

JUNE 27, 2023



Container Security - Strengthening the Heart of your Operations

Kunal Verma

Community Manager at Kubesimplify
[@kverma_twt](https://twitter.com/kverma_twt)

Siddhant Khisty

Community Manager at WeMakeDevs
[@i_siddhant](https://twitter.com/i_siddhant)



@kverma_twt

Kunal Verma

Community Manager at Kubesimplify
DevOps & Open Source Advocate
Cloud Native Contributor

@i_siddhantk

Siddhant Khisty

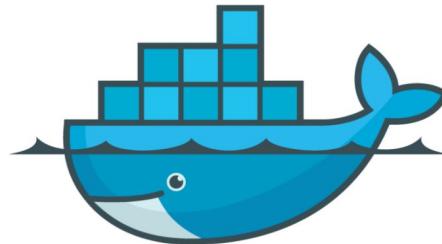
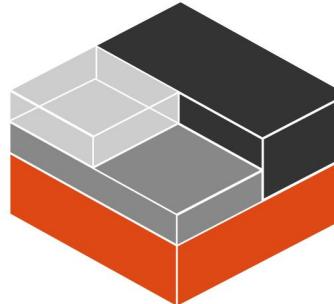
Community Manager at WeMakeDevs
Loves Tinkering with Computers
Linux and OSS Advocate



What are containers and why do they matter?

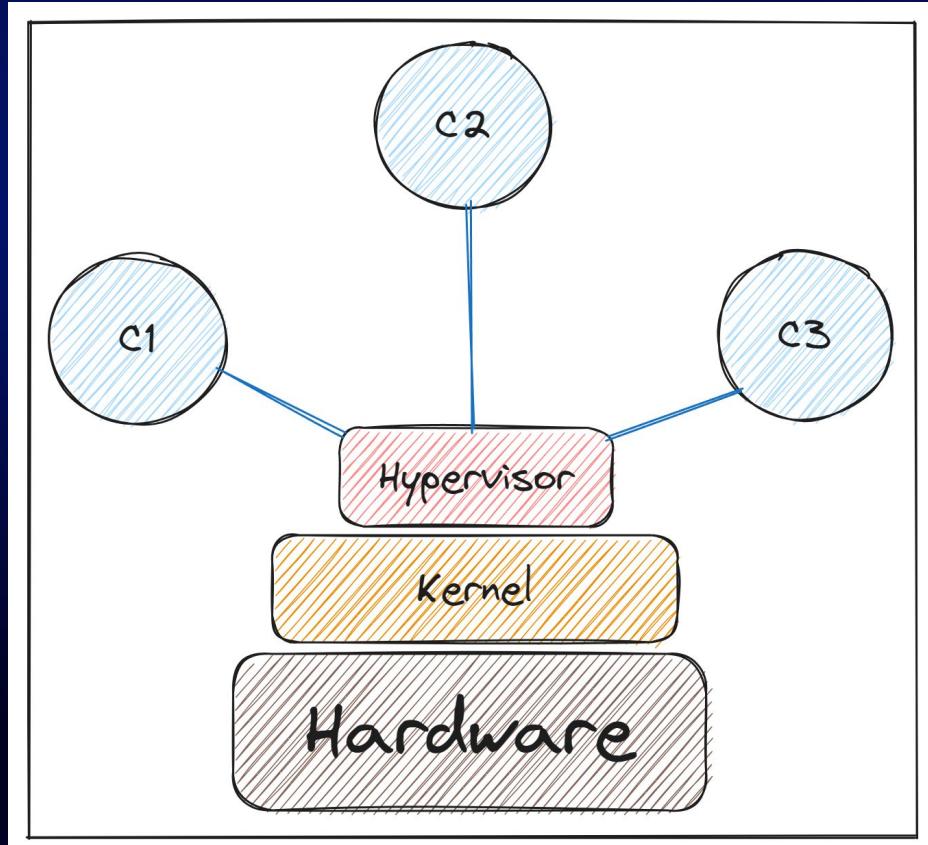
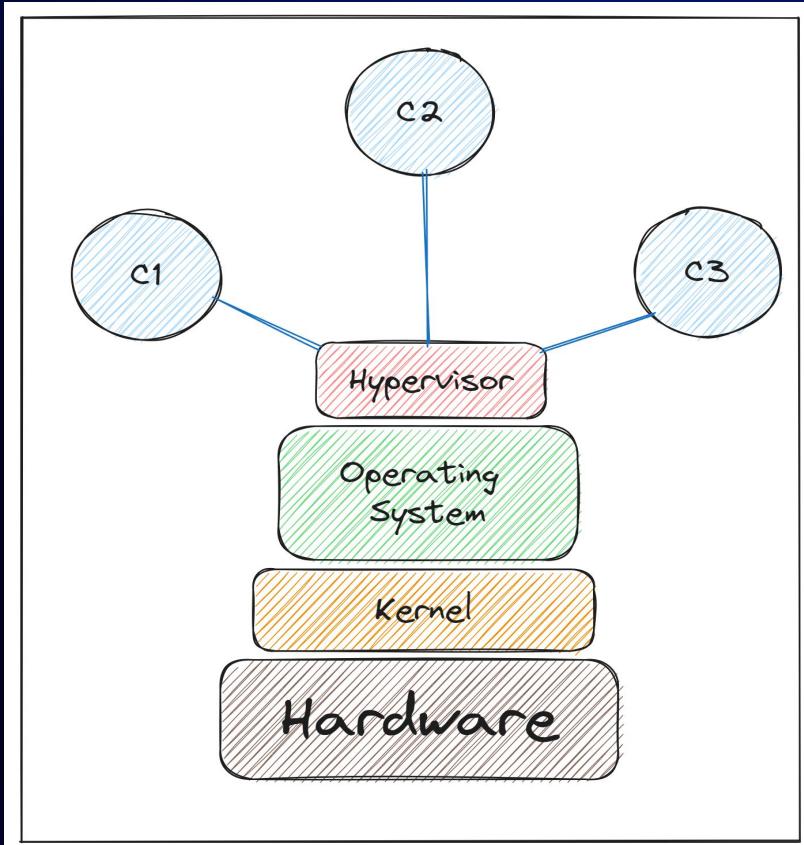


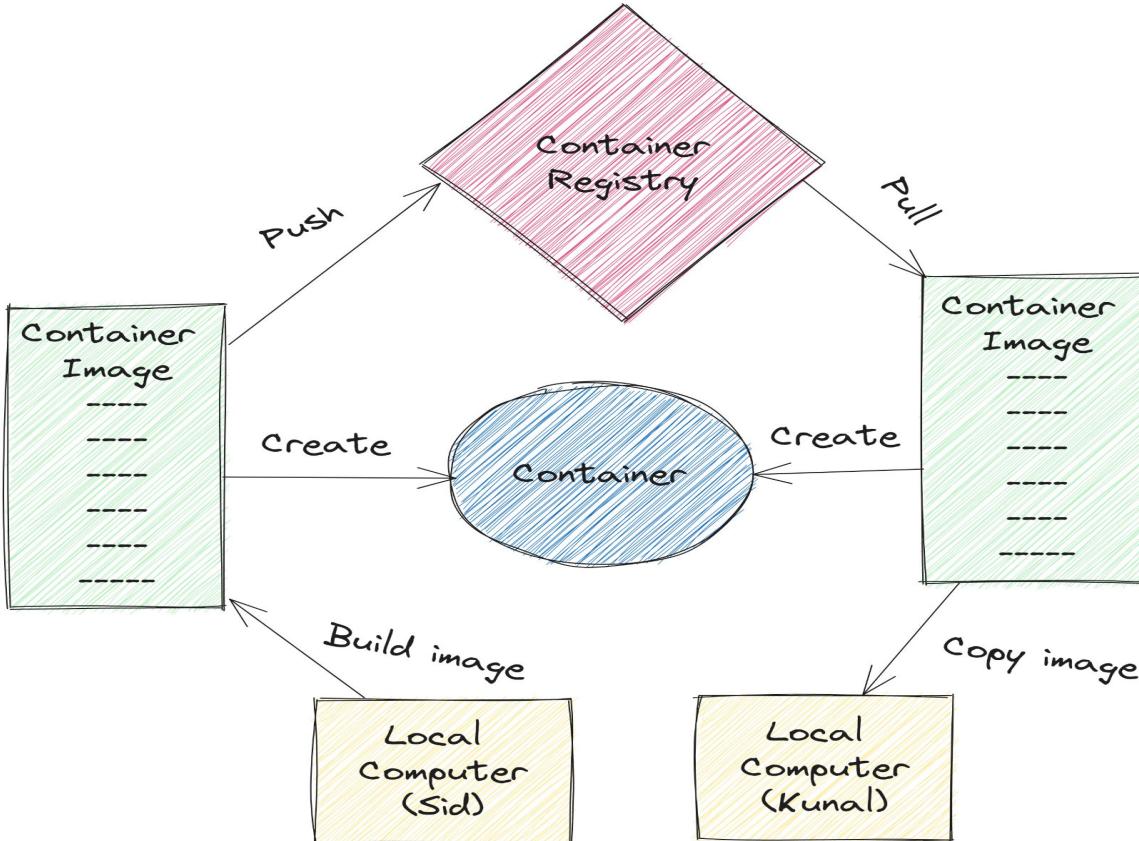
containerd



cri-o

Container Architecture





JUNE 27, 2023



Image Security

Securing Container Images – Why?

- **Goal:** Build a strong foundation
- Images == **Building blocks** of containerised apps
- **Insecure images** lead to:
 - Vulnerabilities
 - Malware
 - Outdated software components

Secure Container
Image

protection
against

Unauth Access
Data Breaches
Vulnerability Exploit
Malicious Code
Malware
Compliance Violation



How?

JUNE 27, 2023



5 Ques. Checklist

Question 1

Are the container images up-to-date?



Use ~~latest~~ stable releases



Check for available **security patches** or **updates**

```
1.25.1 , mainline , 1 , 1.25 , latest , 1.25.1-bookworm , m  
1.25-bookworm , bookworm
```

```
1.25.1-perl , mainline-perl , 1-perl , 1.25-perl , perl ,  
bookworm-perl , 1-bookworm-perl , 1.25-bookworm-perl , bo
```

```
1.25.1-alpine , mainline-alpine , 1-alpine , 1.25-alpine  
mainline-alpine3.17 , 1-alpine3.17 , 1.25-alpine3.17 , al
```

```
1.25.1-alpine-perl , mainline-alpine-perl , 1-alpine-perl ,  
perl , 1.25.1-alpine3.17-perl , mainline-alpine3.17-perl  
alpine3.17-perl , alpine3.17-perl
```

```
1.25.1-alpine-slim , mainline-alpine-slim , 1-alpine-sli  
slim , 1.25.1-alpine3.17-slim , mainline-alpine3.17-slim  
alpine3.17-slim , alpine3.17-slim
```

Question 2

Are container images scanned regularly?

```
james@ilmiontdesktop:~$ kubescape scan framework nsa
ARMO security scanner starting
[progress] Downloading/Loading framework definitions
[success] Downloaded/Loaded framework
[progress] Accessing Kubernetes objects
[success] Accessed successfully to Kubernetes objects, let's start!!!
[progress] Scanning cluster do-lon1-heronweb
[control: Allow privilege escalation] failed [!]
Description: Attackers may gain access to a container and uplift its privilege to enable excessive capabilities.
  Namespace kube-system
    DaemonSet - csi-do-node
  Namespace gitlab-managed-apps
    Deployment - ingress-nginx-ingress-controller
Summary - Passed:18  Warning:0  Failed:2  Total:20
Remediation: If your application does not need it, make sure the allowPrivilegeEscalation field of the securityContext is set to false.

[control: Allowed hostPath] failed [!]
Description: Mounting host directory to the container can be abused to get access to sensitive data and gain persistence on the host machine.
  Namespace kube-system
    DaemonSet - cilium
    DaemonSet - csi-do-node
```



Scanning for **vulnerabilities** using tools like **Kubescape**



Process to **address the discovered security issues**

Question 3

Are container images verified?

- ✓ From **reputable & trusted sources**
- ✓ Verify **integrity & authenticity** (before deploying)

Trusted Content

-  Docker Official Image [i](#)
-  Verified Publisher [i](#)
-  Sponsored OSS [i](#)

Question 4

Are container images signed?



Implement **image signing** using tools like **cosign**



Process to verify **digital signatures** & ensure integrity

```
Pushing signature to: gcr.io/dlorenz-vmtest2/demo:sha256-97fc222cee7991b5b061d4d4afdb5f3428fc0c9054e1690313786befa1e4e36.cosign
cosign:demo dlorenz$ cosign verify -key cosign.pub gcr.io/dlorenz-vmtest2/demo | jq .
{
  "Critical": {
    "Identity": {
      "docker-reference": ""
    },
    "Image": {
      "Docker-manifest-digest": "97fc222cee7991b5b061d4d4afdb5f3428fc0c9054e1690313786befa1e4e36"
    },
    "Type": "cosign container signature"
  },
  "Optional": null
}
cosign:demo dlorenz$ cat cosign.pub
-----BEGIN COSIGN PUBLIC KEY-----
wnf8gxaadcfpCZ9IZTy-W+qAWN3IjomsoISQyenGoWY=
-----END COSIGN PUBLIC KEY-----
cosign:demo dlorenz$
```

Question 5

Is a Zero-trust model being implemented?



- ✓ Follow a **zero-trust approach**
- ✓ Implement strict **access controls & auth. mechanisms**

JUNE 27, 2023



Image Registry

Securing Image Registries - Why?

- Secure registry == safeguard from **unauth. access** or **misuse**
- Reduce risk of **deploying compromised images**
- Ensure **compliance** & **trustworthiness**

Secure Image Registry

protection
against

Unauth Image Access
Image Tampering
Vulnerability Exploit
Malware Injection
Data Breaches
Compliance Violation



How?

JUNE 27, 2023



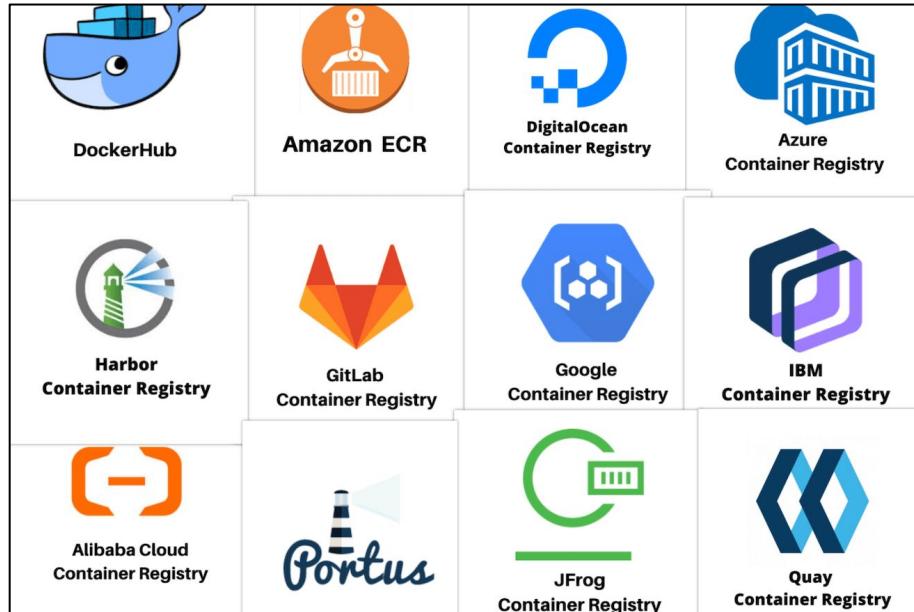
4 Ques. Checklist

Question 1

Is the image registry public or private?

✓ Use **private image registry**

✓ If public, **assess** the
potential risks + implement
security measures



Question 2

Is the image registry server secure?



- ✓ **Restricted network access** to the server
- ✓ Regular **patching, hardening & monitoring**

Question 3

Are access controls properly configured?

- ✓ **Granular access controls** for push, pull & modify images
- ✓ **Auth. mechanism** to allow only authenticated users



MANAGING
ACCESS
MANUALLY

USING
AWS IAM
FOR ACCESS
MANAGEMENT

Question 4

Are vulnerabilities being monitored?

```
bash-3.2$ trivy knqyf263/test-image:1.2.3
2019-05-13T15:19:03.912+0900  INFO  Updating vulnerability database...
2019-05-13T15:19:05.983+0900  INFO  Detecting Alpine vulnerabilities...
2019-05-13T15:19:06.000+0900  INFO  Updating pipenv Security DB...
2019-05-13T15:19:07.846+0900  INFO  Detecting npm vulnerabilities...
2019-05-13T15:19:07.949+0900  INFO  Updating pipenv Security DB...
2019-05-13T15:19:08.507+0900  INFO  Detecting pipenv vulnerabilities...
2019-05-13T15:19:08.508+0900  INFO  Updating bundler Security DB...
2019-05-13T15:19:09.574+0900  INFO  Detecting bundler vulnerabilities...
2019-05-13T15:19:09.575+0900  INFO  Updating cargo Security DB...
2019-05-13T15:19:10.441+0900  INFO  Detecting cargo vulnerabilities...
2019-05-13T15:19:10.441+0900  INFO  Updating composer Security DB...
2019-05-13T15:19:11.649+0900  INFO  Detecting composer vulnerabilities...

knqyf263/test-image:1.2.3 (alpine 3.7.1)
=====
Total: 26 (UNKNOWN: 0, LOW: 3, MEDIUM: 16, HIGH: 5, CRITICAL: 2)

+-----+-----+-----+-----+-----+
| LIBRARY | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION | TITLE
+-----+-----+-----+-----+-----+
| curl   | CVE-2018-14618  | CRITICAL | 7.61.0-r0          | 7.61.1-r0      | curl: NTLM password overflow
|       |                   |           |                  |               | via integer overflow
|       | CVE-2018-16839  | HIGH      | 7.61.1-r1          | 7.61.1-r1      | curl: Integer overflow leading
|       |                   |           |                  |               | to heap-based buffer overflow in
|       |                   |           |                  |               | curl_sas_create_plain_message()
|       | CVE-2019-3822    |           | 7.61.1-r2          | 7.61.1-r2      | curl: NTLM2 type-3 header
|       |                   |           |                  |               | stack buffer overflow
|       | CVE-2018-16840    |           | 7.61.1-r1          | 7.61.1-r1      | curl: Use-after-free when
|       |                   |           |                  |               | closing "easy" handle in
|       |                   |           |                  |               | curl_close()
|       | CVE-2018-16890  | MEDIUM    | 7.61.1-r2          | 7.61.1-r2      | curl: NTLM type-2 heap
|       |                   |           |                  |               | out-of-bounds buffer read
|       | CVE-2019-3823    |           | 7.61.1-r1          | 7.61.1-r1      | curl: SMTP end-of-response
|       |                   |           |                  |               | out-of-bounds read
|       | CVE-2018-16842  |           | 7.61.1-r1          | 7.61.1-r1      | curl: Heap-based buffer
|       |                   |           |                  |               | over-read in the curl tool
|       |                   |           |                  |               | warning formatting
|       | git    | CVE-2018-19486 | HIGH     | 2.15.2-r0          | 2.15.3-r0      | git: Improper handling of
|       |                   |           |                  |               | PATH allows for commands to be
|       |                   |           |                  |               | executed from...
```



Regularly monitor stored container images



Integrate vulnerability scanning tools such as **trivy**

JUNE 27, 2023



Build Process

Dockerfile

```
FROM ubuntu:20.04
```

```
ADD foo.txt /opt
```

```
RUN touch bar.md
```

...

copy from registry

copy from build dir

mount tmp bundle

/bin/sh -c
'touch bar.md'

image

ubuntu layer(s)

/opt/foo.txt

/bar.md

save diff to layer

The biggest security Red flag

```
docker-compose.yml
1  version: "alpine"
2  services:
3    web:
4      build: .
5      container_name: web-container
6      privileged: true
7      ports:
8        - "8080:8080"
9    golang:
10      image: "golang:alpine"
```



Attackers have a huge playground



Reduce your attack surface



Lightweight base images



Multi-stage builds

```
# Base image Stage 1
FROM ubuntu:16.04 as stage1

RUN apt-get update
RUN apt-get -y install make curl
RUN curl http://xyz.com/abc.tar.gz -O
RUN tar zxf abc.tar.gz && cd abc
RUN make DESTDIR=/tmp install

# Stage 2
FROM alpine:3.10

COPY --from=stage1 /tmp /abc

ENTRYPOINT [ "/abc/app" ]
```

Why Alpine?

```
ubuntu $ docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          16.04        b6f507652425  22 months ago  135MB
alpine          3.10         e7b300aee9f9   2 years ago   5.58MB
ubuntu $
```

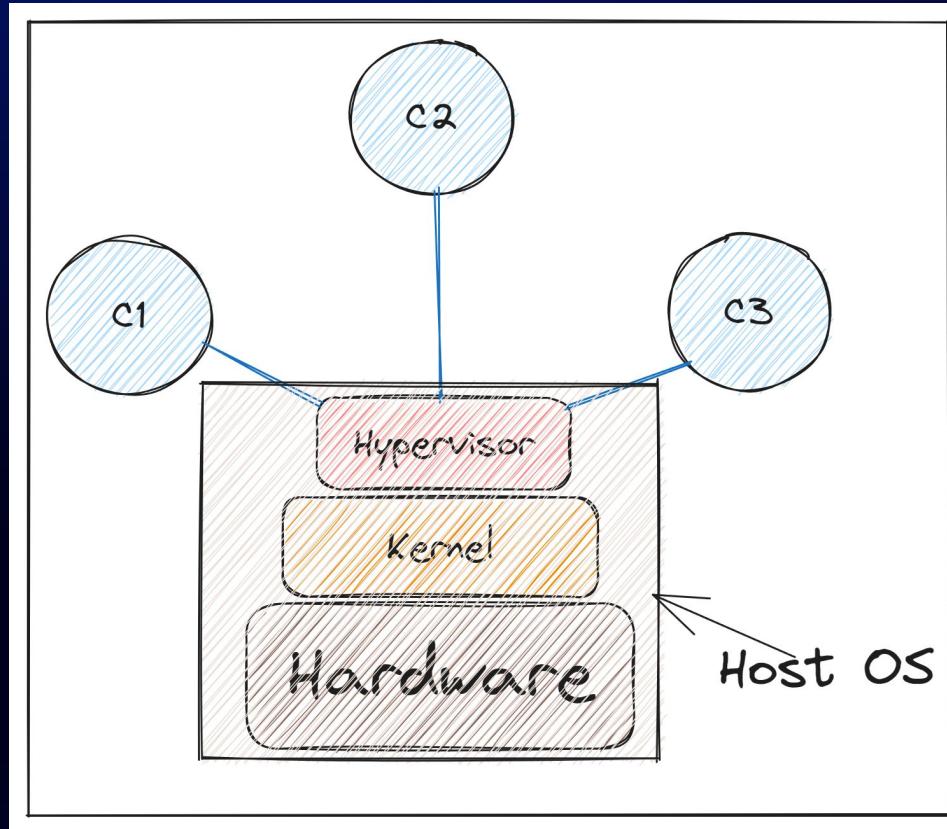
JUNE 27, 2023



Secure the Host

The greatest point of failure

Where do containers run?



Lightweight Host OS – Reduce Attack Surface



Set up robust access control

-  **Use firewalls to block access**
-  **Set robust access control**
-  **Scan for threats regularly**



Firewalld



SELinux



OpenVAS

JUNE 27, 2023



Tools

How to Choose?

– Parameters to Consider



- Vulnerability Scanning
- Image integrity & verification
- Runtime protection
- Compliance and Audit Capabilities



JUNE 27, 2023



Let's Summarize!

Takeaways

- Image security
- Registry security
- Secure the build process
- Host Security
- Tools to make life easier



Additional
Resources

JUNE 27, 2023



Thank You

Kunal Verma

Community Manager at Kubesimplify
@kverma_twt

Siddhant Khisty

Community Manager at WeMakeDevs
@i_siddhantk