Securing Applications in Kubernetes

Tsvi Korren, CISSP

April 2017

CONTAINERS HAVE SECURITY IMPLICATIONS

Risk posture of the images is not understood

Where should security fit in the process

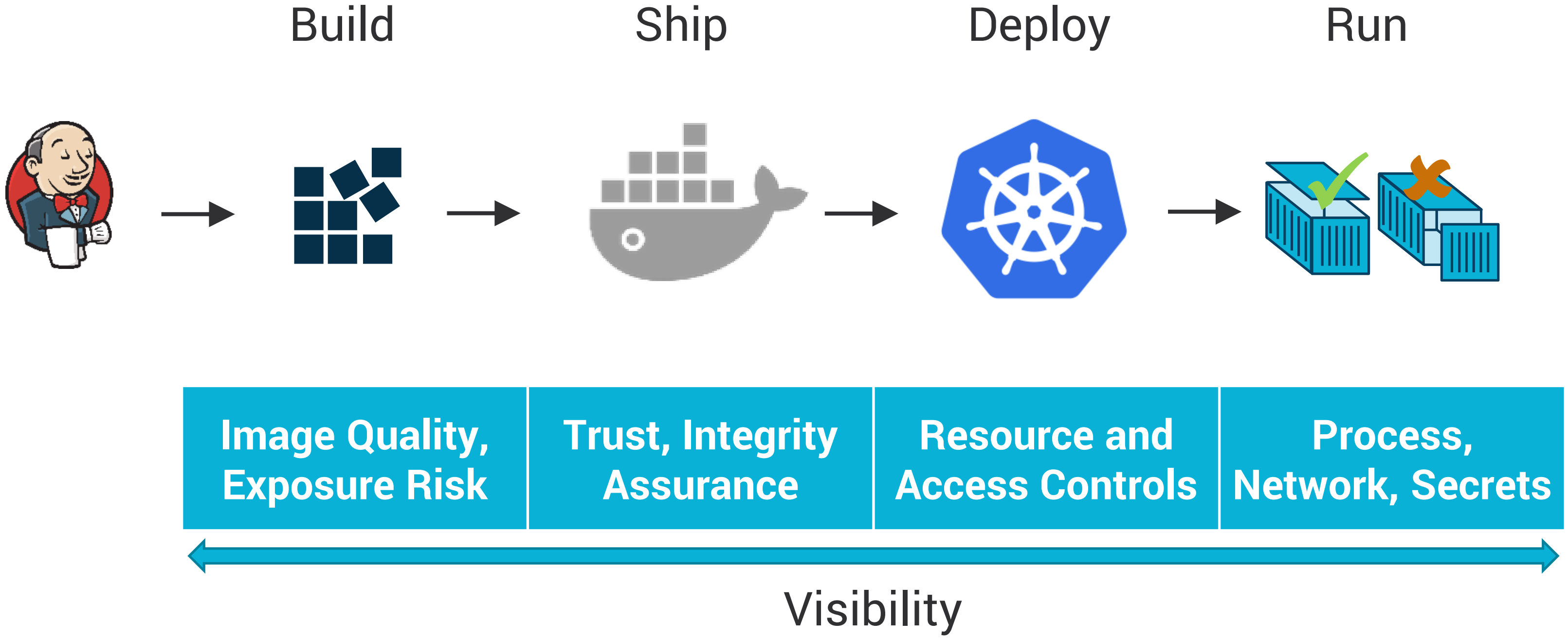
Containers are not visible with current security tools

Open source and other external components used

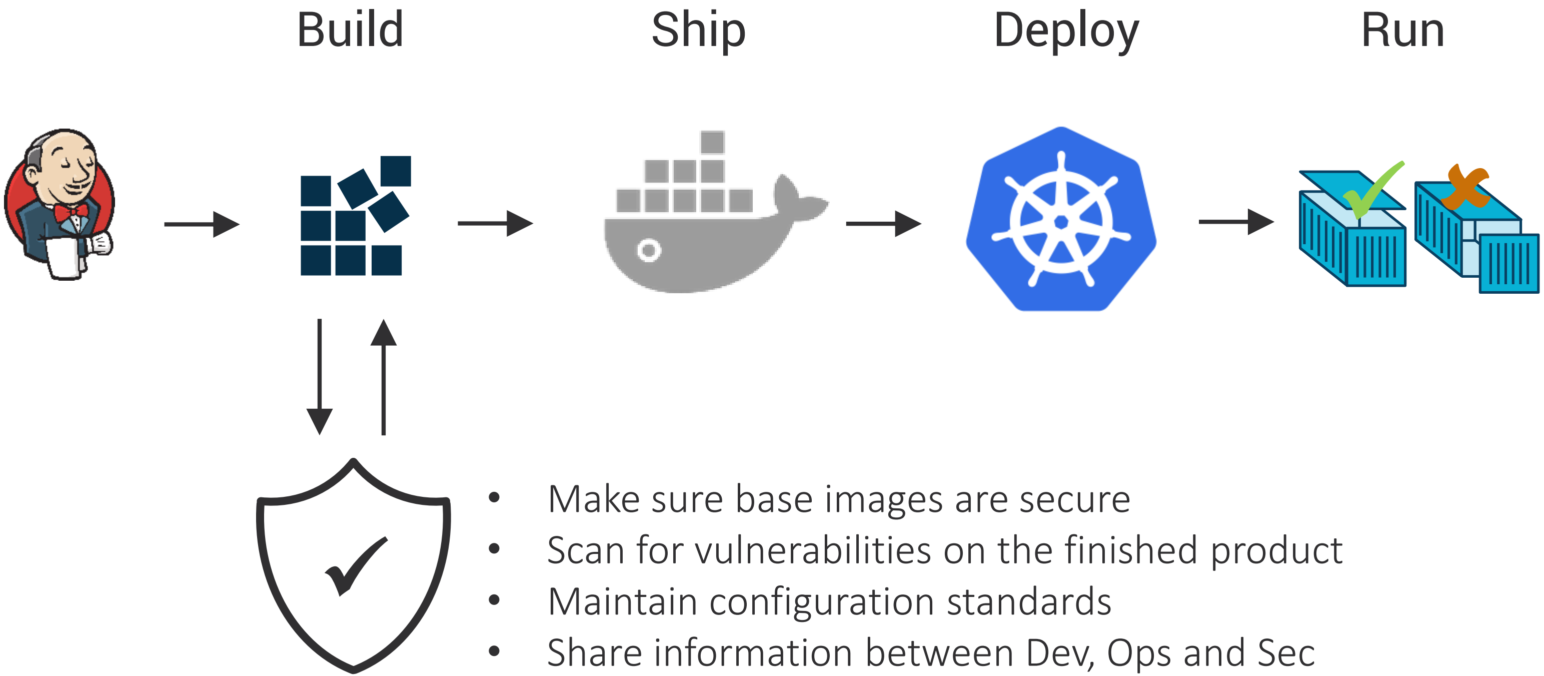
WHAT SECURITY PEOPLE WANT?

- Safe images, from trusted sources, tamper-proof
- Common security practices in the container environment
- Networking segmentation
- Safeguard sensitive data
- Accountability and audit data of container usage
- Data for demonstrating compliance

SECURE EACH STEP IN THE CONTAINER LIFECYCLE

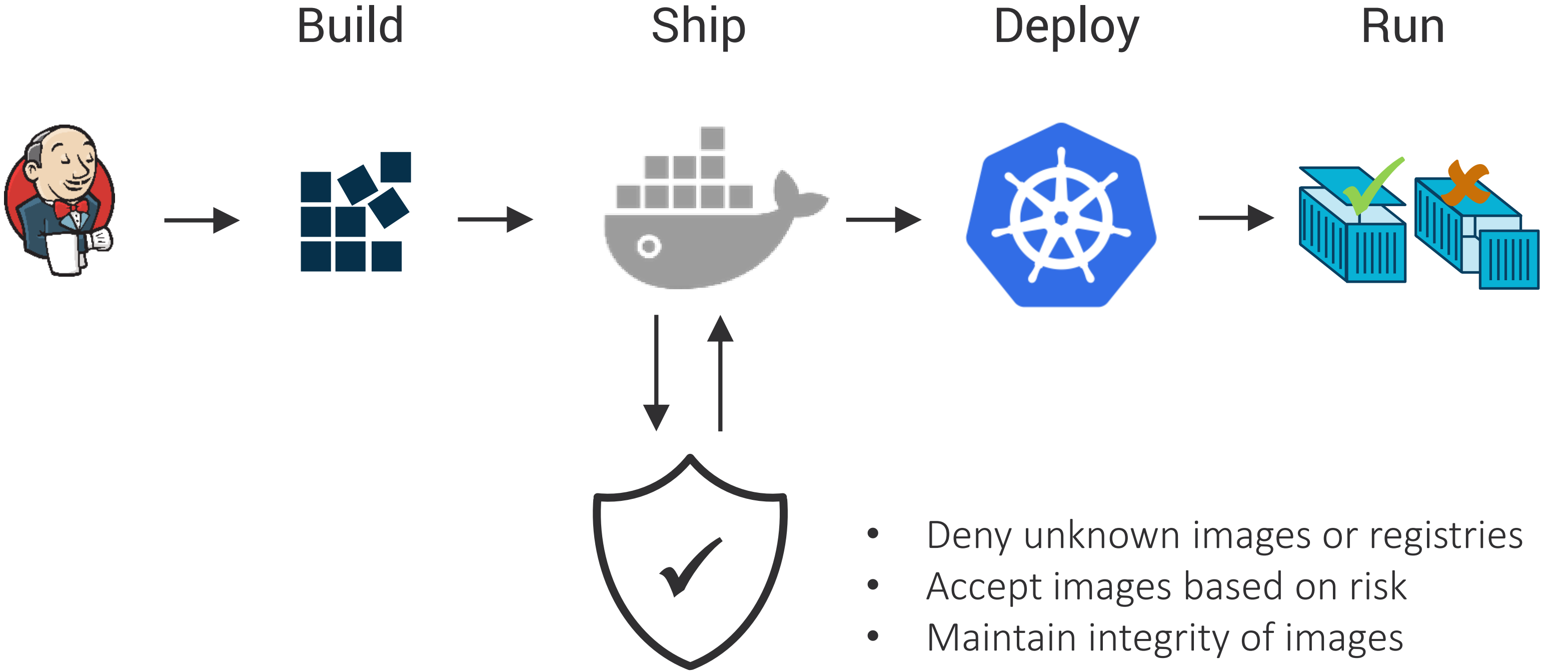

Build
Ship
Deploy
Run
Image Quality, Exposure Risk
Trust, Integrity Assurance
Resource and Access Controls
Process, Network, Secrets
Visibility

SECURITY STARTS IN THE BUILD PHASE



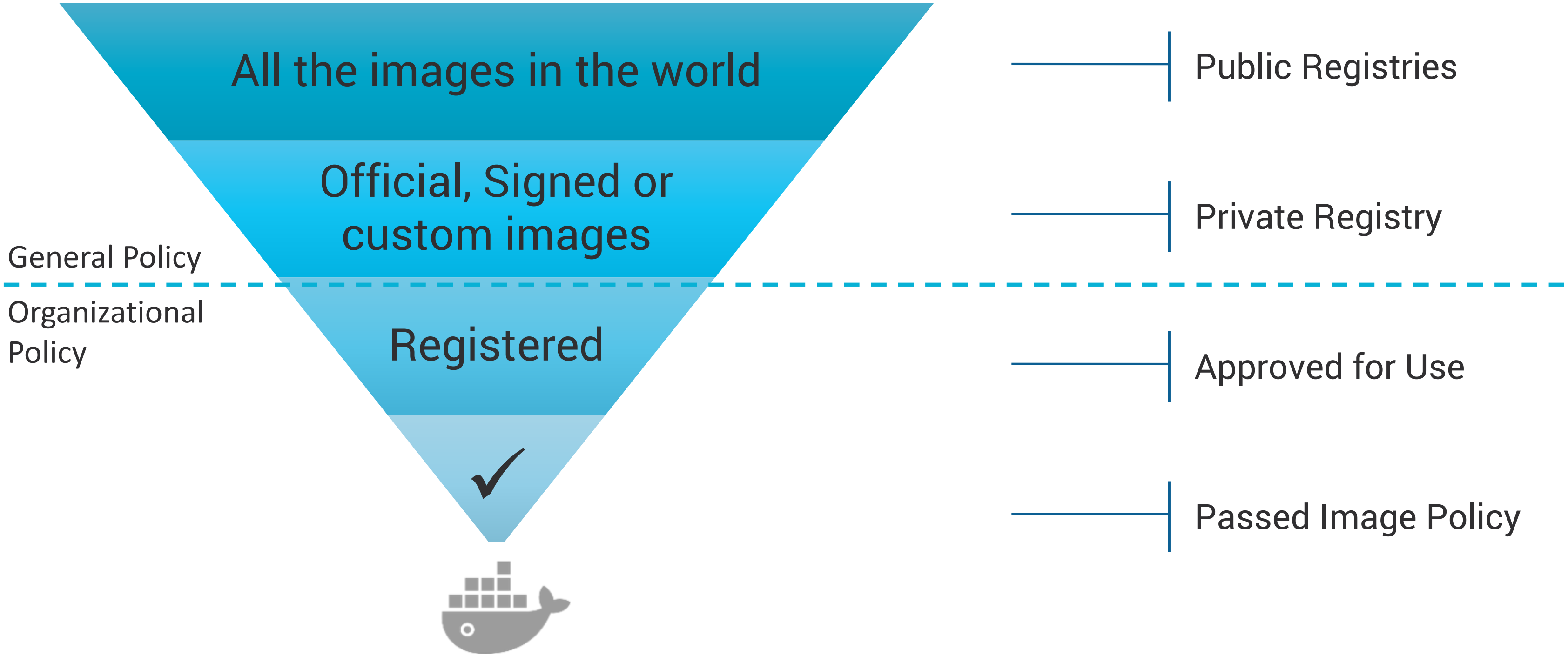
- Make sure base images are secure
- Scan for vulnerabilities on the finished product
- Maintain configuration standards
- Share information between Dev, Ops and Sec

ADD ENFORCEMENT OF IMAGE USAGE



- Deny unknown images or registries
- Accept images based on risk
- Maintain integrity of images

CONTROL THE INFLOW OF IMAGES



All the images in the world — Public Registries

Official, Signed or custom images — Private Registry

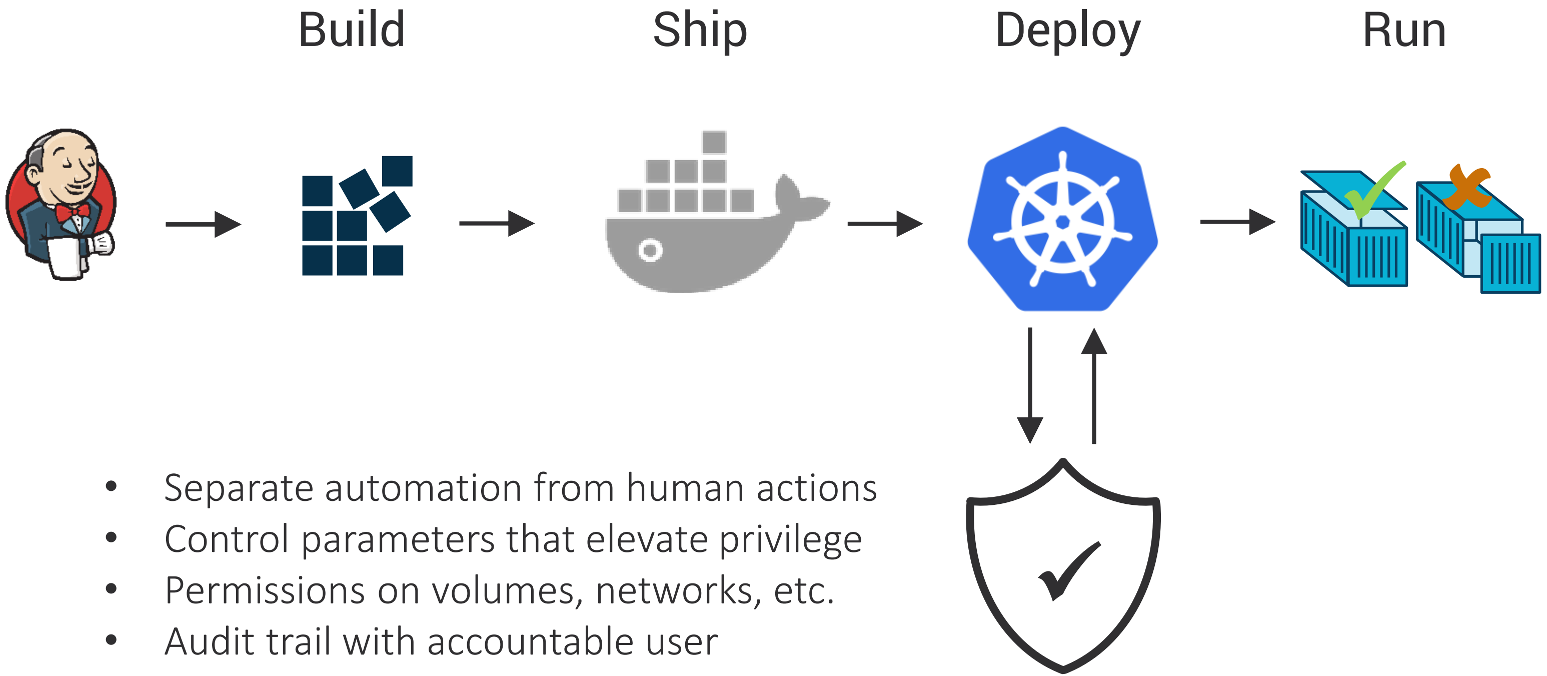
General Policy

Organizational Policy

Registered — Approved for Use

✓ — Passed Image Policy

LIMIT ACCESS TO CONTAINER ENGINE



- Separate automation from human actions
- Control parameters that elevate privilege
- Permissions on volumes, networks, etc.
- Audit trail with accountable user

PROTECT NODES DOCKER ENGINE

- Limit terminal access to Kubernetes nodes
- Keep the Docker API secure
- Use kubectl with proper authorization
  - To manage containers via pods, deployments…
  - To exec into a running container, if needed…
  - To query status via describe…

KUBERNETES AUTHORIZATION

- Enables define fine-grained-access controls on
  - Namespaces
  - Pods, Services, Containers
  - Operations
- Authorization plugins based on
  - ABAC model (attribute-based)
  - RBAC mode (role-based)

ADMINISTRATIVE BOUNDARIES

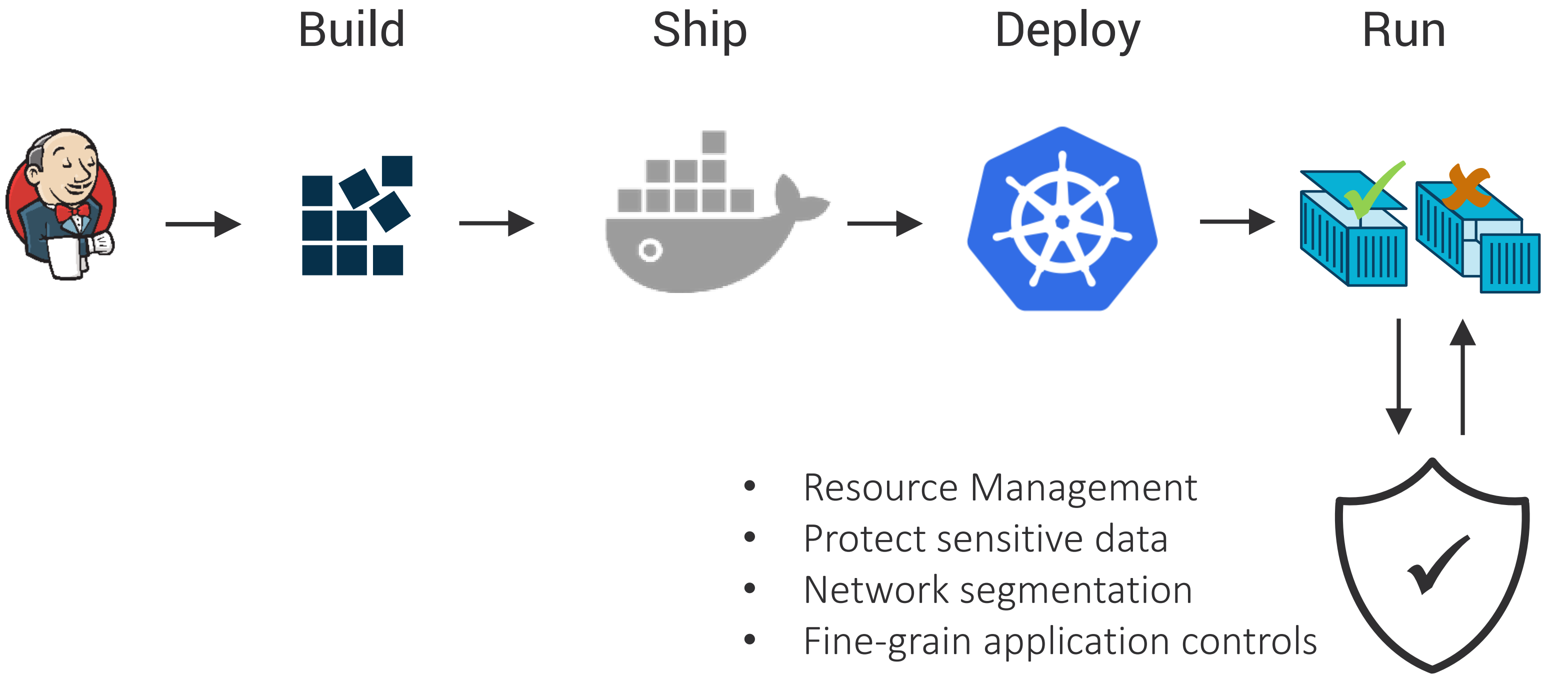
- Example: allow ‘alice’ to read pods from namespace ‘fronto’

```
{
        "apiVersion": "abac.authorization.kubernetes.io/v1beta1",
        "kind": "Policy",
        "spec": {
                "user": "alice",
                "namespace": "fronto",
                "resource": "pods",
                "readonly": true
        }
}
```

WHY AUTHORIZATION?

- Limits the damage of mistake or malicious intent
- Manage resources access by logical groups
- Resource segregation for multi-tenancy
- Compliance with regulations

GRANULAR CONTROLS OF RUNNING CONTAINERS



- Resource Management
- Protect sensitive data
- Network segmentation
- Fine-grain application controls

DEFINE RESOURCE QUOTA

- Avoid Resource-unbound containers in shared cluster
- Create resource quota policies
  - Pods
  - CPUs
  - Memory …
- Assigned to namespace

RESOURCE QUOTA EXAMPLE

- compute-resources.yaml

```
apiVersion: v1
kind: ResourceQuota
metadata:
        name: compute-resources
spec:
        hard:
                pods: "4"
                requests.cpu: "1"
                requests.memory: 1Gi
                limits.cpu: "2"
                limits.memory: 2Gi
```

- kubectl create -f ./compute-resources.yaml --namespace=myspace

WHY RESOURCE QUOTAS?

- Unbound resources add risk of DoS by runaway container
- By default, all resources are created unbound
- Predictable performance of clusters
- Required for capacity planning and disaster recovery

PROTECT SENSITIVE INFORMATION

- Storing sensitive data inside images or deployment definition is not safe
- Basic requirements for secret management
  - Put them in a vault while at rest, encrypted with key management
  - Restrict access to authorized users on authorized resources
  - Protect in transit and never commit to storage
  - Facilitate secrets expiry and rotation

KUBERNETES SECRETS

- Defined as objects consistent with the deployments
- Can be distributed as files or environment variables
- Cautions
  - Simple base64 encoding
  - Values accessible while at rest in etcd
  - No separation of duties: operator can see secret value
  - Secrets might persist on the node regardless of actual usage by containers

KUBERNETES SECRETS - EXAMPLE

```
$ echo -n "admin" | base64
YWRtaW4=
$ echo -n "1f2d1e2e67df" | base64
MWYyZDFlMmU2N2Rm
```

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=
  password: MWYyZDFlMmU2N2Rm
```

```
echo "MWYyZDFlMmU2N2Rm" | base64 --decode
1f2d1e2e67df
```

NETWORK SEGMENTATION

- Integration with external network enforcement points
- Kubernetes Network Policies work on pod-to-pod isolation (with only incoming traffic rules)
- Dynamic nature of container network identities makes container network segmentation a true challenge

IMPLEMENT NETWORK SEGMENTATION: EXAMPLE

```
apiVersion: extensions/v1beta1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
spec:
  ingress:
    -
      from:
        -
          namespaceSelector:
            matchLabels:
              project: myproject
        -
          podSelector:
            matchLabels:
              role: frontend
      ports:
        -
          port: 6379
          protocol: tcp
  podSelector:
    matchLabels:
      role: db
```

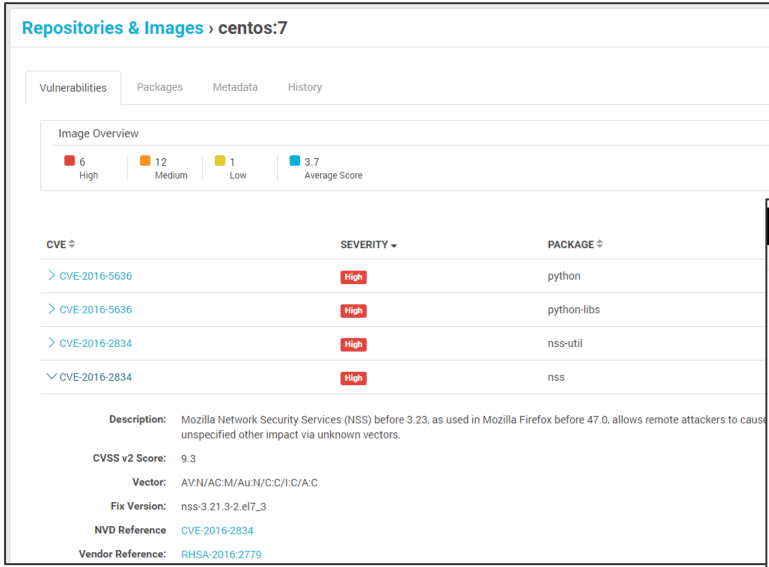
WHY NETWORK SEGMENTATION

- One compromised application is an door open into the cluster
- Ensures containers communicate based on required function
- Enables more co-locating of applications in the cluster
- Network segmentation is required for compliance

BEYOND KUBERNETES NATIVE CONTROLS

- Image Assurance
- User Access controls
- Application granular controls
- Secrets distribution
- Network segmentation

DETAILED IMAGE RISK INFORMATION



Repositories & Images › centos:7

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

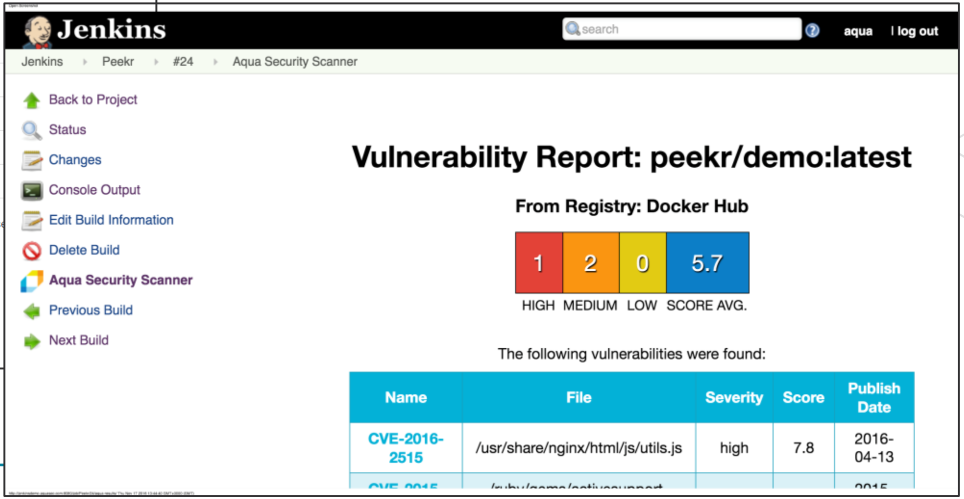
Vector: AV:N/AC:M/Au:N/C:C/I:C/A:C

Fix Version: nss-3.21.3-2.el7_3

NVD Reference CVE-2016-2834

Vendor Reference: RHSA-2016:2779

For Ops and Security



Vulnerability Report: peekr/demo:latest

From Registry: Docker Hub

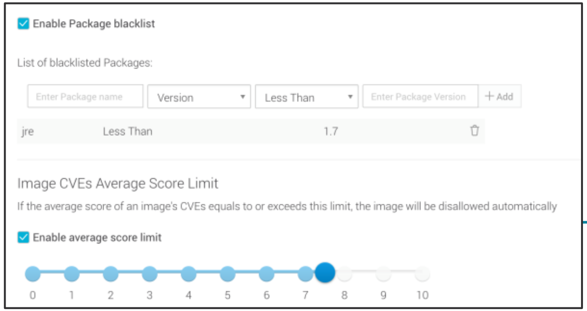
HIGH	MEDIUM	LOW	SCORE AVG.
1	2	0	5.7

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	/ruby/gems/activesupport			2015

For Developers

PREVENT UNAUTHORIZED IMAGES FROM RUNNING



☑ Enable Package blacklist

List of blacklisted Packages:

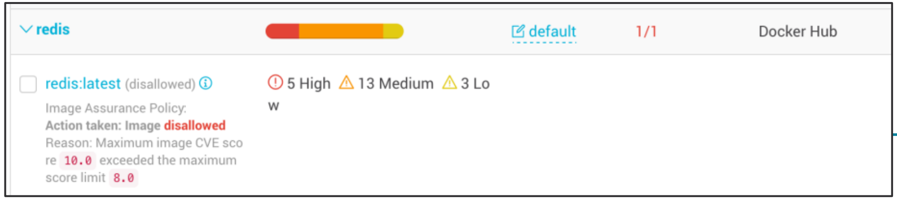
Enter Package name	Version ▾	Less Than ▾	Enter Package Version	+ Add
jre		Less Than	1.7	

Image CVEs Average Score Limit

If the average score of an image's CVEs equals to or exceeds this limit, the image will be disallowed automatically

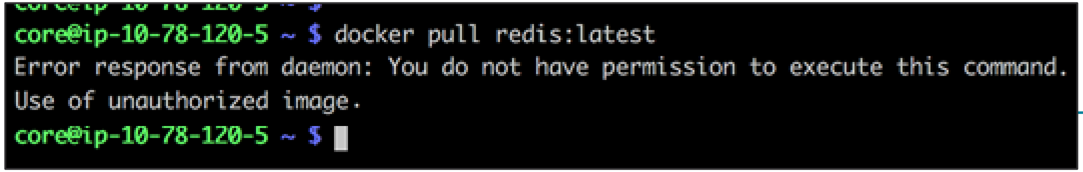
☑ Enable average score limit

Image Policy



∨ redis		default	1/1	Docker Hub
□ redis:latest (disallowed) ⓘ Image Assurance Policy: Action taken: Image disallowed Reason: Maximum image CVE score 10.0 exceeded the maximum score limit 8.0	ⓘ 5 High ⚠ 13 Medium ⚠ 3 Low			

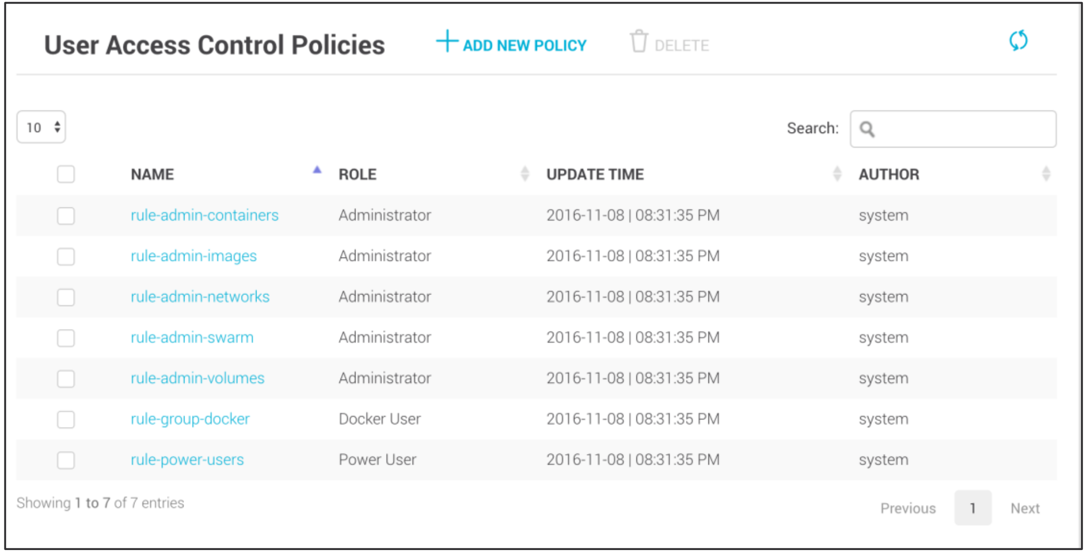
Registration



```
core@ip-10-78-120-5 ~ $
core@ip-10-78-120-5 ~ $ docker pull redis:latest
Error response from daemon: You do not have permission to execute this command.
Use of unauthorized image.
core@ip-10-78-120-5 ~ $ █
```

Prevention

LIMIT HUMAN INTERACTION WITH AUTOMATION

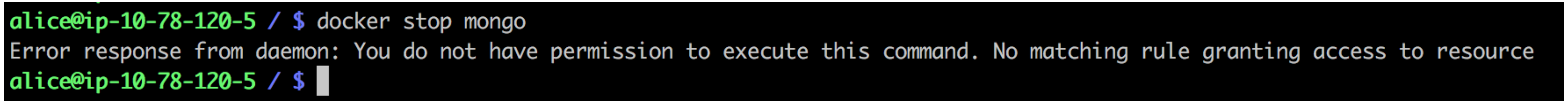


User Access Control Policies

□	NAME	ROLE	UPDATE TIME	AUTHOR
□	rule-admin-containers	Administrator	2016-11-08 | 08:31:35 PM	system
□	rule-admin-images	Administrator	2016-11-08 | 08:31:35 PM	system
□	rule-admin-networks	Administrator	2016-11-08 | 08:31:35 PM	system
□	rule-admin-swarm	Administrator	2016-11-08 | 08:31:35 PM	system
□	rule-admin-volumes	Administrator	2016-11-08 | 08:31:35 PM	system
□	rule-group-docker	Docker User	2016-11-08 | 08:31:35 PM	system
□	rule-power-users	Power User	2016-11-08 | 08:31:35 PM	system

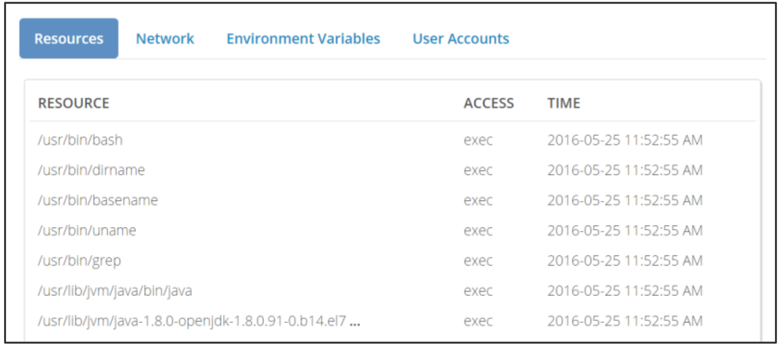
Showing 1 to 7 of 7 entries

Previous 1 Next

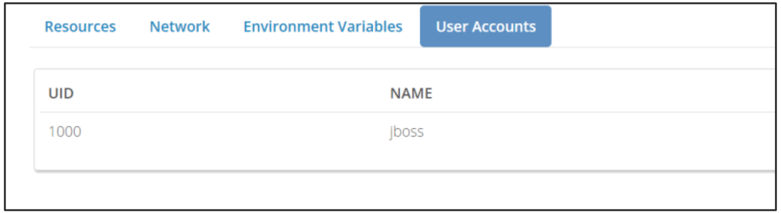


```
alice@ip-10-78-120-5 / $ docker stop mongo
Error response from daemon: You do not have permission to execute this command. No matching rule granting access to resource
alice@ip-10-78-120-5 / $
```

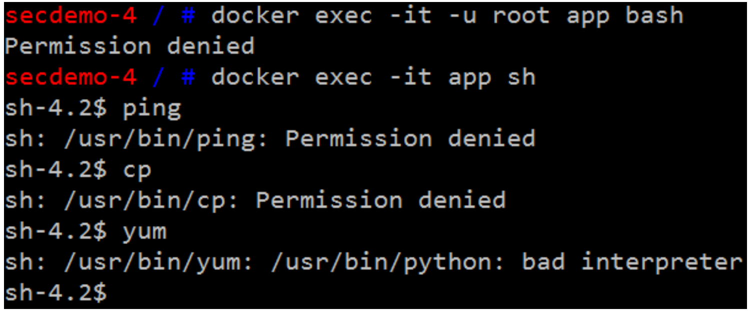
APPLICATION GRANULAR CONTROLS



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

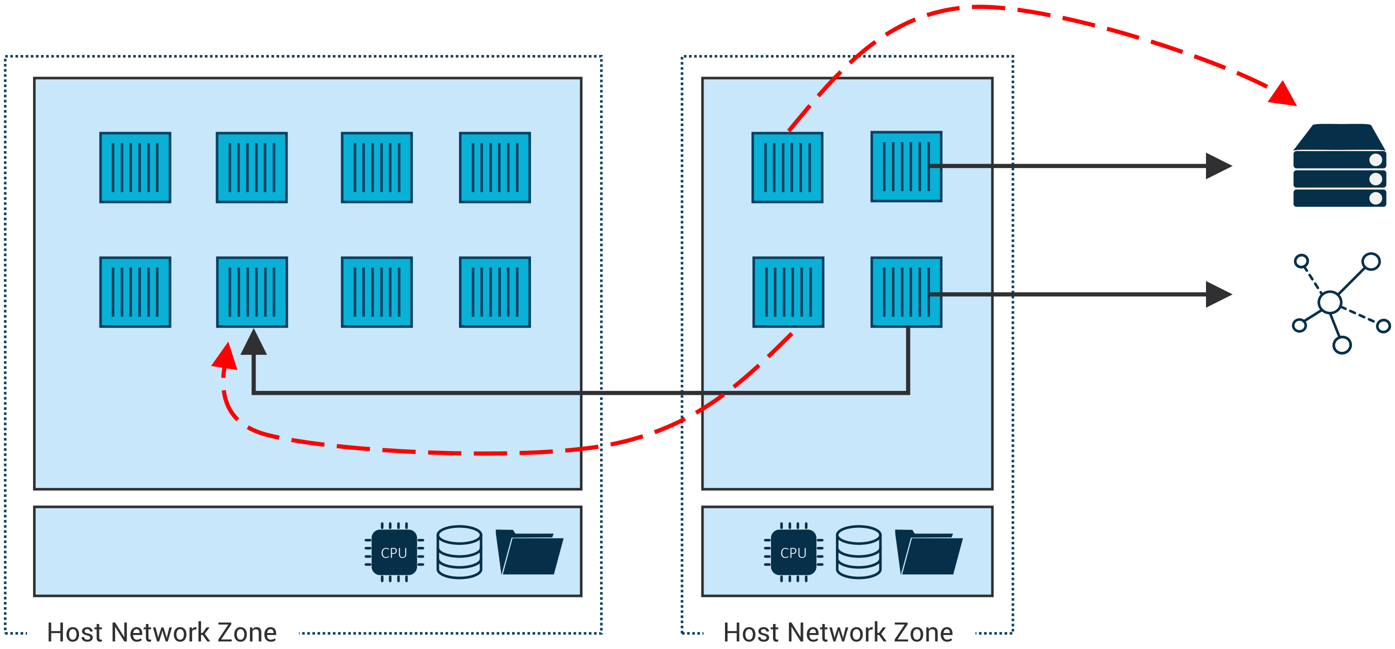


UID	NAME
1000	jboss

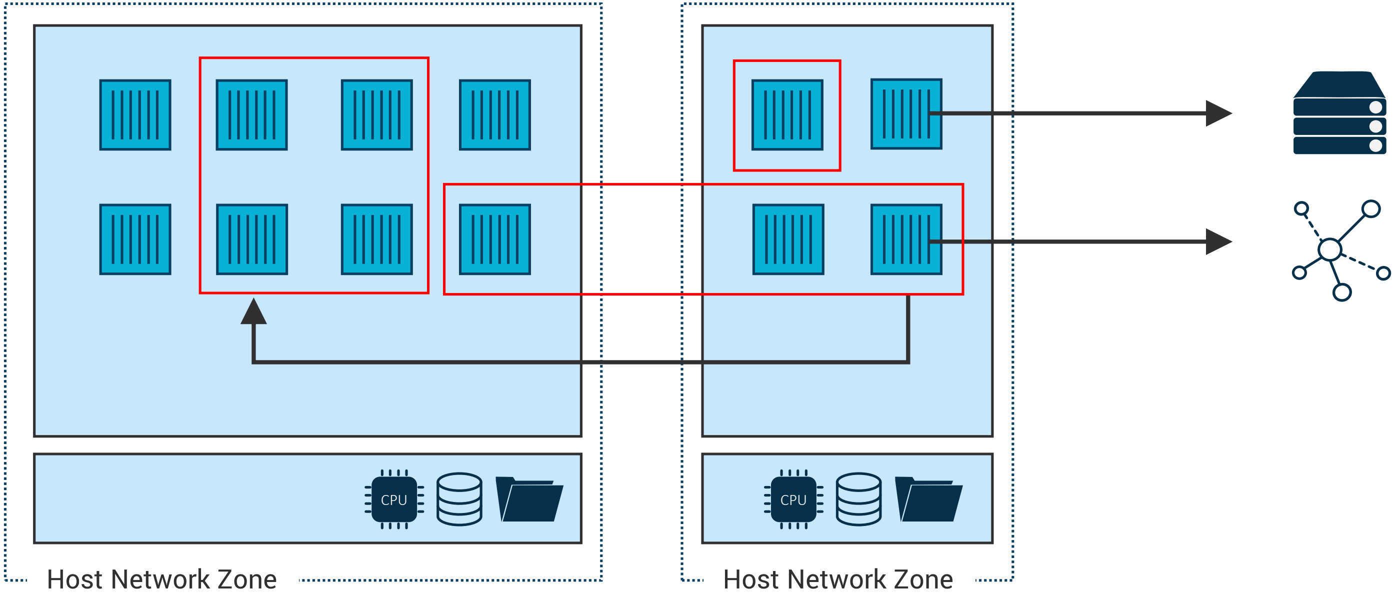


```
secdemo-4 / # docker exec -it -u root app bash
Permission denied
secdemo-4 / # docker exec -it app sh
sh-4.2$ ping
sh: /usr/bin/ping: Permission denied
sh-4.2$ cp
sh: /usr/bin/cp: Permission denied
sh-4.2$ yum
sh: /usr/bin/yum: /usr/bin/python: bad interpreter
sh-4.2$
```

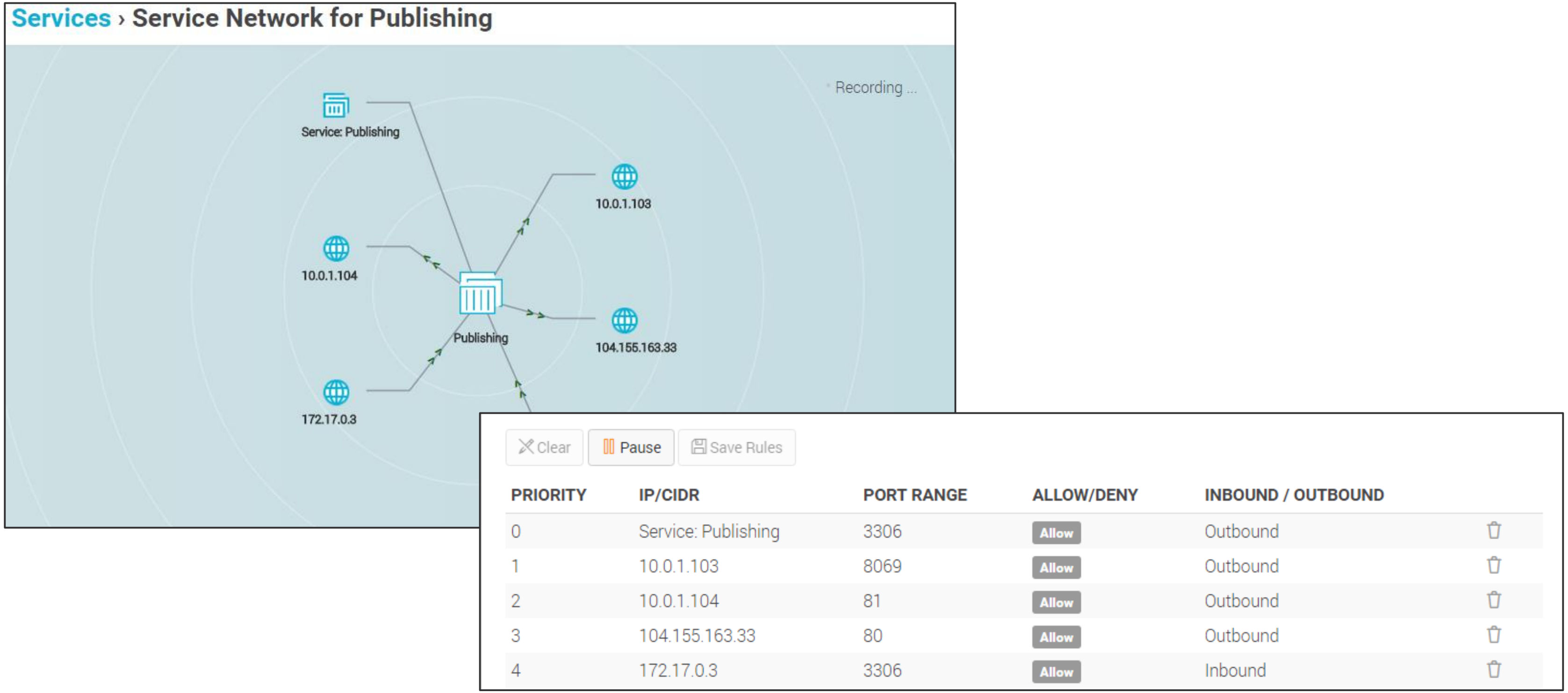
HOST NETWORK ZONES ARE NOT ENOUGH



NEED FOR CONTAINER-SPECIFIC NETWORK ZONES



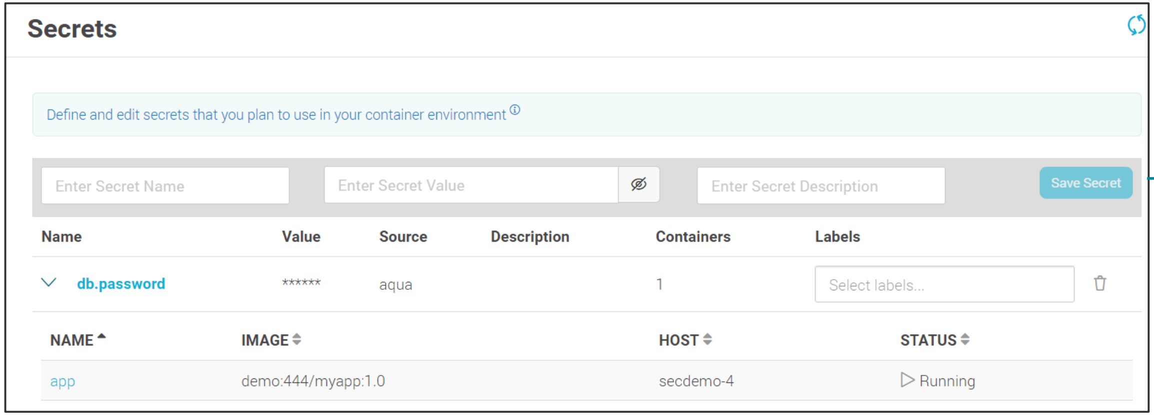
LEARN AND APPLY FIREWALL RULES



Services › Service Network for Publishing

PRIORITY	IP/CIDR	PORT RANGE	ALLOW/DENY	INBOUND / OUTBOUND
0	Service: Publishing	3306	Allow	Outbound
1	10.0.1.103	8069	Allow	Outbound
2	10.0.1.104	81	Allow	Outbound
3	104.155.163.33	80	Allow	Outbound
4	172.17.0.3	3306	Allow	Inbound

SECRETS MANAGEMENT



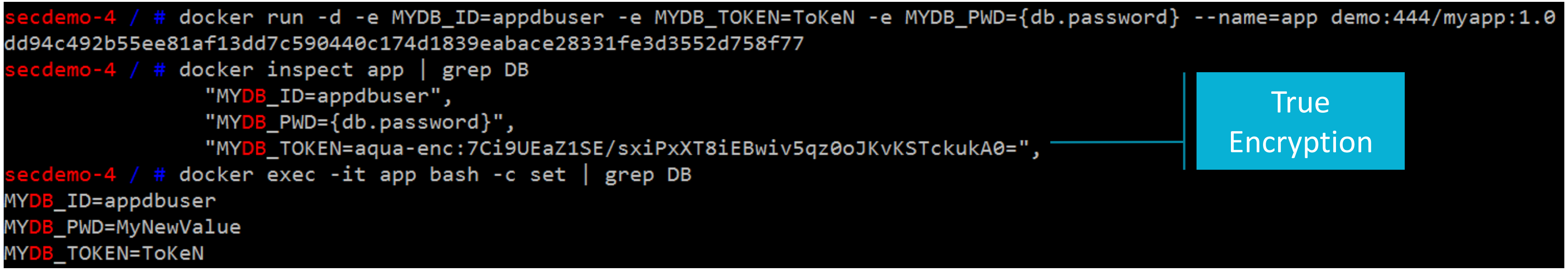
Secrets

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

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

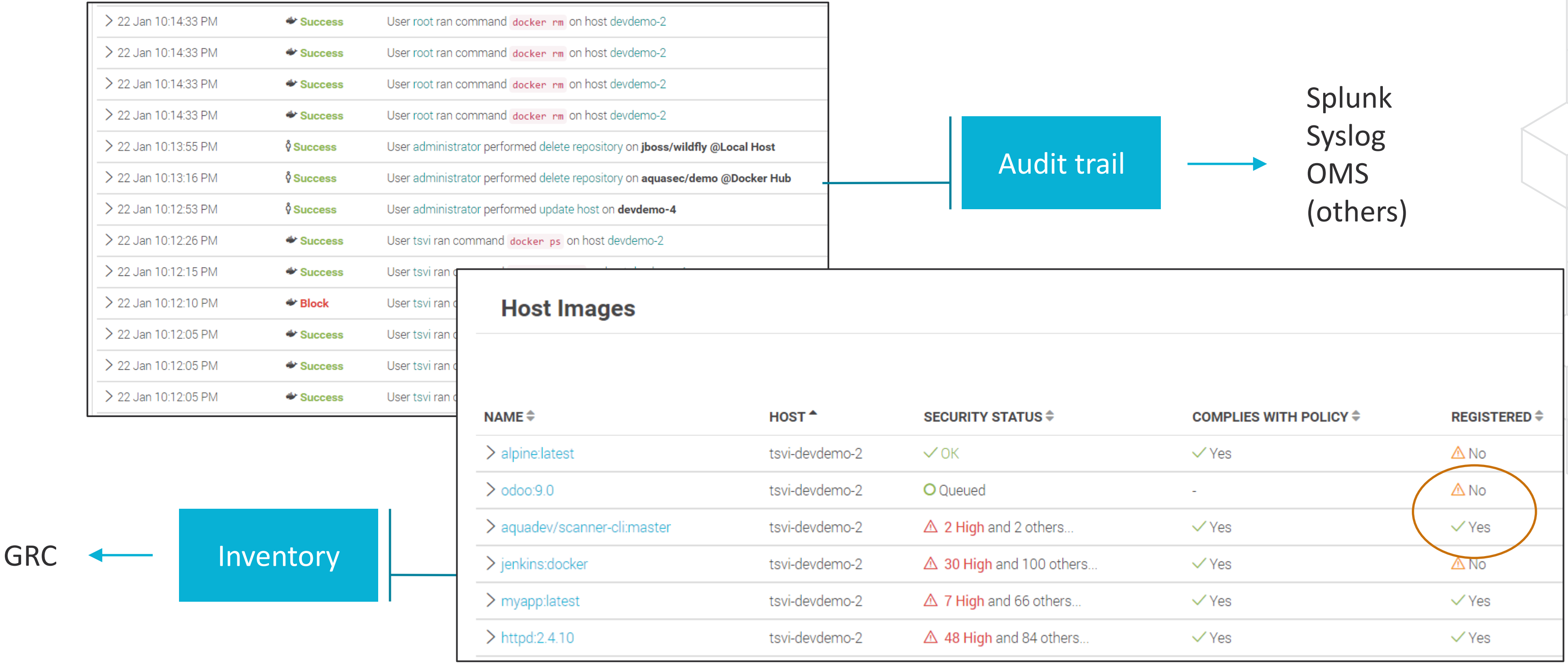
Define Secret



```
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
```

True Encryption

ADDED VISIBILITY



22 Jan 10:14:33 PM	Success	User root ran command `docker rm` on host devdemo-2
22 Jan 10:14:33 PM	Success	User root ran command `docker rm` on host devdemo-2
22 Jan 10:14:33 PM	Success	User root ran command `docker rm` on host devdemo-2
22 Jan 10:14:33 PM	Success	User root ran command `docker rm` on host devdemo-2
22 Jan 10:13:55 PM	Success	User administrator performed delete repository on **jboss/wildfly @Local Host**
22 Jan 10:13:16 PM	Success	User administrator performed delete repository on **aquasec/demo @Docker Hub**
22 Jan 10:12:53 PM	Success	User administrator performed update host on **devdemo-4**
22 Jan 10:12:26 PM	Success	User tsvi ran command `docker ps` on host devdemo-2
22 Jan 10:12:15 PM	Success	User tsvi ran
22 Jan 10:12:10 PM	Block	User tsvi ran
22 Jan 10:12:05 PM	Success	User tsvi ran
22 Jan 10:12:05 PM	Success	User tsvi ran
22 Jan 10:12:05 PM	Success	User tsvi ran

Audit trail → Splunk
Syslog
OMS
(others)

Splunk
Syslog
OMS
(others)

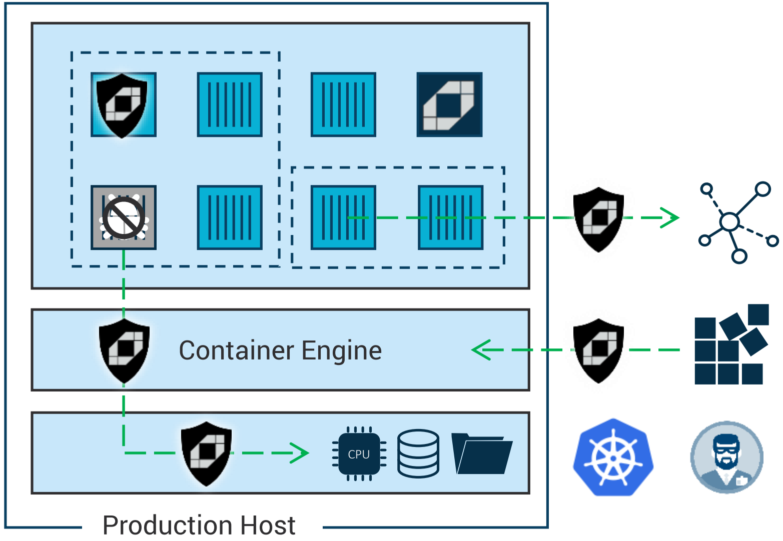
Host Images

NAME	HOST	SECURITY STATUS	COMPLIES WITH POLICY	REGISTERED
alpine:latest	tsvi-devdemo-2	✓ OK	✓ Yes	⚠ No
odoo:9.0	tsvi-devdemo-2	○ Queued	-	⚠ No
aquadev/scanner-cli:master	tsvi-devdemo-2	⚠ 2 High and 2 others...	✓ Yes	✓ Yes
jenkins:docker	tsvi-devdemo-2	⚠ 30 High and 100 others...	✓ Yes	⚠ No
myapp:latest	tsvi-devdemo-2	⚠ 7 High and 66 others...	✓ Yes	✓ Yes
httpd:2.4.10	tsvi-devdemo-2	⚠ 48 High and 84 others...	✓ Yes	✓ Yes

GRC ← Inventory

OPPORTUNITY FOR BETTER SECURITY

- Prevent unknown images
- Stop image by CVEs and score
- Stop user privilege escalation
- Stop suspicious processes
- Control capabilities
- Enforce network isolation
- Protect the host resources
- Encrypt sensitive variables
- Enforce use of automation tools
- Visibility across the environment



WWW.AQUASEC.COM