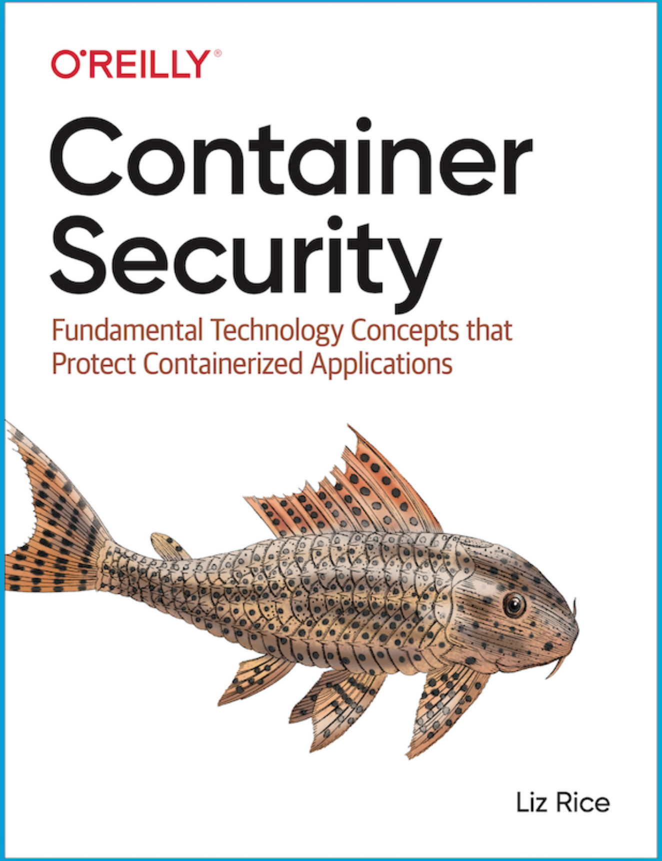
The Container Security Checklist

Liz Rice
VP Open Source Engineering, Aqua Security

@lizrice | @aquasecteam

containersecurity.tech



O'REILLY

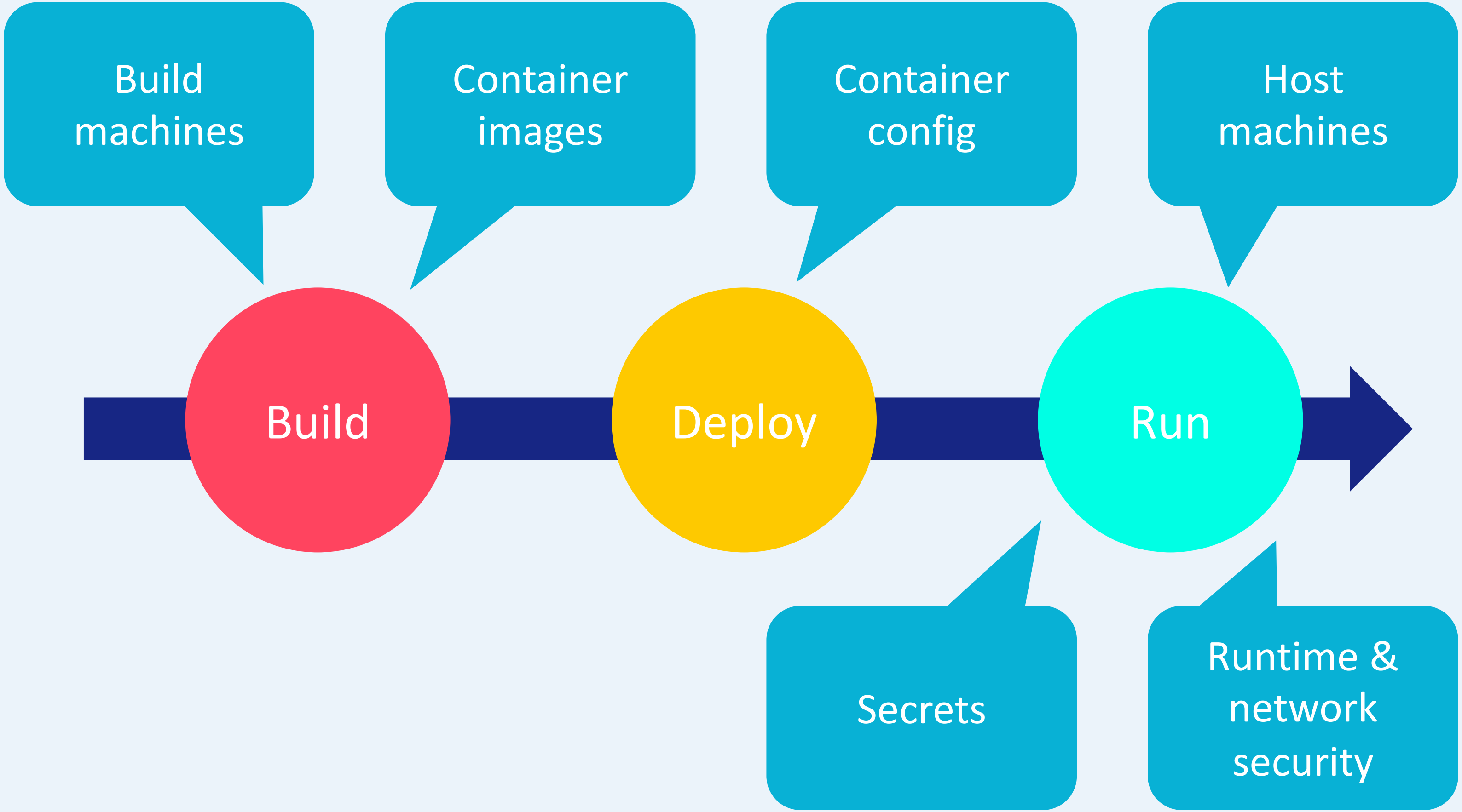
Container Security

Fundamental Technology Concepts that Protect Containerized Applications

Liz Rice

Appendix: Security Checklist

This appendix covers some important items you should at least think about when considering how best to secure your container deployments. In your environment it might well not make sense to apply *every* item, but if you have thought about them, you will be off to a good start. No doubt this list is not absolutely comprehensive!


Build machines
Container images
Container config
Host machines
Build
Deploy
Run
Secrets
Runtime & network security

@lizrice

Choose your own adventure

1. Are your builds running separately from your production cluster?

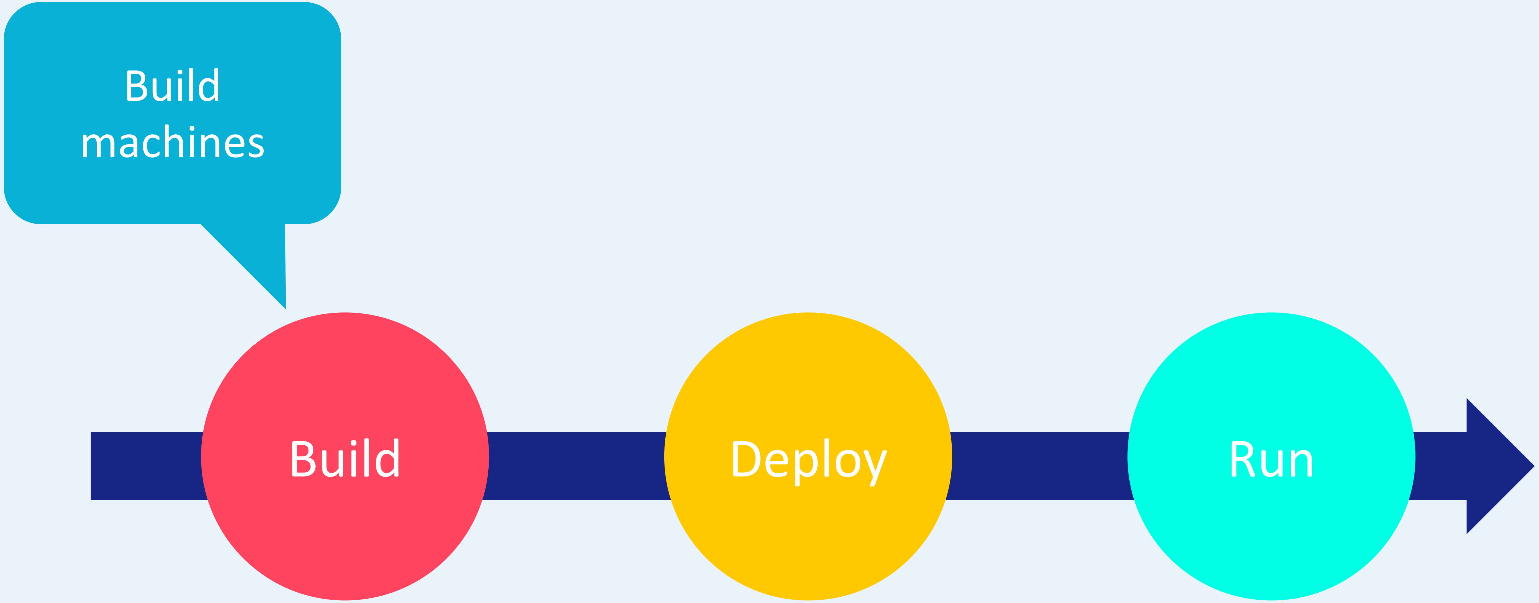
2. Is all executable code added to a container image at build time?

3. Are you avoiding –privileged?

4. Are you keeping hosts up to date with the latest security releases?

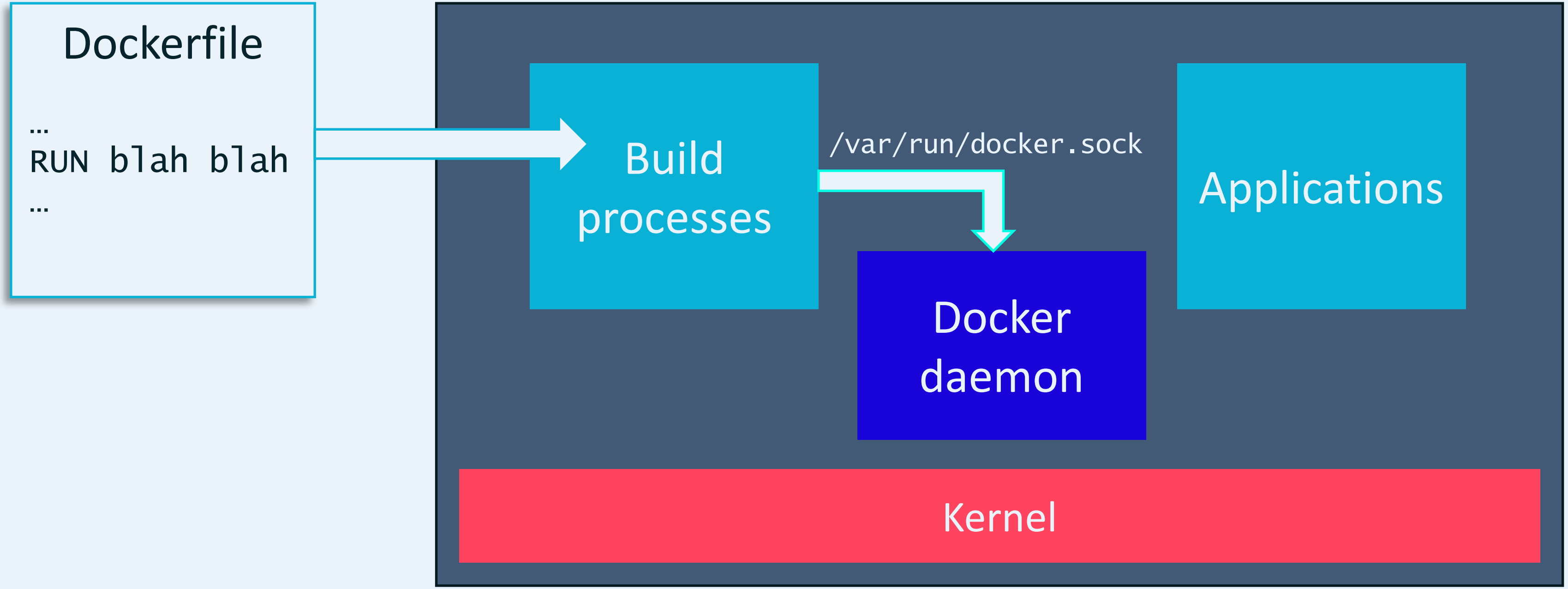
5. Are your secrets encrypted at rest and in transit?

6. Can you prevent container drift? *Not open source…*



1. Are your builds running separately from your production cluster?

Don’t run builds in your production cluster



Dockerfile

```
…
RUN blah blah
…
```

Build processes

`/var/run/docker.sock`

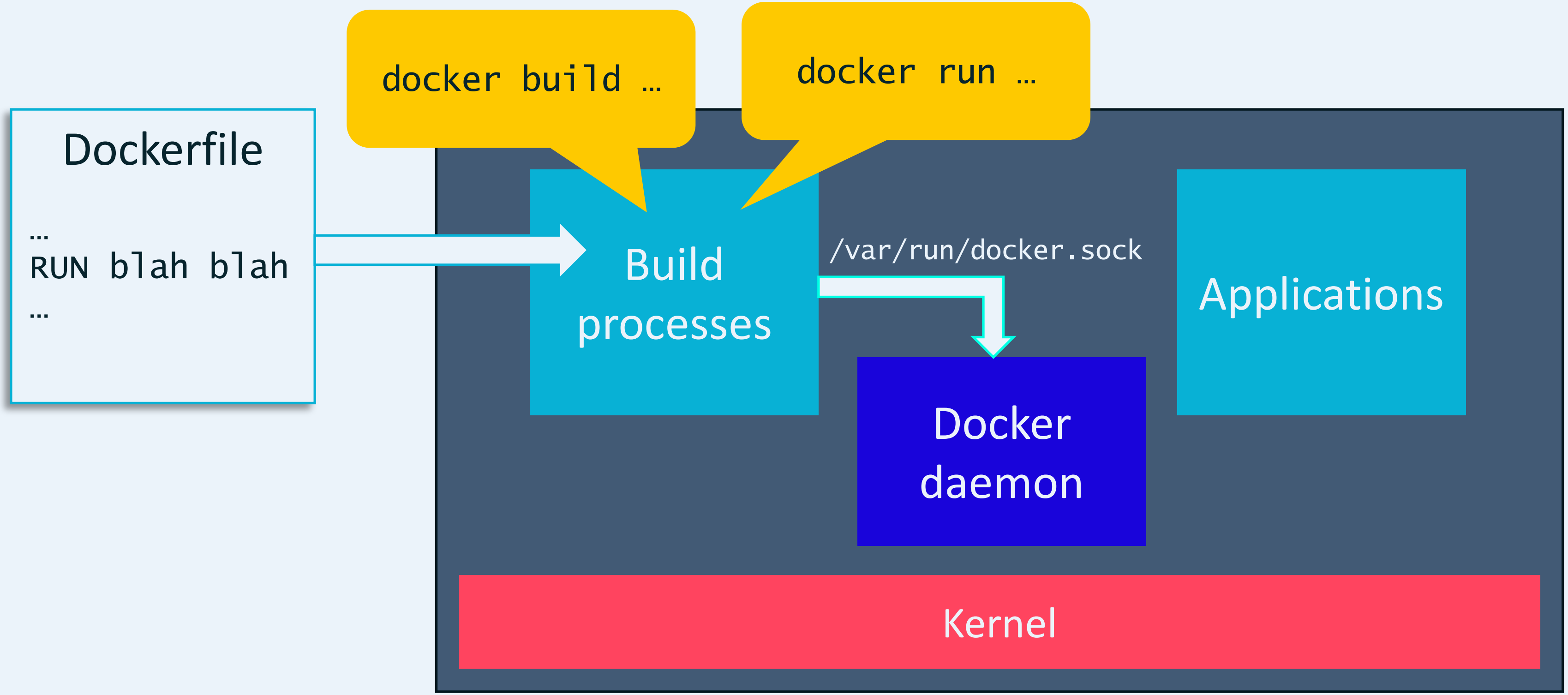
Docker daemon

Applications

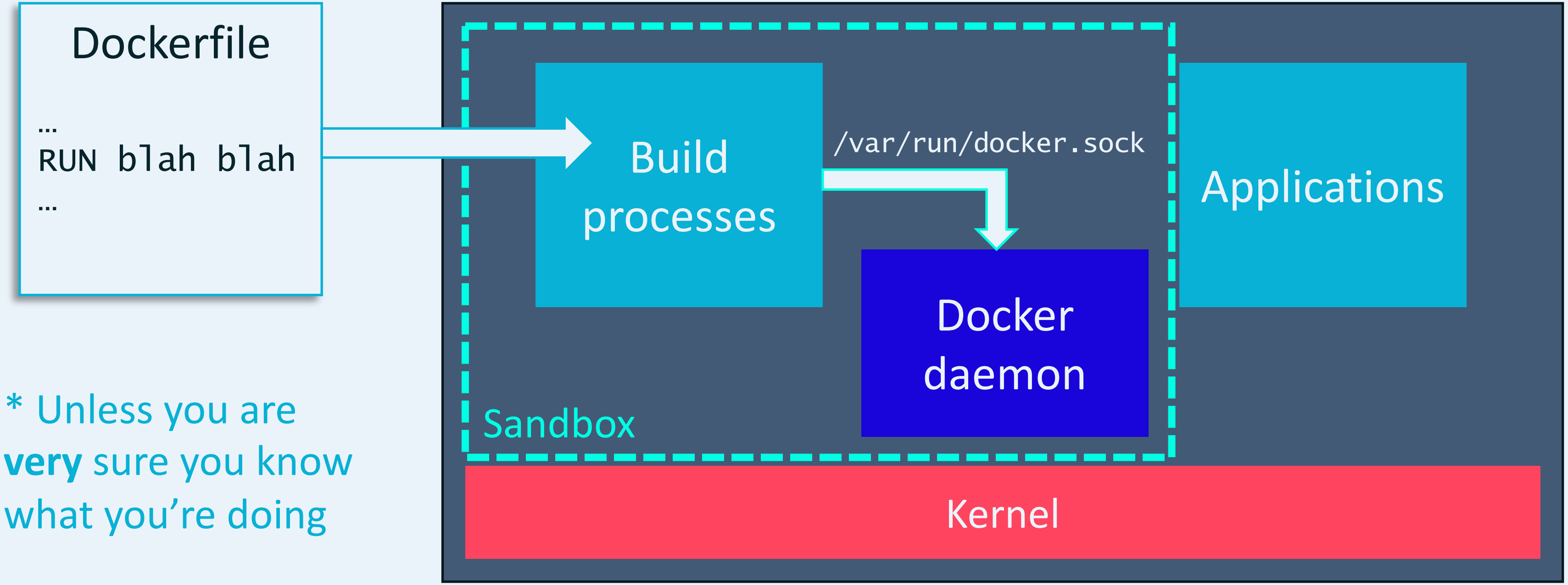
Kernel

Demo – dangerous dockerfiles

Don’t run builds in your production cluster

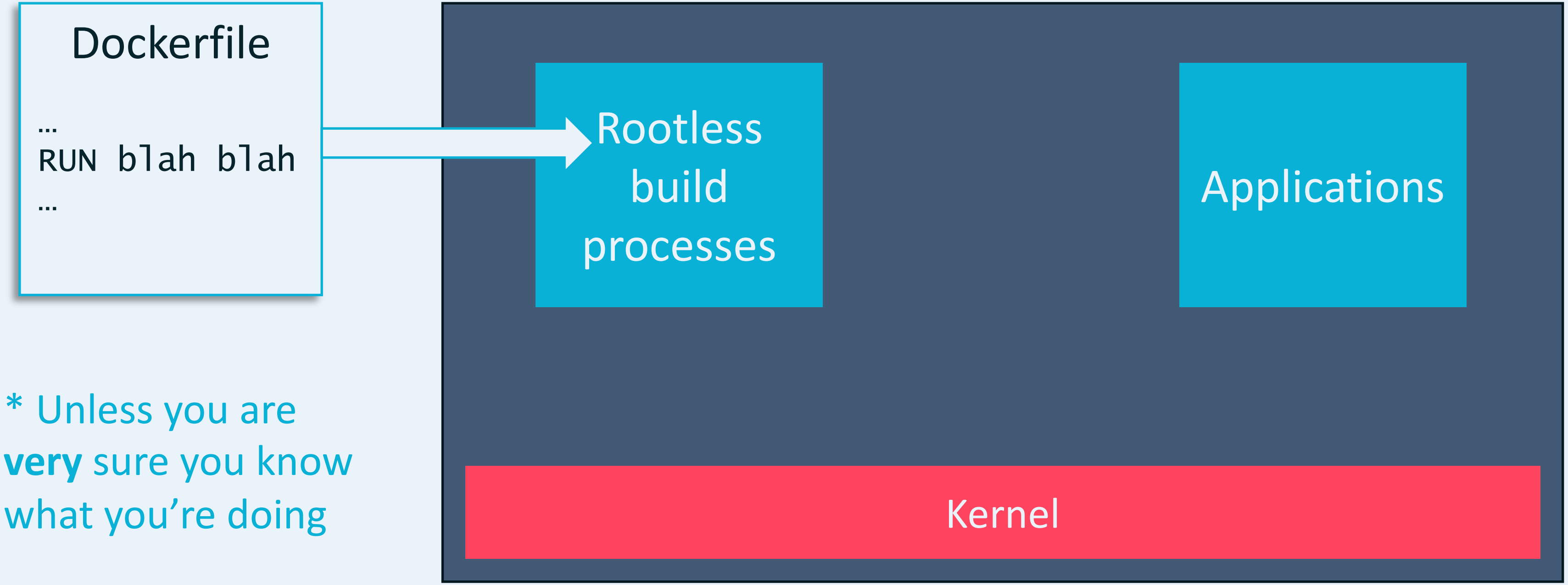


Don’t run builds in your production cluster



* Unless you are **very** sure you know what you’re doing

Don’t run builds in your production cluster*



Dockerfile

```
…
RUN blah blah
…
```

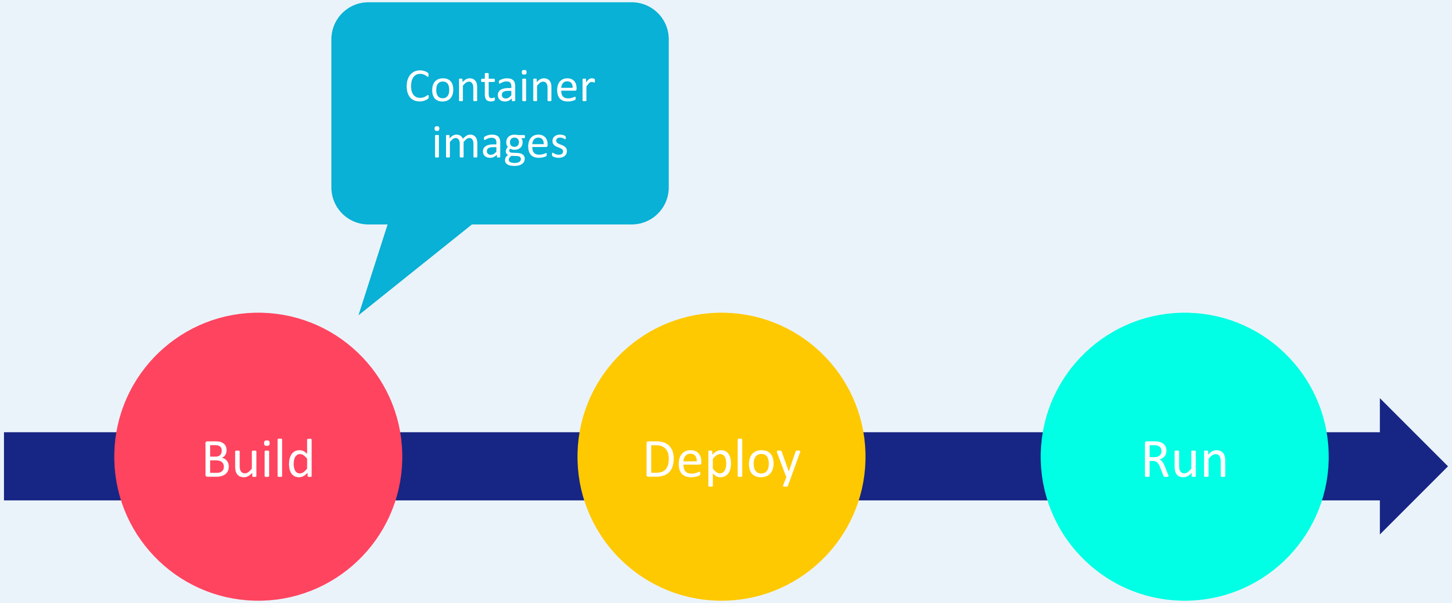
Rootless build processes

Applications

Kernel

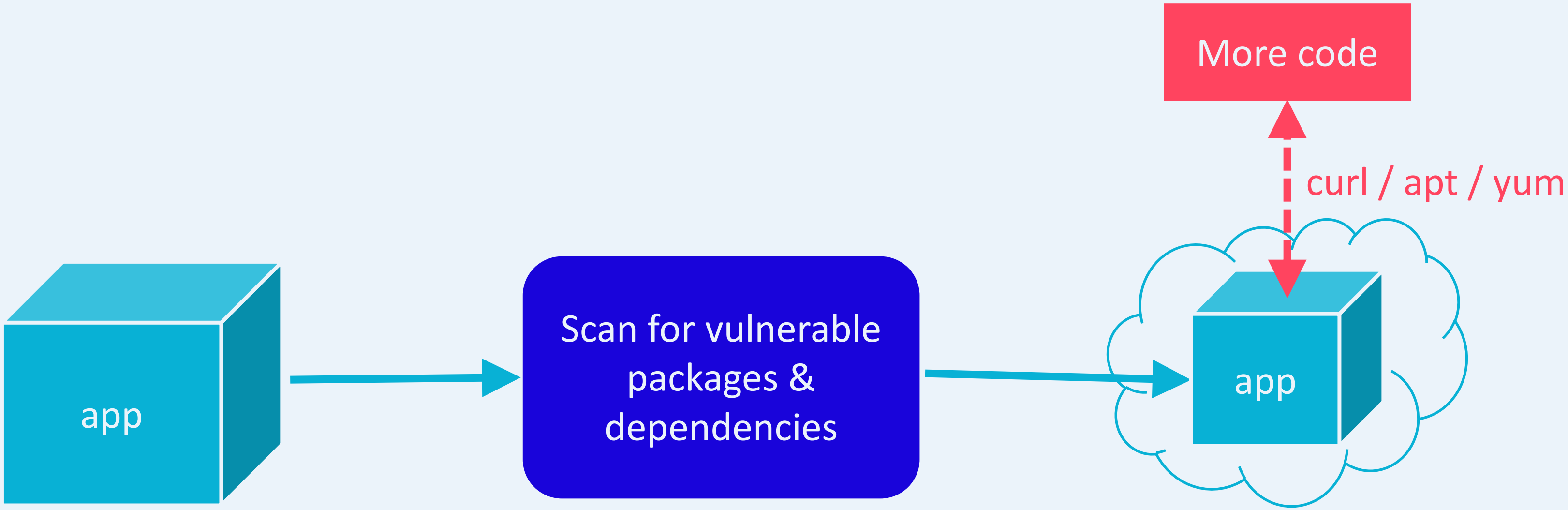
* Unless you are **very** sure you know what you’re doing

1. Are your builds running separately from your production cluster?
2. Is all executable code added to a container image at build time?
3. Are you avoiding –privileged?
4. Are you keeping hosts up to date with the latest security releases?
5. Are your secrets encrypted at rest and in transit?
6. Can you prevent container drift? *Not open source…*



2. Is all executable code added to a container image at build time?

Treat containers as immutable



Demo – image drift

1. Are your builds running separately from your production cluster?

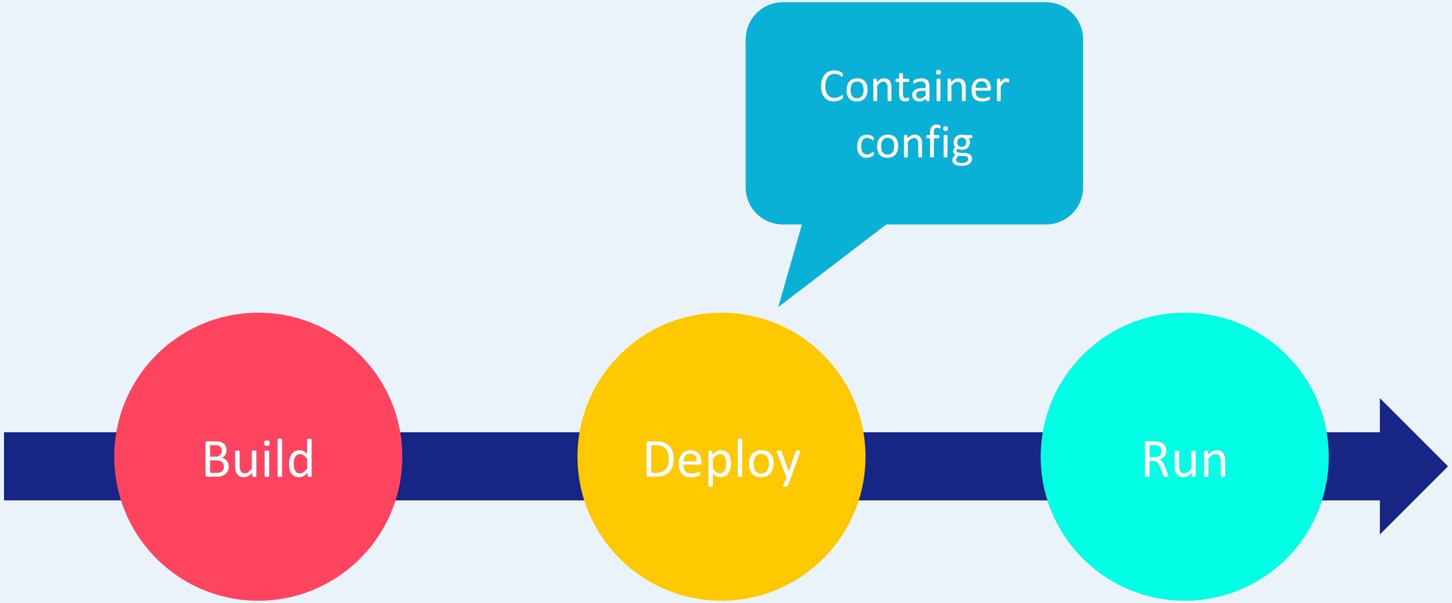
2. Is all executable code added to a container image at build time?

3. Are you avoiding –privileged?

4. Are you keeping hosts up to date with the latest security releases?

5. Are your secrets encrypted at rest and in transit?

6. Can you prevent container drift? *Not open source…*



3. Are you avoiding –privileged?

`--privileged`

“The most dangerous flag in computing”

- Andrew Martin

Linux capabilities

More granular permissions than root

Most containers don't need to:

- Install Kernel modules (CAP_SYS_MODULE)
- Change the system time (CAP_SYS_TIME)
- Trace / modify arbitrary processes (CAP_SYS_PTRACE)

Demo – privileged container

```
$ docker run --rm -it --cap-add=ALL ubuntu
/# more /proc/1/status | grep CapEff
CapEff:    0000003fffffffff

$ docker run --rm -it --cap-drop=ALL ubuntu
/# more /proc/1/status | grep CapEff
CapEff:    0000000000000000

$ docker run --rm -it ubuntu
/# more /proc/1/status | grep CapEff
CapEff:    00000000a80425fb

$ docker run --rm -it --privileged ubuntu
/# more /proc/1/status | grep CapEff
CapEff:    0000003fffffffff
```

All capabilities

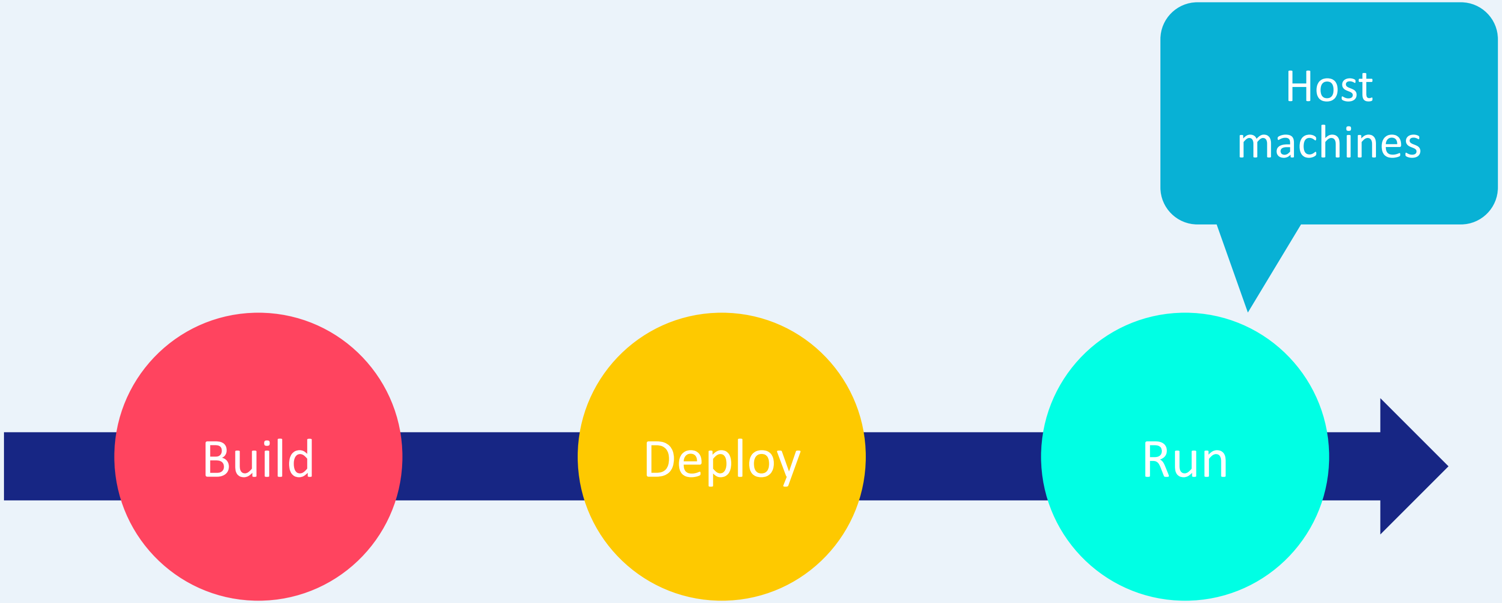
```
$ docker run --rm -it ubuntu
root@316a2ab0ddcb:/# ls /dev
console  core  fd  full  mqueue  null  ptmx  pts  random  shm  stderr  stdin  stdout  tty  urandom
zero

$ docker run --rm -it --privileged ubuntu
root@87c19bbc393a:/# ls /dev
autofs           loop-control        ptmx      tty14  tty33  tty52   ttyS13  ttyS4      vcsa
bsg              loop0               pts       tty15  tty34  tty53   ttyS14  ttyS5      vcsa1
btrfs-control    loop1               random    tty16  tty35  tty54   ttyS15  ttyS6      vcsa2
console          loop2               rfkill    tty17  tty36  tty55   ttyS16  ttyS7      vcsa3
core             loop3               rtc0      tty18  tty37  tty56   ttyS17  ttyS8      vcsa4
cpu_dma_latency  loop4               sda       tty19  tty38  tty57   ttyS18  ttyS9      vcsa5
cuse             loop5               sda1      tty2   tty39  tty58   ttyS19  ttyprintk  vcsa6
dm-0             loop6               sg0       tty20  tty4   tty59   ttyS2   udmabuf    vcsu
dm-1             loop7               shm       tty21  tty40  tty6    ttyS20  uhid       vcsu1
dri              mapper              snapshot  tty22  tty41  tty60
ecryptfs         mcelog              snd       tty23  tty42  tty6
fb0              mem                 stderr    tty24  tty43  tty6
fd               memory_bandwidth    stdin     tty25  tty44  tty6
…
```

All host devices

You don't need `--privileged` to be root

1. Are your builds running separately from your production cluster?
2. Is all executable code added to a container image at build time?
3. Are you avoiding –privileged?
4. Are you keeping hosts up to date with the latest security releases?
5. Are your secrets encrypted at rest and in transit?
6. Can you prevent container drift? *Not open source…*



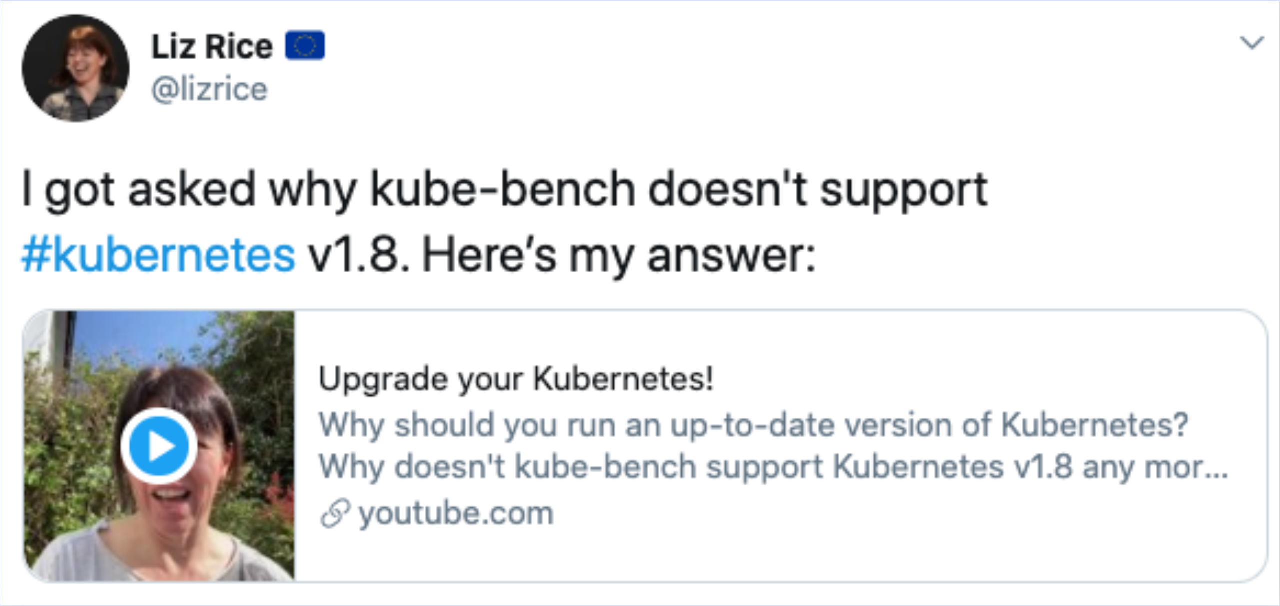
Host machines

Build

Deploy

Run

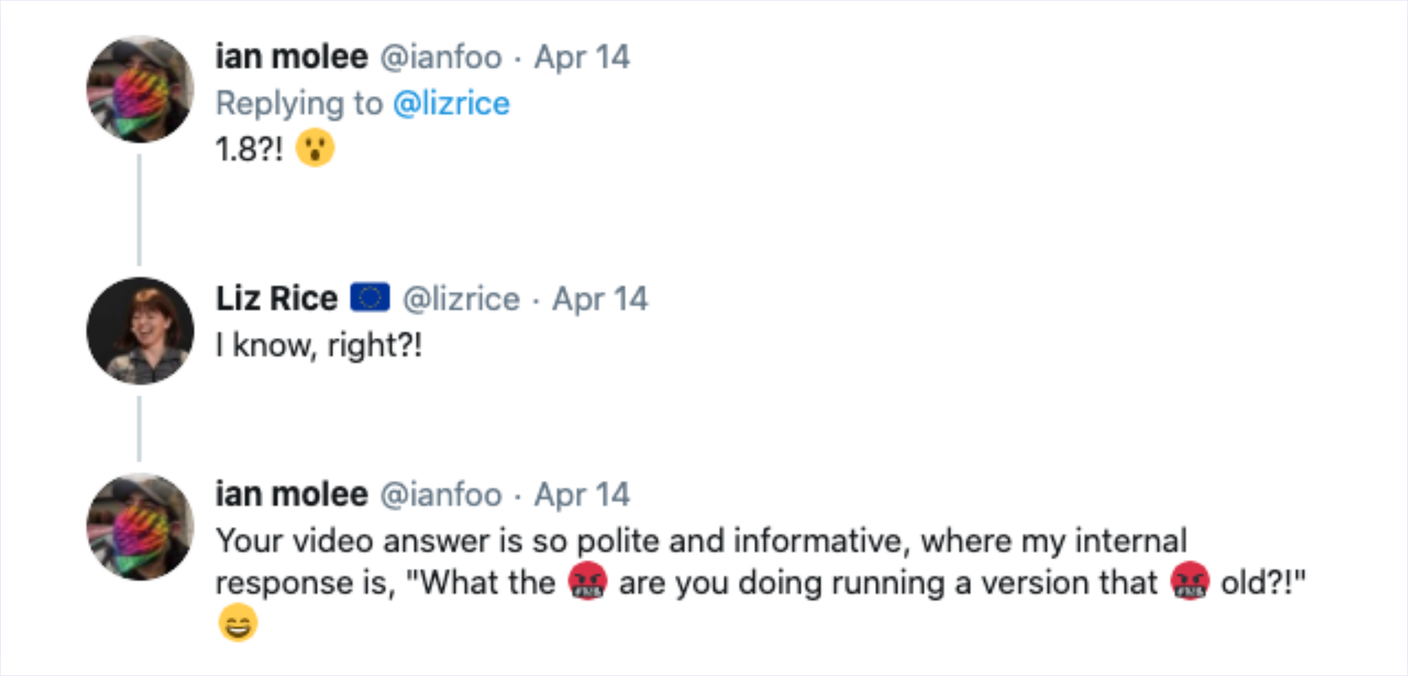
4. Are you keeping hosts up to date with the latest security releases?



Liz Rice 🇪🇺
@lizrice

I got asked why kube-bench doesn't support #kubernetes v1.8. Here’s my answer:

Upgrade your Kubernetes!
Why should you run an up-to-date version of Kubernetes? Why doesn't kube-bench support Kubernetes v1.8 any mor...
🔗 youtube.com

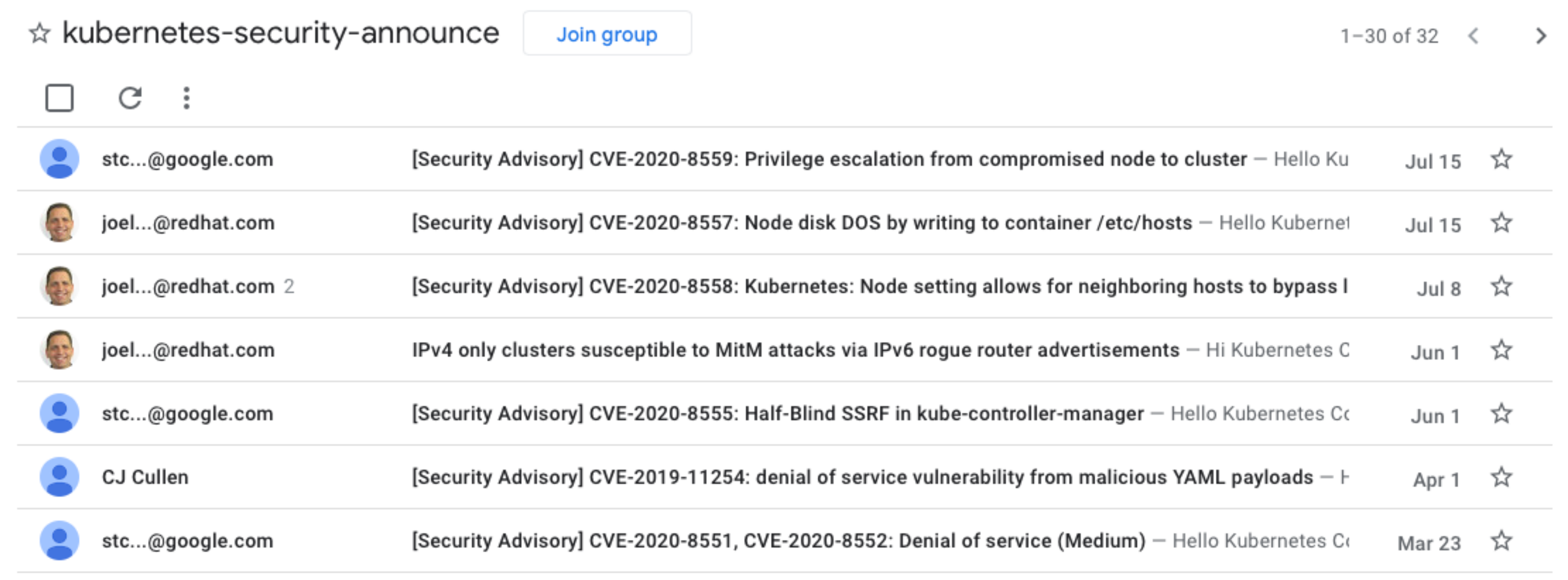


ian molee @ianfoo · Apr 14
Replying to @lizrice
1.8?! 😮

Liz Rice 🇪🇺 @lizrice · Apr 14
I know, right?!

ian molee @ianfoo · Apr 14
Your video answer is so polite and informative, where my internal response is, "What the 🤬 are you doing running a version that 🤬 old?!" 😄

Kubernetes security announcements



☆ kubernetes-security-announce

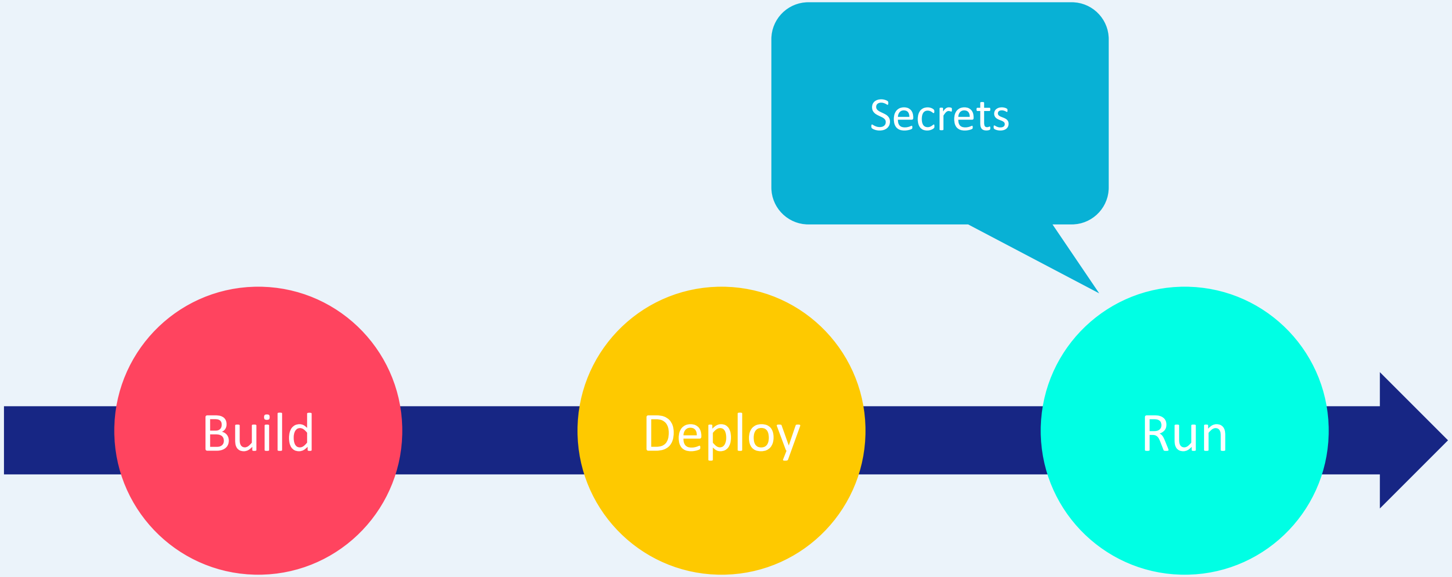
☐

stc...@google.com	[Security Advisory] CVE-2020-8559: Privilege escalation from compromised node to cluster — Hello Ku	Jul 15
joel...@redhat.com	[Security Advisory] CVE-2020-8557: Node disk DOS by writing to container /etc/hosts — Hello Kubernet	Jul 15
joel...@redhat.com 2	[Security Advisory] CVE-2020-8558: Kubernetes: Node setting allows for neighboring hosts to bypass l	Jul 8
joel...@redhat.com	IPv4 only clusters susceptible to MitM attacks via IPv6 rogue router advertisements — Hi Kubernetes C	Jun 1
stc...@google.com	[Security Advisory] CVE-2020-8555: Half-Blind SSRF in kube-controller-manager — Hello Kubernetes Co	Jun 1
CJ Cullen	[Security Advisory] CVE-2019-11254: denial of service vulnerability from malicious YAML payloads — H	Apr 1
stc...@google.com	[Security Advisory] CVE-2020-8551, CVE-2020-8552: Denial of service (Medium) — Hello Kubernetes Co	Mar 23

@lizrice

Host scans and updates

1. Are your builds running separately from your production cluster?
2. Is all executable code added to a container image at build time?
3. Are you avoiding –privileged?
4. Are you keeping hosts up to date with the latest security releases?
5. Are your secrets encrypted at rest and in transit?
6. Can you prevent container drift? *Not open source…*



5. Are your secrets encrypted at rest and in transit?

Demo – secret encryption

```
$ kubectl get secret my-secret -o jsonpath="{.data.password}" |
base64 --decode
Keepthissecret

$ sudo grep keepthissecret /var/lib/etcd/member/snap/db
Binary file /var/lib/etcd/member/snap/db matches
```

Encrypting secrets

Use EncryptionConfiguration

Secrets injection: Hashicorp Vault, CyberArk Conjur, Aqua etc...

```
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
  - resources:
    - secrets
    providers:
    - aescbc:
        keys:
        - name: key1
          secret: <BASE 64 ENCODED SECRET>
    - identity: {}
```

1. Are your builds running separately from your production cluster?

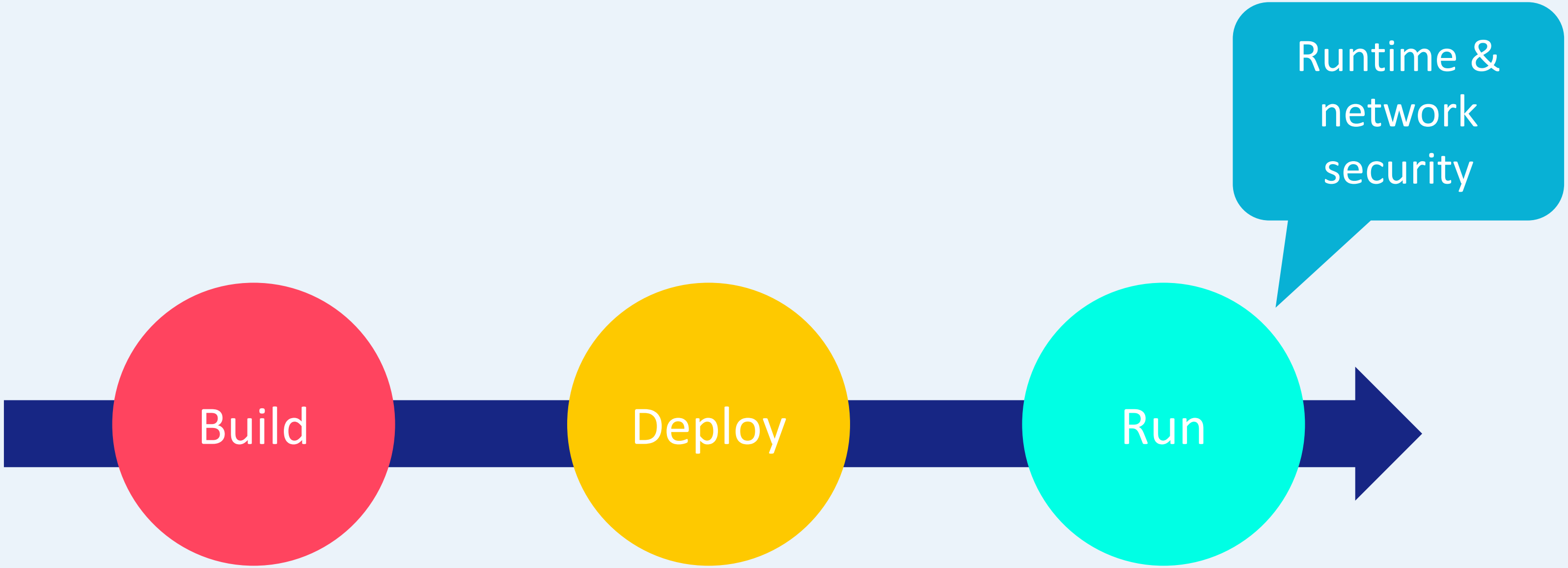
2. Is all executable code added to a container image at build time?

3. Are you avoiding –privileged?

4. Are you keeping hosts up to date with the latest security releases?

5. Are your secrets encrypted at rest and in transit?

6. Can you prevent container drift? *Not open source…*



6. Can you prevent container drift? *Not open source…*

Demo – drift prevention

1. Are your builds running separately from your production cluster?

2. Is all executable code added to a container image at build time?

3. Are you avoiding –privileged?

4. Are you keeping hosts up to date with the latest security releases?

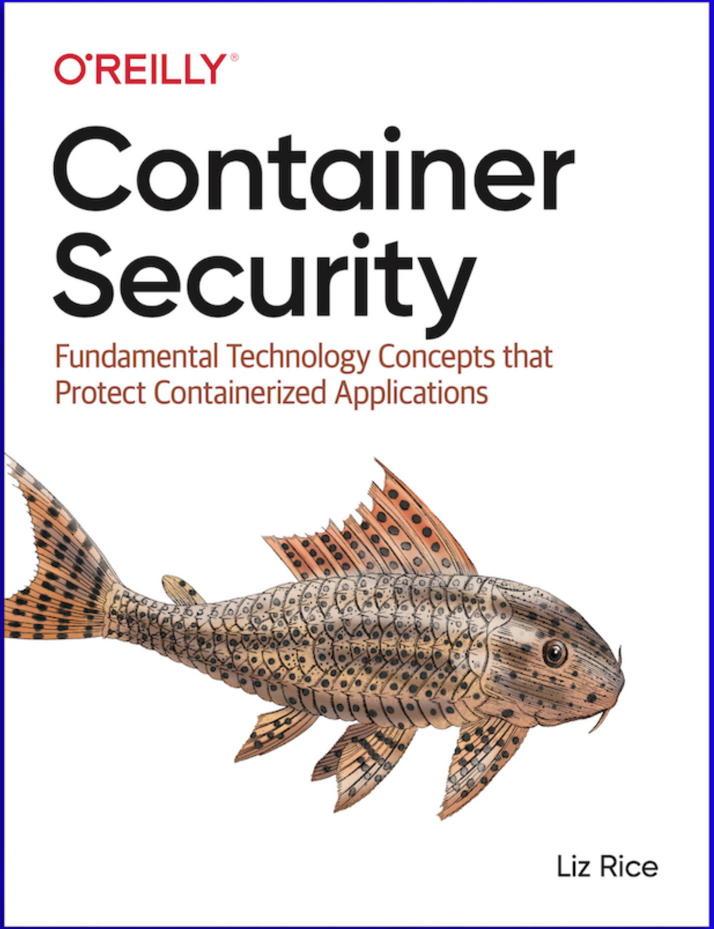
5. Are your secrets encrypted at rest and in transit?

6. Can you prevent container drift? *Not open source…*

aqua

containersecurity.tech

@lizrice | @aquasecteam



O'REILLY

Container Security

Fundamental Technology Concepts that Protect Containerized Applications

Liz Rice