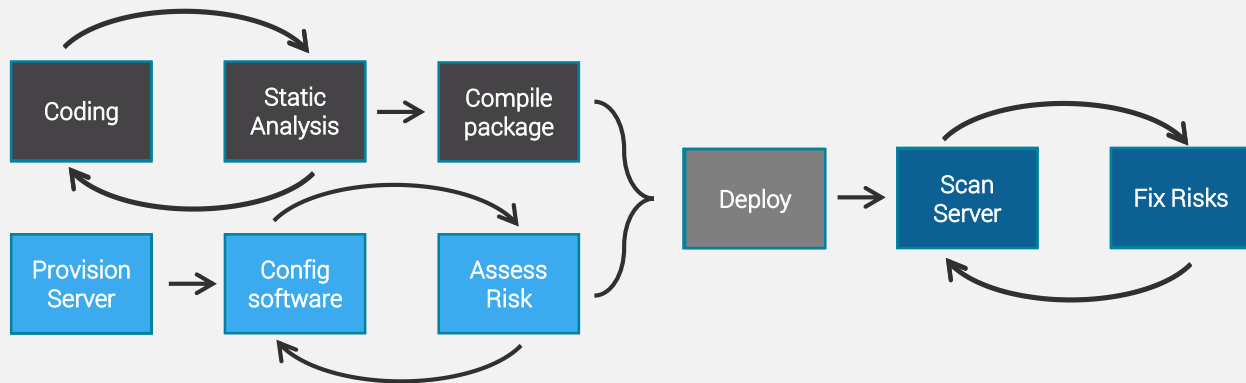


Application

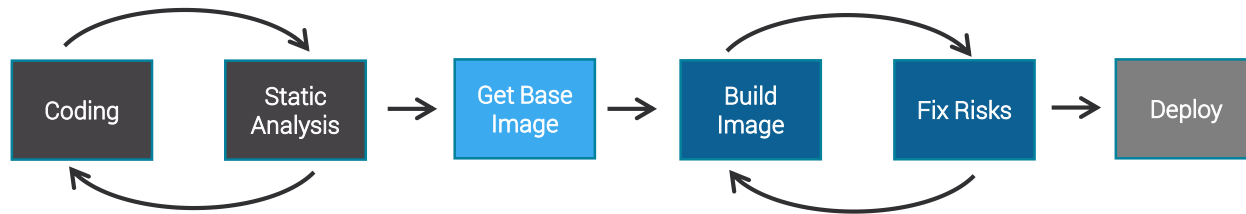
```
/
├── bin
├── etc
├── lib
├── proc
├── root
├── run
├── sys
├── usr
├── var
├── opt
└── ┬── app
```

# Where security fits in

Using Servers



Using Containers





# Detailed image risk information

Repositories & Images > centos:7

Vulnerabilities Packages Metadata History

Image Overview

6 High 12 Medium 1 Low 3.7 Average Score

CVE	SEVERITY	PACKAGE
> CVE-2016-5636	High	python
> CVE-2016-5636	High	python-libs
> CVE-2016-2834	High	nss-util
∨ CVE-2016-2834	High	nss

**Description:** Mozilla Network Security Services (NSS) before 3.23, as used in Mozilla Firefox before 47.0, allows remote attackers to cause unspecified other impact via unknown vectors.

**CVSS v2 Score:** 9.3

**Vector:** AV:N/AC:M/Au:N/C:C/I:C/A:C

**Fix Version:** nss-3.21.3-2.el7\_3

**NVD Reference:** CVE-2016-2834

**Vendor Reference:** RHSA-2016.2779

For Ops and Security

For Developers

Jenkins

Jenkins > Peekr > #24 > Aqua Security Scanner

Back to Project  
Status  
Changes  
Console Output  
Edit Build Information  
Delete Build  
Aqua Security Scanner  
Previous Build  
Next Build

### Vulnerability Report: peekr/demo:latest

From Registry: Docker Hub

1 2 0 5.7  
HIGH MEDIUM LOW SCORE AVG.

The following vulnerabilities were found:

Name	File	Severity	Score	Publish Date
CVE-2016-2515	/usr/share/nginx/html/js/utils.js	high	7.8	2016-04-13
CVE-2015-7501	/usr/share/nginx/html/js/utils.js	high	7.8	2015-04-13

# Provide targeted fix advice

webapp/orders-nginx:latest **Current** 411.71 GB

5 (↑ + 1) High | 30 (↑ + 2) Med | 34 Low

Show less ^

webapp/nginx-centos-builder:latest **Base** 411.1 GB

4 High | 28 Med | 34 Low

Scan date: 2018-04-26 | 11:37:16 AM

Negligible Vulnerabilities: There are 2 such vulnerabilities.  Off

Hide Base Image Vulnerabilities: Base image has 68 vulnerabilities (incl. negligible).  On

Text search (vulr)

Vulnerability	Severity	Resource Type	Resource	Installed Version	Fix Version
> CVE-2016-2515	High	File	/tmp/src/utils.js		4.1.1

# Controls beyond vulnerabilities

⊕ Super User

+

⊕ CVE Blacklist

+

⊕ Package Blacklist

+

⊕ Required Packages

+

⊕ Vulnerability Severity

+

⊕ Vulnerability Score

+

⊕ SCAP

+

⊕ OSS Licenses Blacklist

+

⊕ Approved Base Images

+

⊕ Custom Compliance Checks

+

⊕ Sensitive Data

+

⊕ Malware

+

# Determine image acceptance state

Images > bpdockerlab/pii-data:1.0



Risk Vulnerabilities Resources Sensitive Data Malware Information Scan History Audit



**Image Is Disallowed**

Image scanned on 2018-04-10 | 20:05 PM



Rescan Image

Image Assurance



**Image Scan**

Completed



**Package Blacklist**

Passed



**Custom Compliance Checks**

Rejected



**CVE Blacklist**

Passed



**Malware**

Passed



**Sensitive Data**

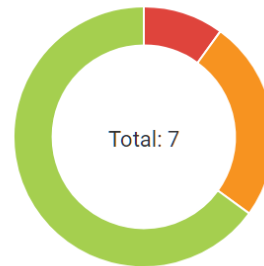
Passed

Details

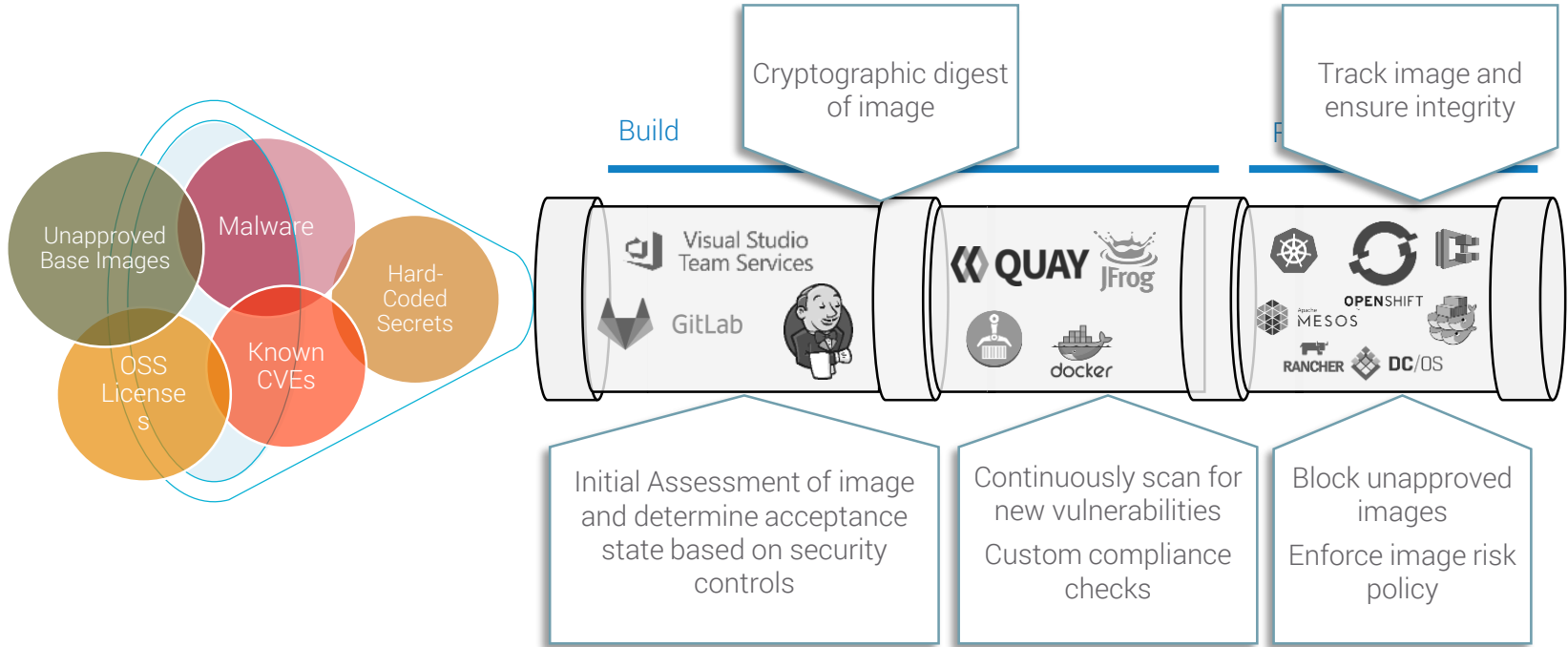
bpdockerlab/pii-data:1.0

Created about a year ago

High Medium None



# Automating DevSecOps



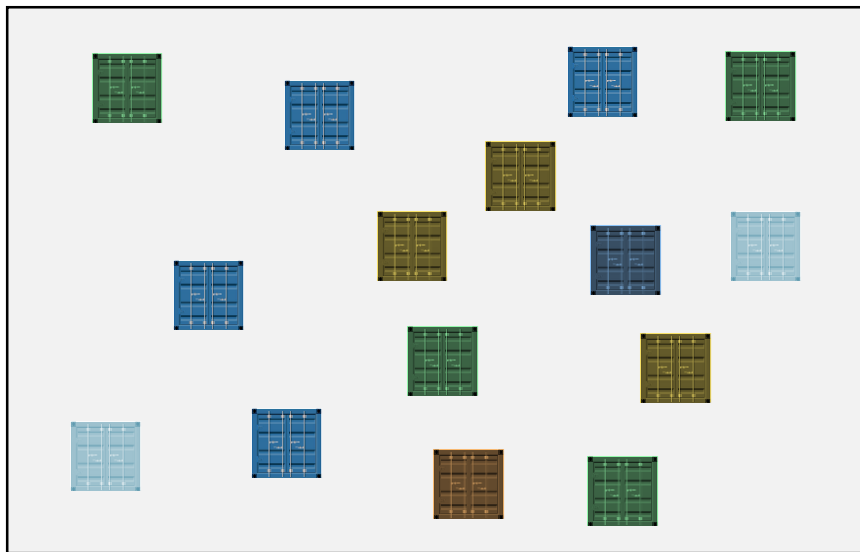
# Aligned with security practices

- Mandatory security in the pipeline
  - Actionable feedback for Developers (who are not security experts)
  - Lean base images, supplied internally
  - Multiple controls in CI and CD – Fail early, fail often
- Inventory and visibility
  - Keep track of artifact state at all times
  - Prevent changes after promotion
- Integration
  - Alignment with Enterprise toolset and apps:  
CI/CD, Orchestration, Logging, Ticketing, SIEM, Secrets Store, Cloud Provider, Container Platforms

# Image is good, what's next?

**What images  
are running?**

**Host user  
actions?**

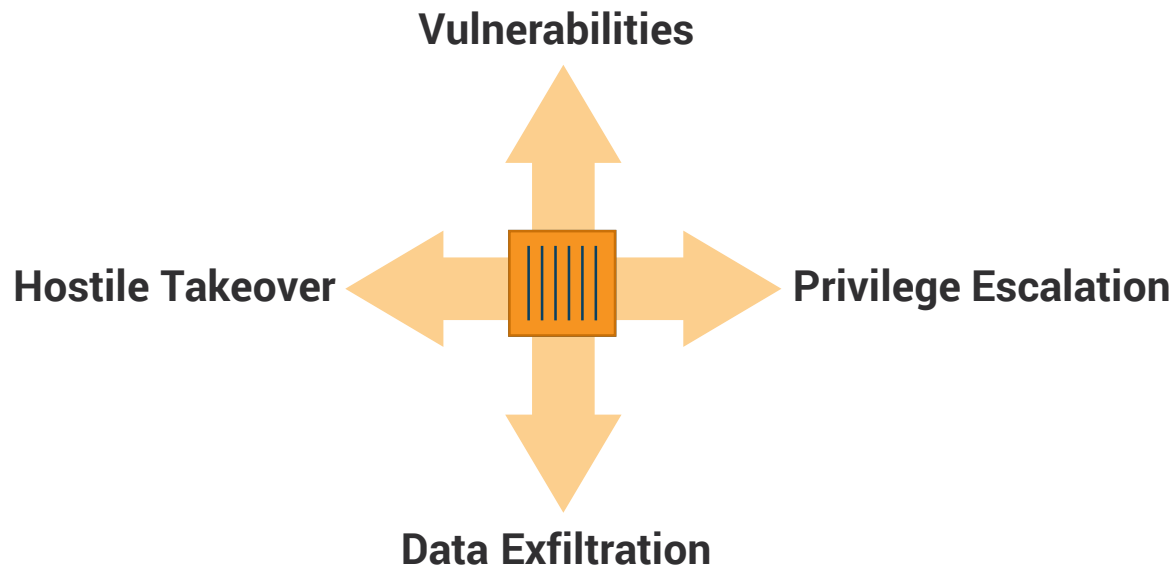


**Networking  
between  
containers?**

**Processes  
inside  
containers?**

**Patching?**

# Running containers are applications





# Visibility and search inside containers

## Containers



Hosts ▾ Vulnerabilities ▾ Status ▾ Registration Status ▾ Container Type ▾

More ▾

Container Type: Container

Name ▲

app-server

jenkins-3.2

orders-nginx

web-server

wp-db

wp-server

Search for vulnerabilities

CVE-2001-1228

CVE-2001-1534

CVE-2002-2439

CVE-2003-0097

CVE-2003-0166

CVE-2003-0860

CVE-2003-0861

demo128-vm11

CVE-2002-2439

Image Profile

None

None

orders-nginx-aquademio

None

None

None

Image Name

jboss/wildfly:10.0.0.Final

jenkins:latest

aquademio.azurecr.io/orders-nginx:2.0

httpd:2.4.28

mysql:8.0.0

wordpress:4.7.0-apache

# Best-practices for running containers

Status Enable Disable

Enforcement Mode Enforce Audit Only

---

Port Scanning Detection < ×

---

Prevent Override Default Configurations ∨ ×

Prevent running containers that override default configurations

- Running without default seccomp profile (seccomp=unconfined)
- Disabling SELinux separation (label:disable)
- Running with no apparmor security profile (apparmor=unconfined)

---

Drift Prevention ∨ ×

- Prevent running executable not in original image
- Prevent running container when image parameters are changed ⓘ

---

Block Unregistered Images < ×

**Available Runtime Policy Controls**

Add pre-defined controls for policy:











NIST ∨ ADD











---

To add control to the Runtime Policy, click the + button or drag and drop the control to the Runtime Policy area.

<span>+</span> IP Reputation	<span>+</span>
<span>+</span> Fork Guard	<span>+</span>
<span>+</span> Network Link	<span>+</span>
<span>+</span> Executable Blacklist	<span>+</span>
<span>+</span> Volumes Blacklist	<span>+</span>
<span>+</span> Limit New Privileges	<span>+</span>
<span>+</span> Limit Container Privileges	<span>+</span>
<span>+</span> Bypass Scope	<span>+</span>

# Specific controls

 Read-Only Directories and Files	
 Allowed Executables	
 Blacklisted Executables	
 Identity Inside The Container	
 Lockdown	

 Container Engine Controls	
 Volumes	
 Limits	
 Environment Variables	
 Restricted Volumes	

# Container controls applied at runtime

RESOURCE	ACCESS	TIME
/usr/bin/bash	exec	2016-05-25 11:52:55 AM
/usr/bin/dirname	exec	2016-05-25 11:52:55 AM
/usr/bin/basename	exec	2016-05-25 11:52:55 AM
/usr/bin/uname	exec	2016-05-25 11:52:55 AM
/usr/bin/grep	exec	2016-05-25 11:52:55 AM
/usr/lib/jvm/java/bin/java	exec	2016-05-25 11:52:55 AM
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.91-0.b14.el7 ...	exec	2016-05-25 11:52:55 AM


UID	NAME
1000	jboss


```
secdemo-4 / # docker exec -it -u root app bash
Permission denied
secdemo-4 / # docker exec -it app sh
sh-4.2$ ping
sh: /usr/bin/ping: Permission denied
sh-4.2$ cp
sh: /usr/bin/cp: Permission denied
sh-4.2$ yum
sh: /usr/bin/yum: /usr/bin/python: bad interpreter
sh-4.2$
```

# Securely distribute secrets

## Secrets

Define and edit secrets that you plan to use in your container environment

Enter Secret Name    Enter Secret Value        Enter Secret Description    Save Secret

Name	Value	Source	Description	Containers	Labels
db.password	*****	aqua		1	Select labels... 

NAME ^	IMAGE ⇅	HOST ⇅	STATUS ⇅
app	demo:444/myapp:1.0	secdemo-4	▶ Running

```
secdemo-4 / # docker run -d -e MYDB_ID=appdbuser -e MYDB_TOKEN=ToKeN -e MYDB_PWD={db.password} --name=app demo:444/myapp:1.0
dd94c492b55ee81af13dd7c590440c174d1839eabace28331fe3d3552d758f77
secdemo-4 / # docker inspect app | grep DB
    "MYDB_ID=appdbuser",
    "MYDB_PWD={db.password}",
    "MYDB_TOKEN=aqua-enc:7Ci9UEaZ1SE/sxiPxXT8iEBwiv5qz0oJKvKSTckukA0=",
secdemo-4 / # docker exec -it app bash -c set | grep DB
MYDB_ID=appdbuser
MYDB_PWD=MyNewValue
MYDB_TOKEN=ToKeN
```

# Container firewall built in

Service: Website

Service: Blog

Service: Orders

Blog

172.17.0.6

104.155.163.33

Clear Record Save Rules

PRIORITY	IP/CIDR	PORT RANGE	ALLOW/DENY
0	Service: Website	8000	Allow
1	Service: Blog	3306	Allow
2	Service: Orders	8500	Deny
3	104.155.163.33	80	Deny
4	172.17.0.6	3306	Allow

### Effective Container Firewall Rules

#### Outbound Networks


Policy	Priority	Destination IP/CIDR	Port Range	Allow/Deny
Blog	1	Service: Website	8000	Allow
Blog	2	Service: Blog	0-65535	Allow
Blog	3	Anywhere	0-65535	Deny
Default	4	Anywhere	0-65535	Allow

# Equipped to handle any threat


- Rogue container (e.g. bitcoin) → Block unapproved image
- Malicious code injection → Prevent image drift (=immutability)
- Unwanted admin actions → User access controls enforce least privilege
- Data exfiltration → Secured secrets; block unapproved network connections
- Network lateral movement → Container firewall stops unpermitted connections
- Unknown vectors (“zero days”) → Image drift prevention & Behavioral whitelisting – container can’t do what it wasn’t meant to do (executables, processes, files, volumes, host resources...)

# Security for the full container SDLC


## Build


 CI/CD Image Scan


## Ship


 Registry Image Scan

## Run

 Image Assurance


 Runtime Protection


 Container Firewall

 Threat Mitigation

 Secrets

 Host Scanning

 User Access Control

 Compliance



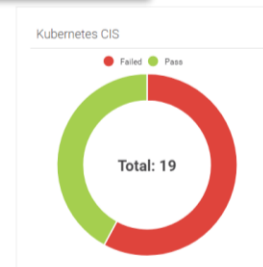
# Kubernetes & Docker CIS Benchmarks

- Runs checks against all 200+ CIS tests
- Provides a scored report of the results
- Can be scheduled to run daily

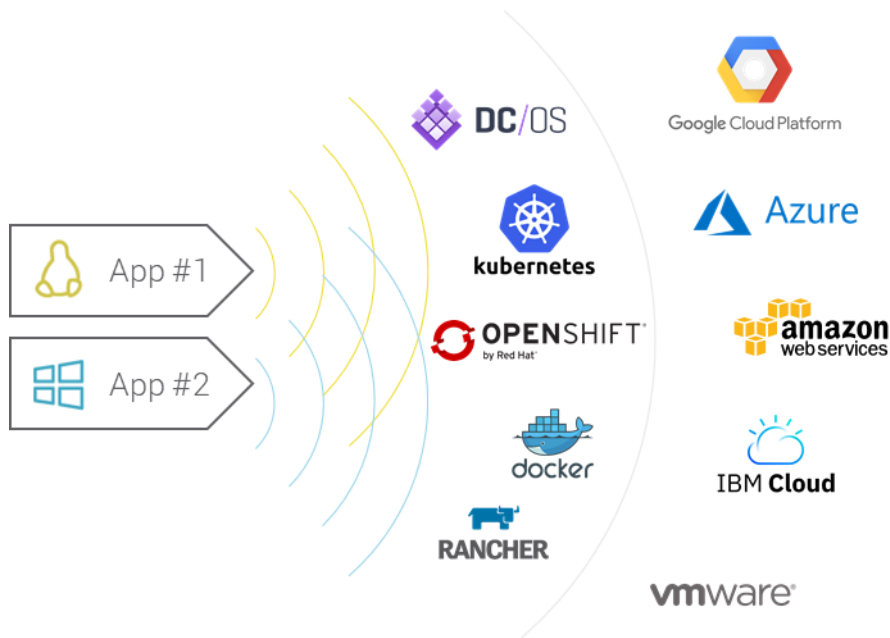
Aqua is a CIS SecureSuite member



HOST	LAST CHECK	FAIL	WARN	PASS	INFO
k8s-agentpool1-13666666-0.orwoeb1g1euxu14atgdvbarf.xx.internal.cloudapp.net	2018-03-01   04:00:17 PM	11	0	3	0
<b>2.1 Kubelet</b>					
2.1.1	Ensure that the --allow-privileged argument is set to false (Scored)	Fail	0	0	0
2.1.2	Ensure that the --anonymous-auth argument is set to false (Scored)	Pass	0	0	0
2.1.3	Ensure that the --authorization-mode argument is not set to AlwaysAllow (Scored)	Pass	0	0	0
2.1.4	Ensure that the --client-ca-file argument is set as appropriate (Scored)	Pass	0	0	0
2.1.5	Ensure that the --read-only-port argument is set to 0 (Scored)	Fail	0	0	0
2.1.6	Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Scored)	Fail	0	0	0
2.1.7	Ensure that the --protect-kernel-defaults argument is set to true (Scored)	Fail	0	0	0

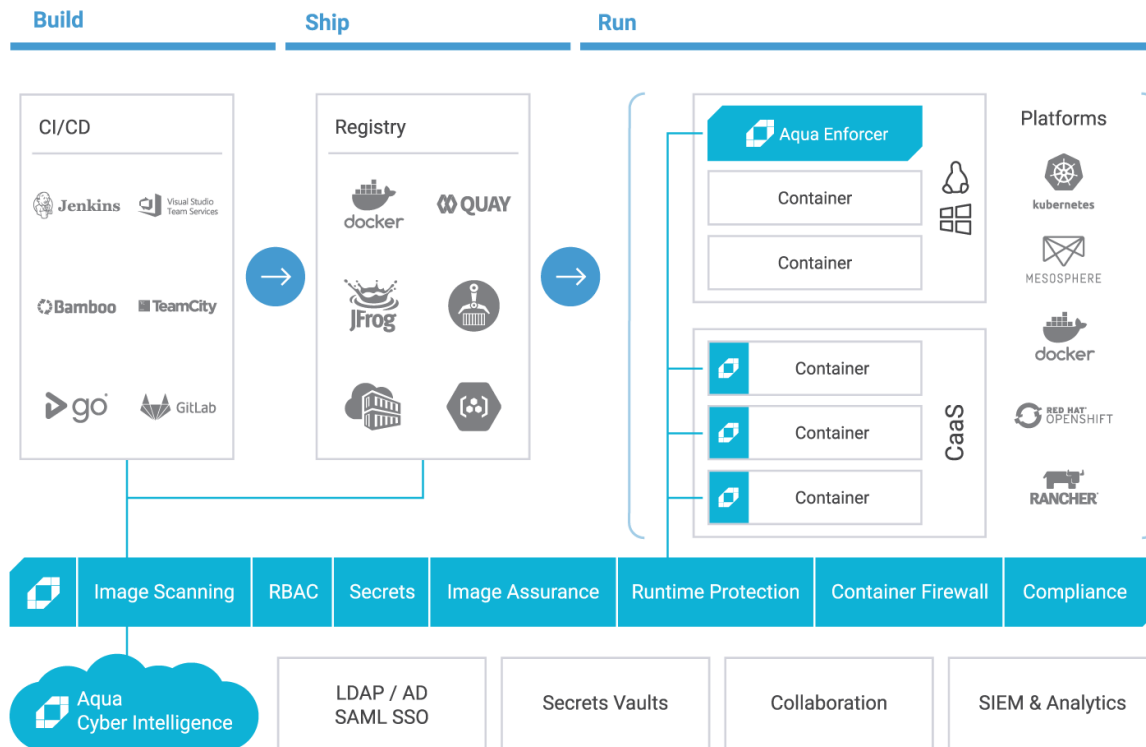


# Secure once, run anywhere



- ✓ Linux and Windows containers
- ✓ Any orchestrator: Kubernetes, OpenShift, DC/OS, Docker Swarm
- ✓ Cloud or On-Prem: AWS, Azure, GCP, IBM cloud, or VM environments
- ✓ CaaS: AWS Fargate and Azure ACI
- ✓ Multi-tenant management
- ✓ *Coming Soon:* Pivotal Cloud Foundry

# Multi-cloud deployment options



# For additional information

- Our Resource Center:  
[www.aquasec.com/resources/](http://www.aquasec.com/resources/)
- Container security Wiki:  
[www.aquasec.com/wiki](http://www.aquasec.com/wiki)
- Free community image scanner:  
<https://github.com/aquasecurity/microscanner>
- Partners and integrations:  
<https://www.aquasec.com/partners/>

[www.aquasec.com](http://www.aquasec.com)

---

