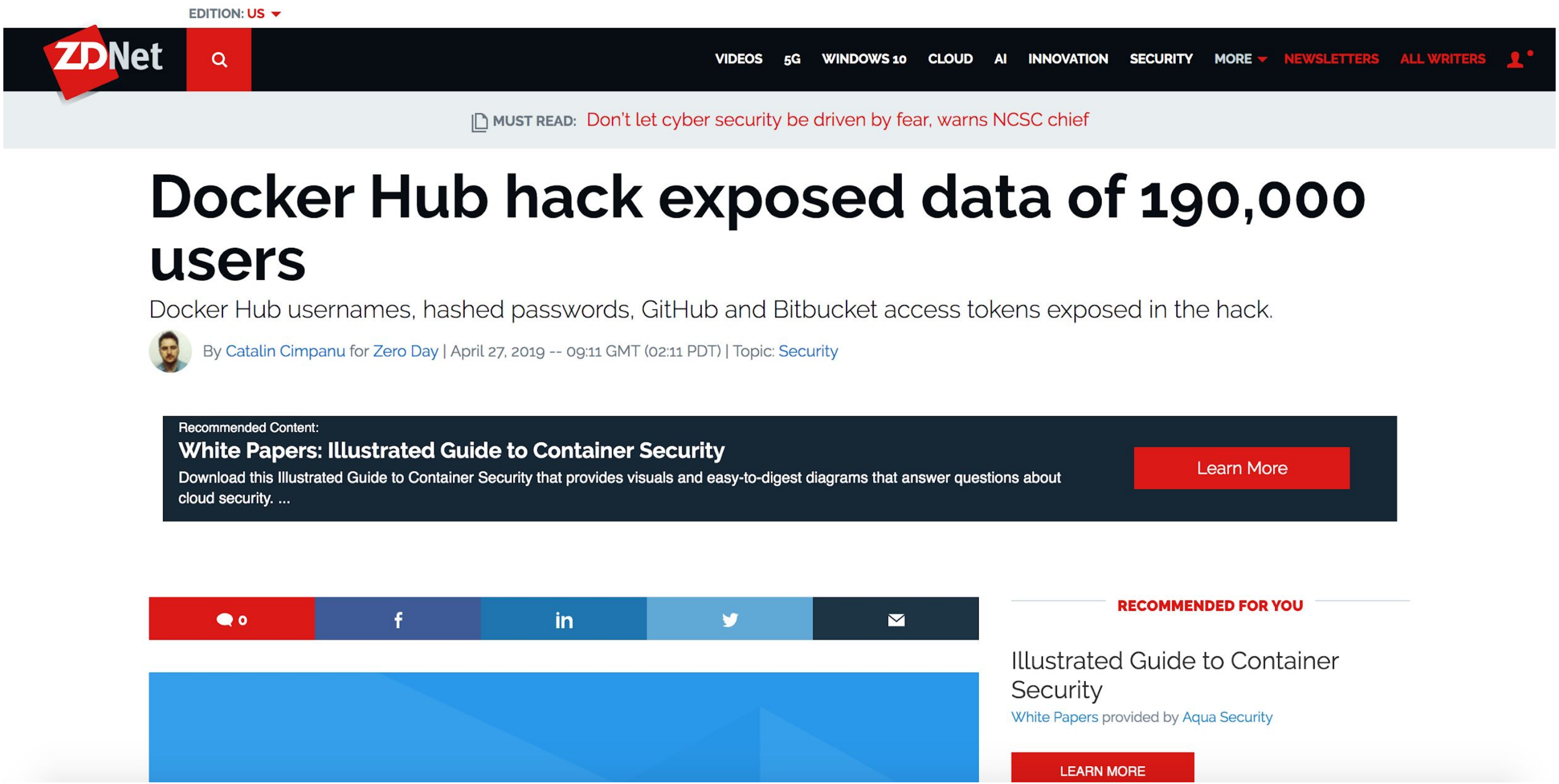
Protecting Sensitive Code with Encrypted Container Images on Kubernetes

Harshal Patil, Brandon Lum
IBM



MUST READ: Don't let cyber security be driven by fear, warns NCSC chief

Docker Hub hack exposed data of 190,000 users

Docker Hub usernames, hashed passwords, GitHub and Bitbucket access tokens exposed in the hack.

By Catalin Cimpanu for Zero Day | April 27, 2019 -- 09:11 GMT (02:11 PDT) | Topic: Security

What does this mean for your images?

Notary + PORTIERIS

Image Signing with Notary & Portieris will ensure the integrity of your deployment images!

But…

Private Images’ sensitive content will be exposed!

Container Image Encryption

Build

- Build as normal
- **Encrypt**
- Push

Encrypt

- Encrypted image stored
- Cannot be read

Run

- Pull
- **Decrypt**
- Run

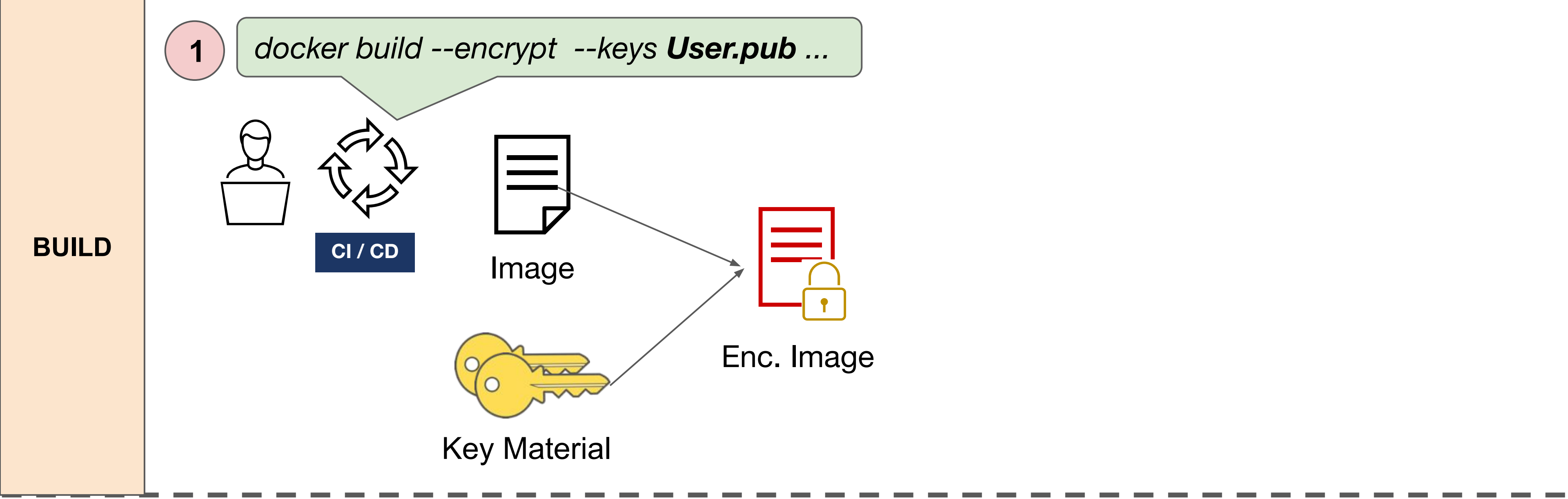
Benefits?

- Image Confidentiality & Deprivileged Registry

- Execution Boundary Control

  "If my code is running, I know it's in my cluster"

  - Encrypted Containers Images + Key management could provide guarantees about where an image can run.
  - i.e. Image X can only run in the EU nodes.



BUILD

1

```
docker build --encrypt  --keys User.pub ...
```

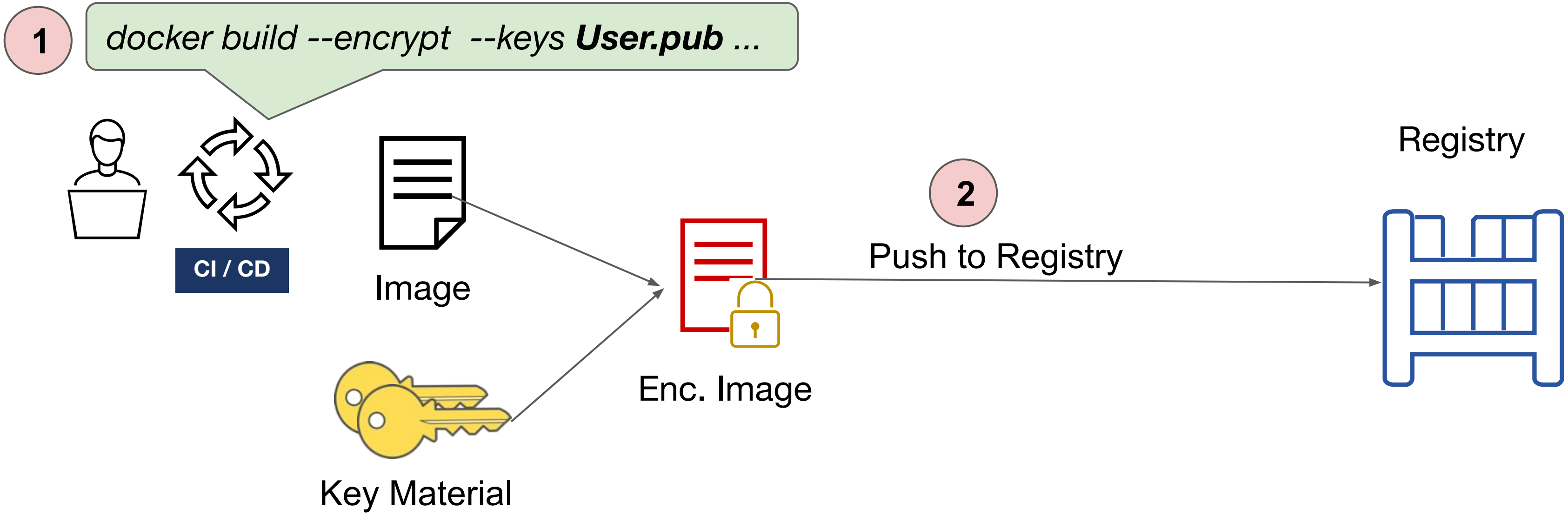
CI / CD

Image

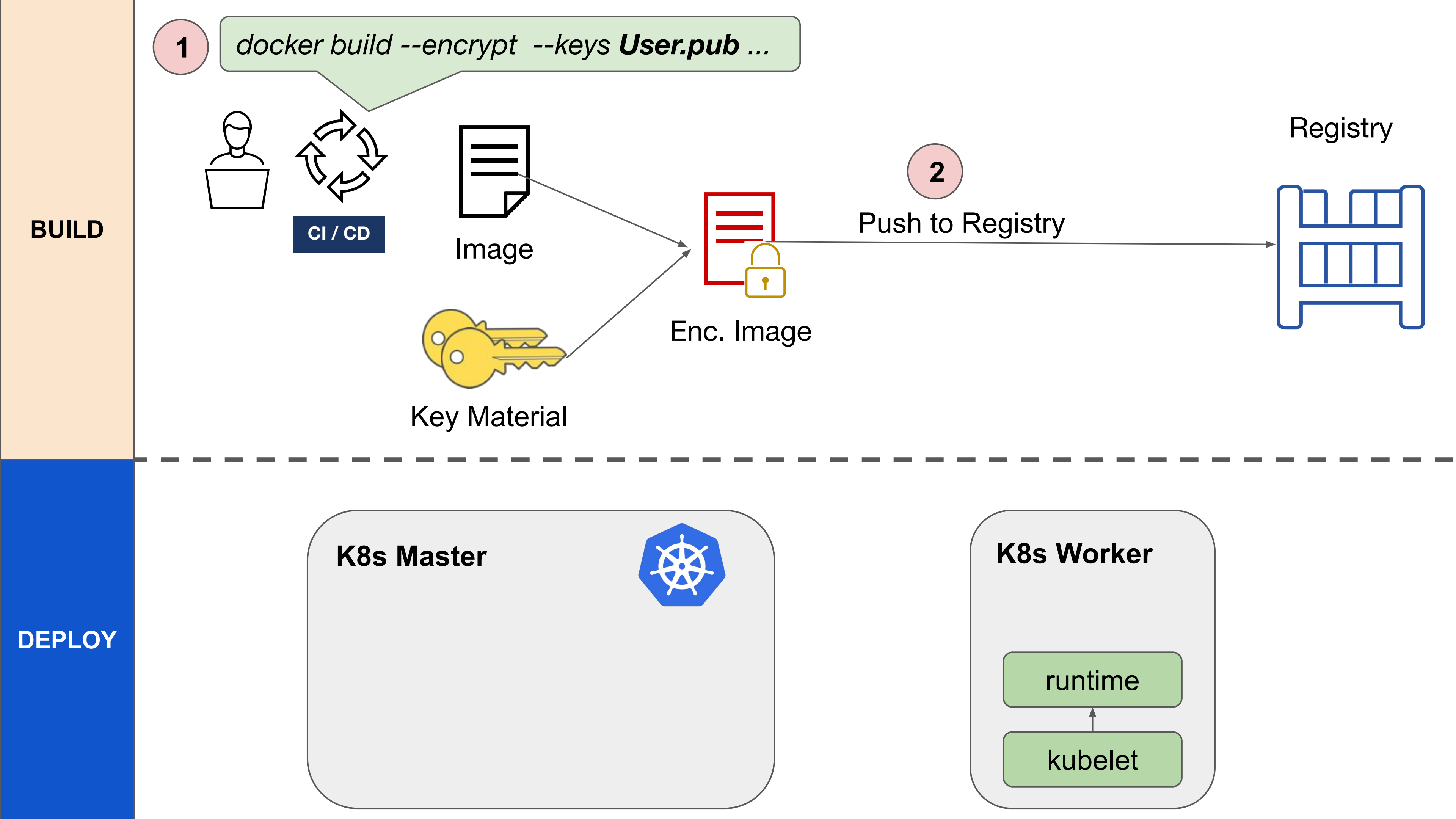
Key Material

Enc. Image

BUILD



```
docker build --encrypt  --keys User.pub ...
```



BUILD

1

```
docker build --encrypt  --keys User.pub ...
```

CI / CD

Image

Key Material

Enc. Image

2

Push to Registry

Registry

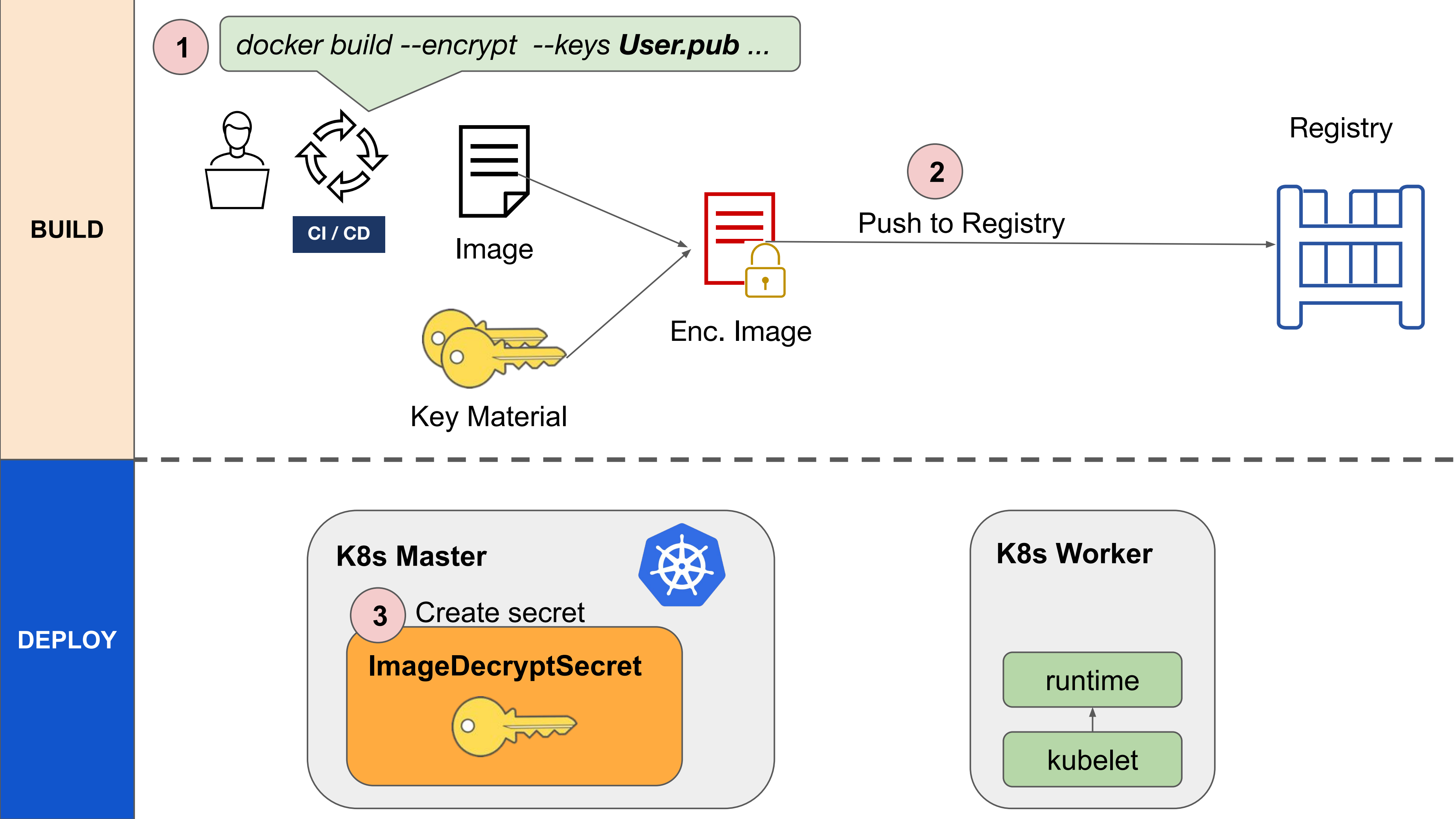
DEPLOY

K8s Master

K8s Worker

runtime

kubelet



BUILD

1

```
docker build --encrypt  --keys User.pub ...
```

CI / CD

Image

Key Material

Enc. Image

2

Push to Registry

Registry

DEPLOY

K8s Master

3 Create secret

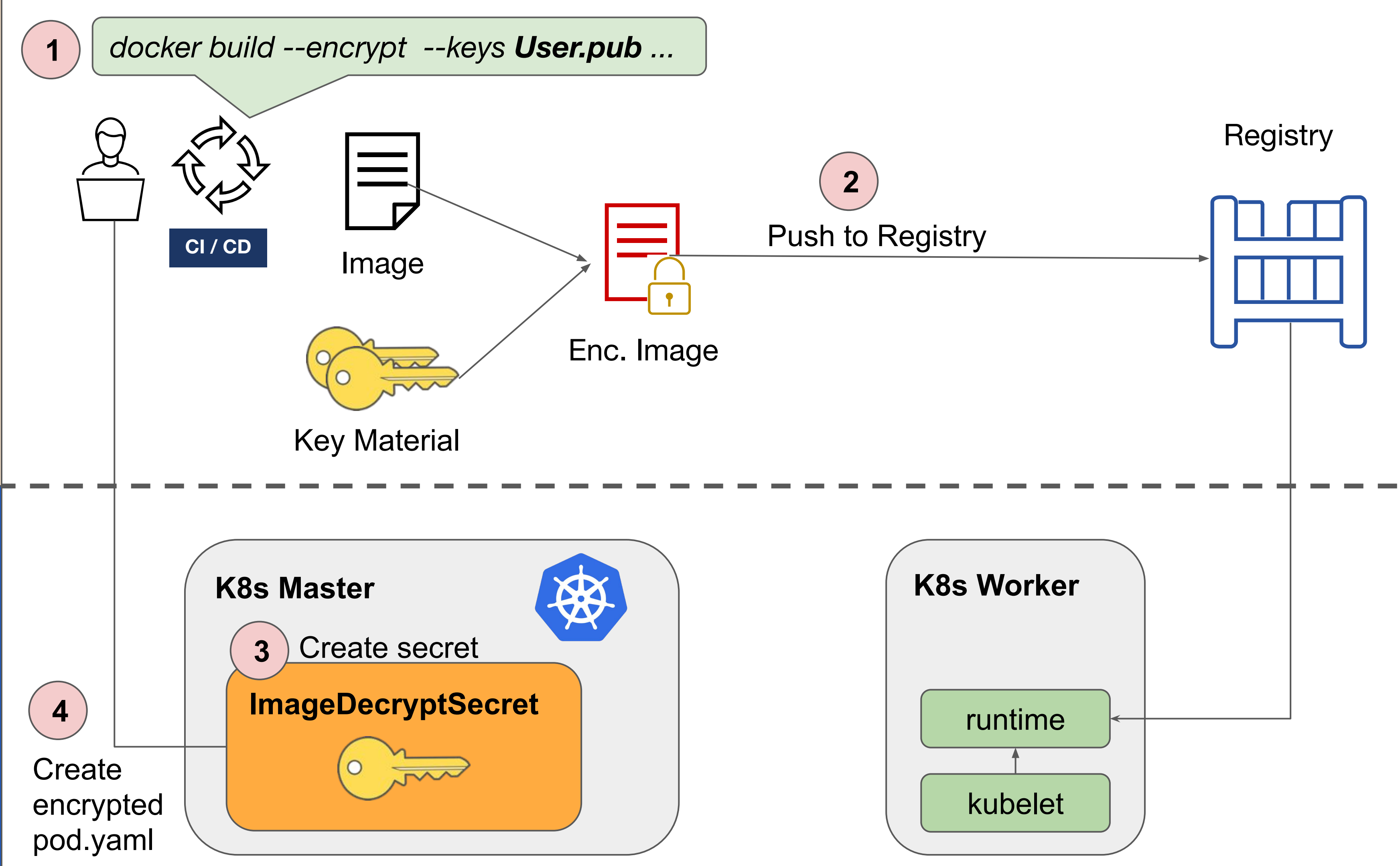
ImageDecryptSecret

K8s Worker

runtime

kubelet

BUILD



1

```
docker build --encrypt  --keys User.pub ...
```

CI / CD

Image

Key Material

Enc. Image

2 Push to Registry

Registry

DEPLOY

K8s Master

3 Create secret

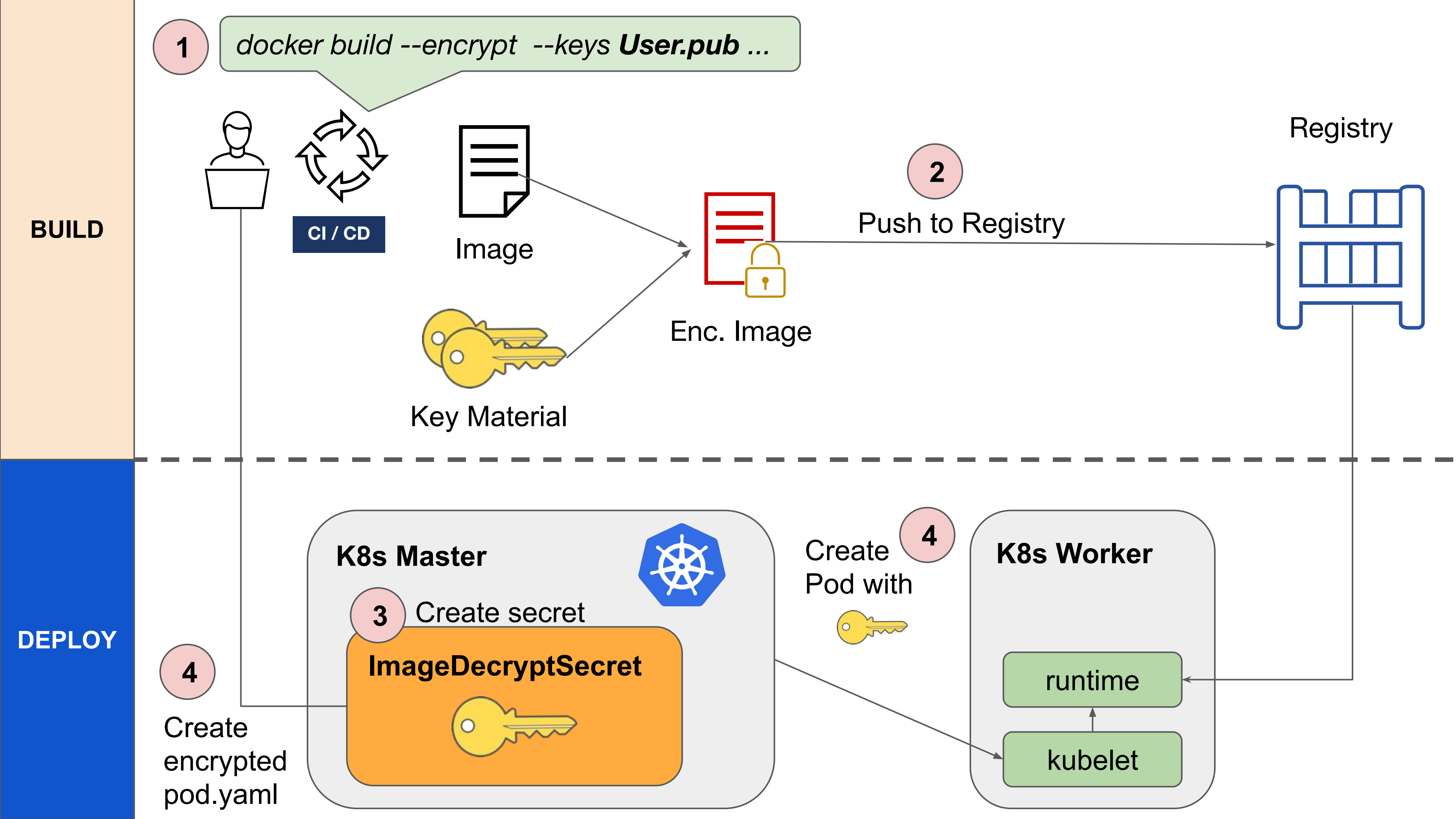
ImageDecryptSecret

4 Create encrypted pod.yaml

K8s Worker

runtime

kubelet



BUILD

1

```
docker build --encrypt  --keys User.pub ...
```

CI / CD

Image

Key Material

Enc. Image

2

Push to Registry

Registry

DEPLOY

K8s Master

3 Create secret

ImageDecryptSecret

4

Create encrypted pod.yaml

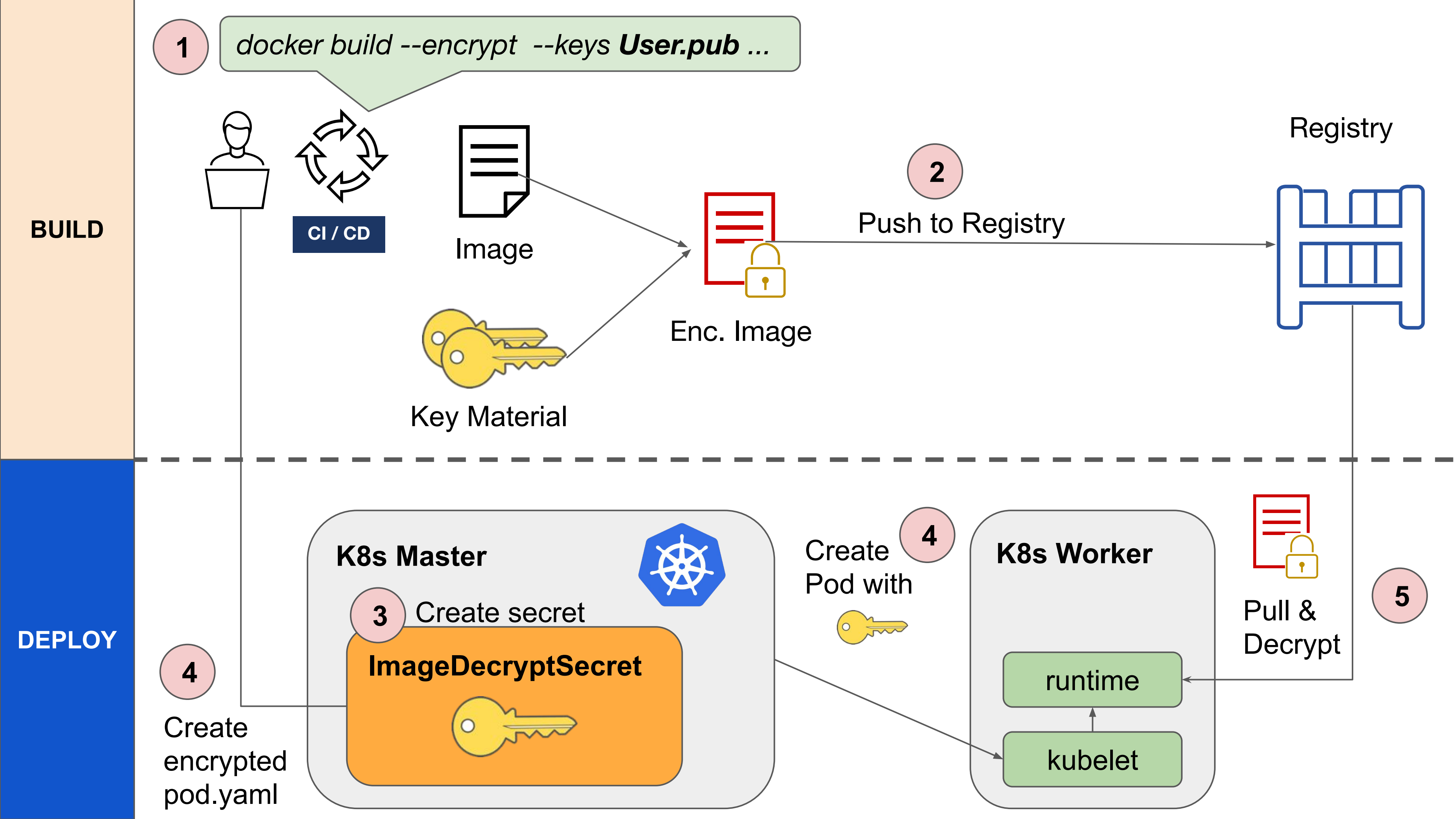
4

Create Pod with

K8s Worker

runtime

kubelet



BUILD

1

```
docker build --encrypt  --keys User.pub ...
```

CI / CD

Image

Key Material

Enc. Image

2

Push to Registry

Registry

DEPLOY

K8s Master

3 Create secret

ImageDecryptSecret

4

Create
encrypted
pod.yaml

Create
Pod with

4

K8s Worker

runtime

kubelet

Pull &
Decrypt

5

DEMO

Demo Link/Materials

KEP https://github.com/kubernetes/enhancements/issues/1067

OCI SPEC Issue
https://github.com/opencontainers/image-spec/issues/747

OCI SPEC PR
https://github.com/opencontainers/image-spec/pull/775

POC and Demo
https://github.com/harche/kubernetes/commit/8a0ef9898a55110d7e41f8e1a846c66cfc2fa691

https://youtu.be/S3FK4y5McOk

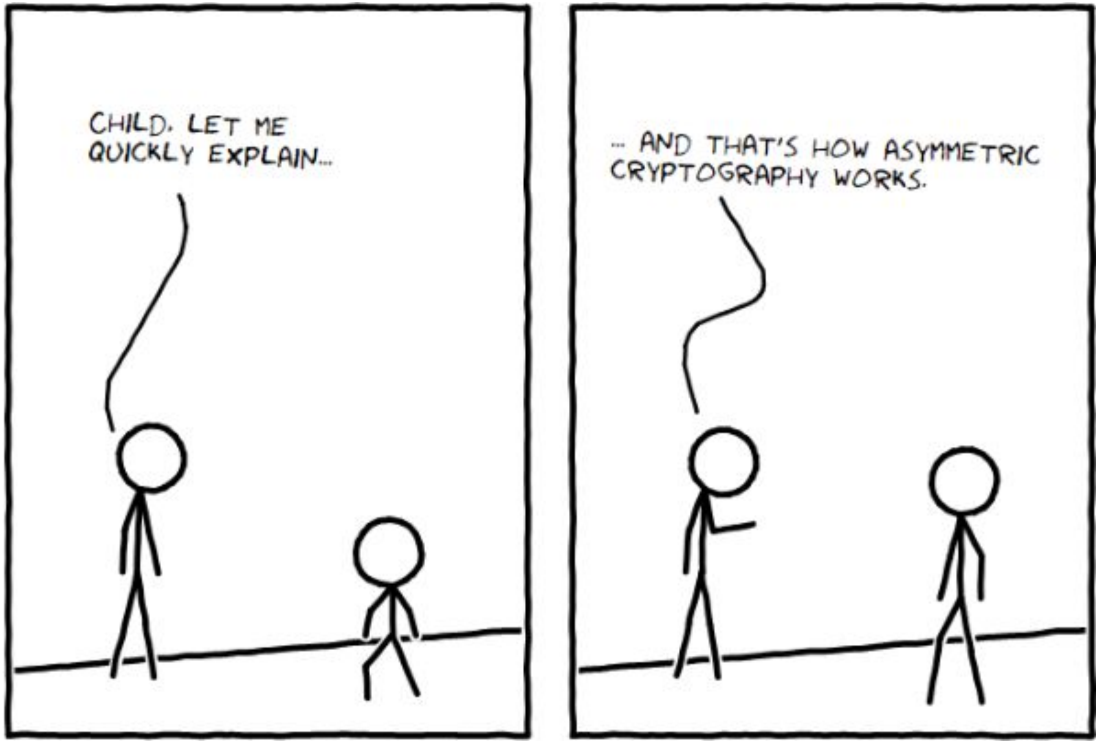
Conventional Knowledge

Encryption → Bad deduplication

Claim:
With container images, this doesn’t have to be the case!

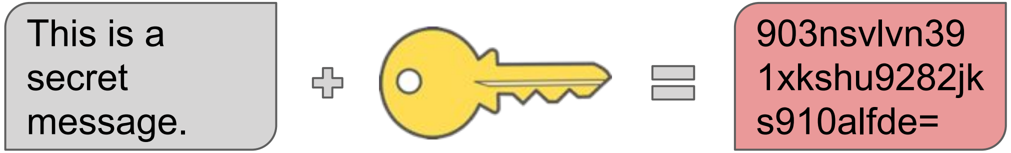
Deep Dive

Encryption Primer

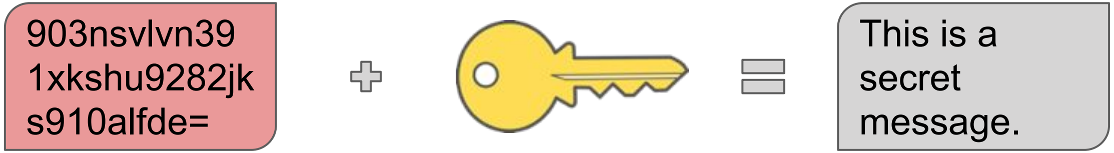

CHILD. LET ME QUICKLY EXPLAIN...
... AND THAT'S HOW ASYMMETRIC CRYPTOGRAPHY WORKS.

https://www.outsystems.com/blog/posts/how-to-teach-child-about-asymmetric-cryptography/

Encryption Primer - Symm Enc.



This is a secret message. + = 903nsvlvn39 1xkshu9282jk s910alfde=

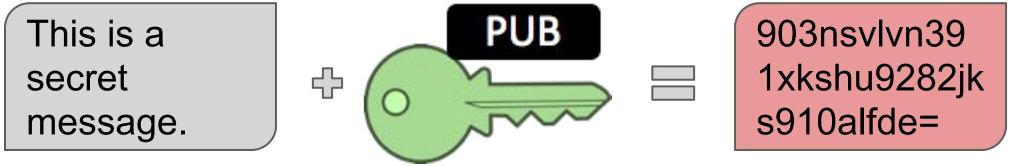


903nsvlvn39 1xkshu9282jk s910alfde= + = This is a secret message.

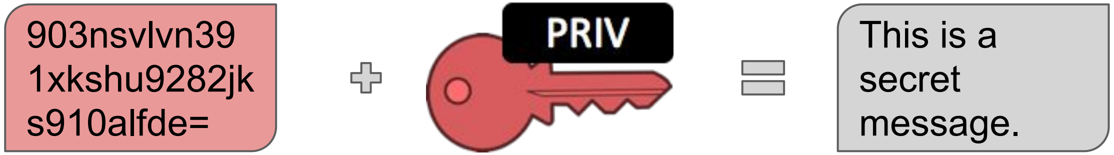
👍 Fast, good for large data

👎 Key needs to be securely shared somehow

Encryption Primer - Assym Enc.



This is a secret message. + PUB = 903nsvlvn39 1xkshu9282jk s910alfde=

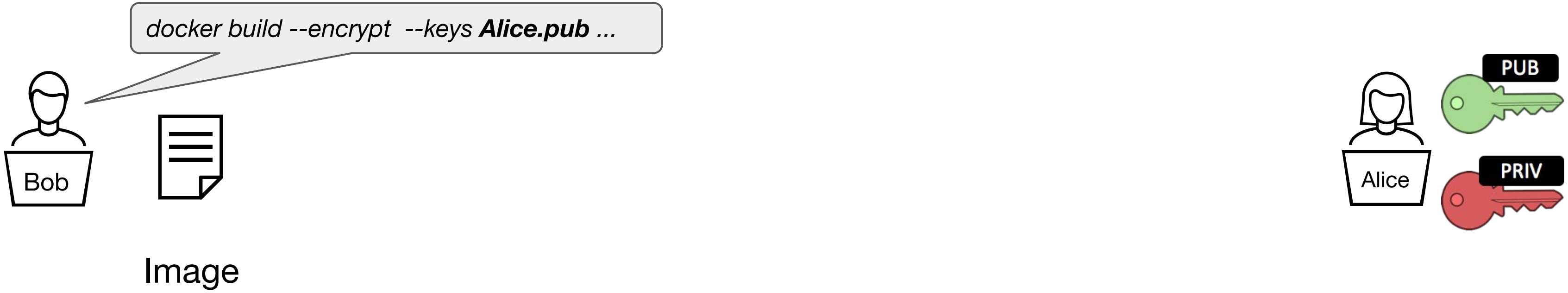


903nsvlvn39 1xkshu9282jk s910alfde= + PRIV = This is a secret message.

👍 Addresses key sharing: Each user has a Public-Private key pair, where Public Key is not secret, can be published.

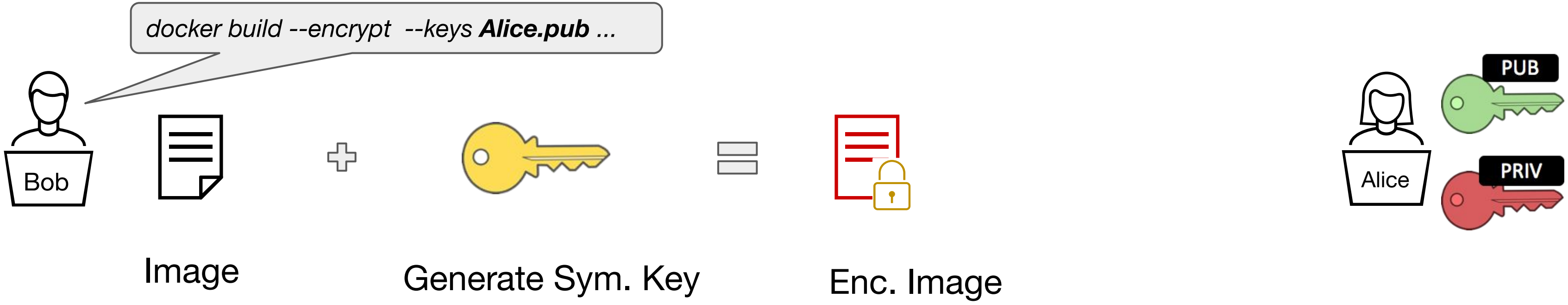
👎 Slow

Encrypt and distribution flow



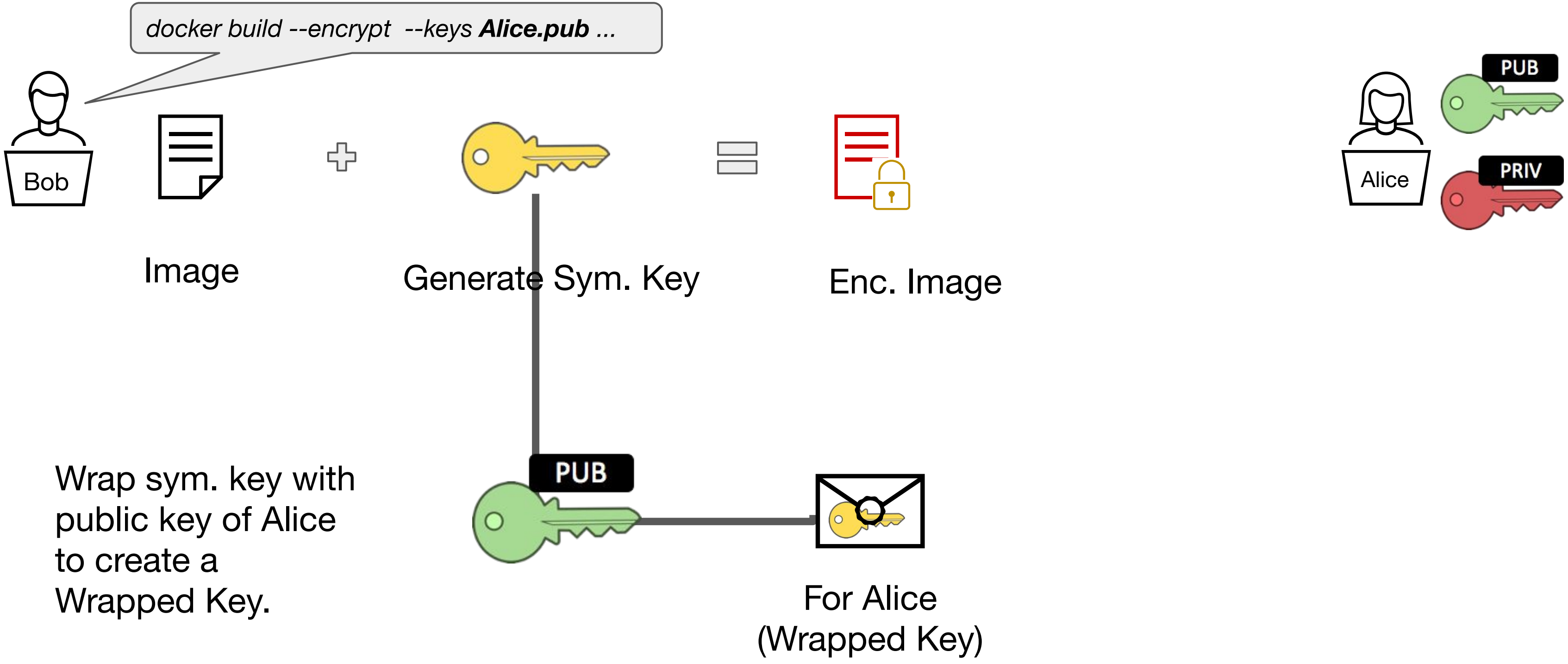
```
docker build --encrypt  --keys Alice.pub ...
```

Encrypt and distribution flow



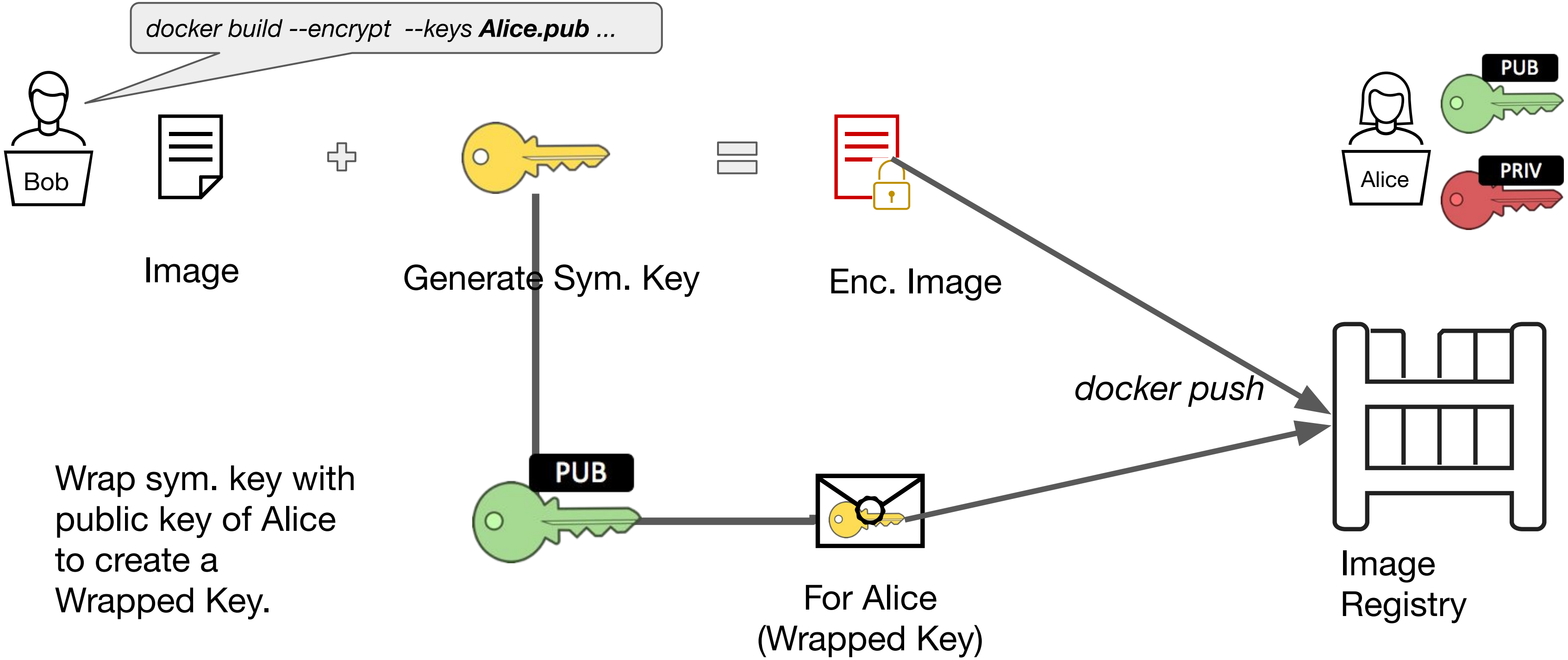
```
docker build --encrypt  --keys Alice.pub ...
```

Encrypt and distribution flow



Wrap sym. key with public key of Alice to create a Wrapped Key.

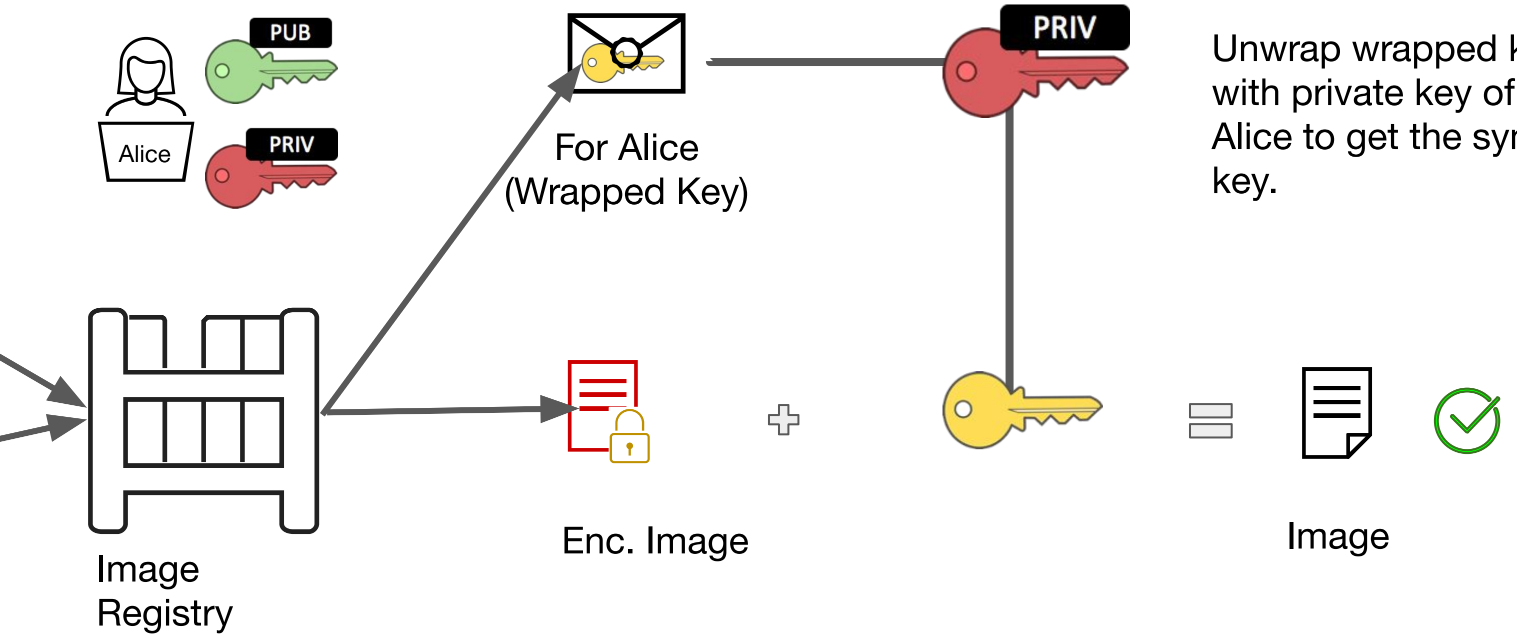
Encrypt and distribution flow



*docker build --encrypt --keys **Alice.pub** ...*

Wrap sym. key with public key of Alice to create a Wrapped Key.

Encrypt and distribution flow



Unwrap wrapped key with private key of Alice to get the sym. key.

Spec Details

OCI Proposal

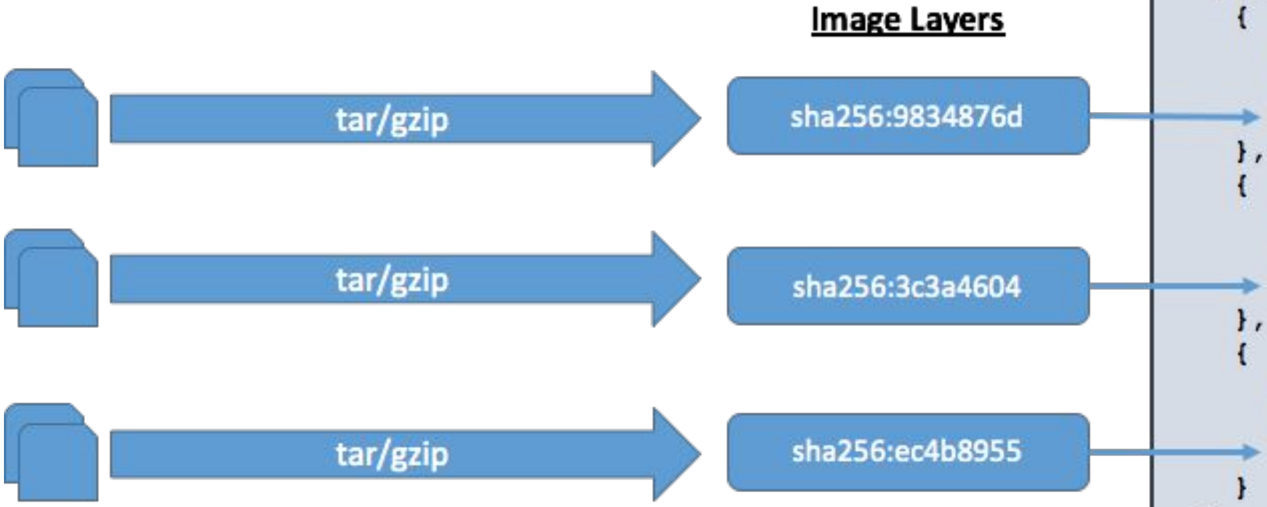


Image Spec

```
{
  "schemaVersion": 2,
  "config": {
    "mediaType":
"...image.config.v1+json",
    "size": 7023,
    "digest": "sha256:b5b2b2c507..."
  },
  "layers": [
    {
      "mediaType": "...tar+gzip",
      "size": 32654,
      "digest": "sha256:9834876d..."
    },
    {
      "mediaType": "...tar+gzip",
      "size": 16724,
      "digest": "sha256:3c3a4604..."
    },
    {
      "mediaType": "...tar+gzip",
      "size": 73109,
      "digest": "sha256:ec4b8955..."
    }
  ],
}
```

OCI Proposal

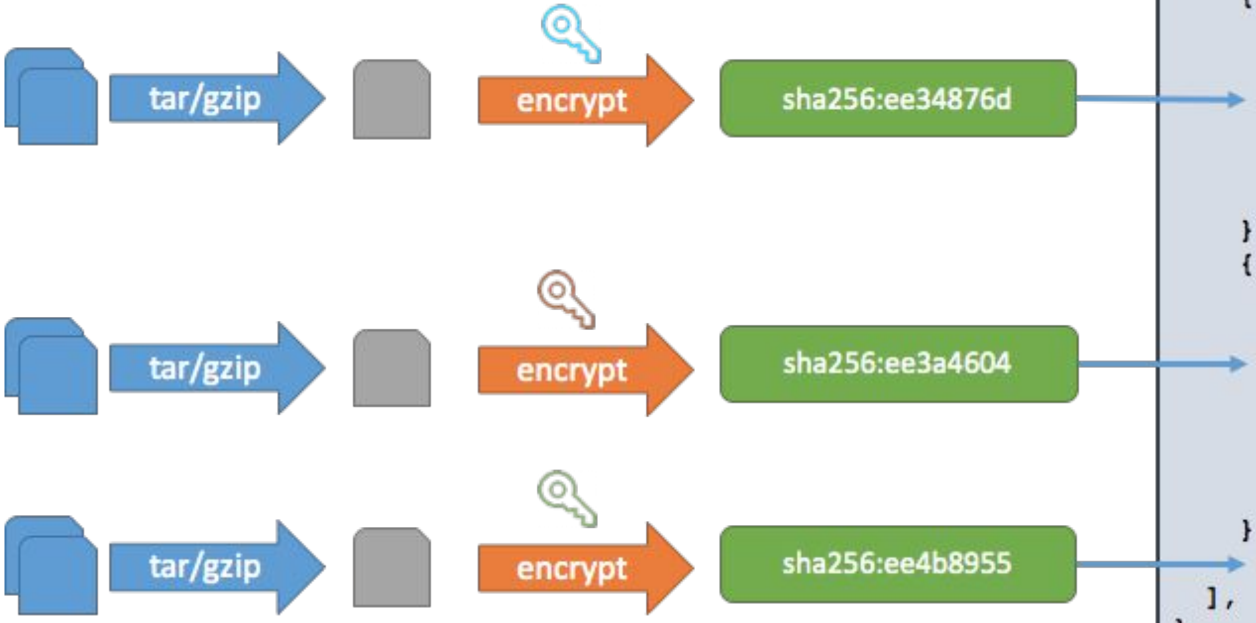
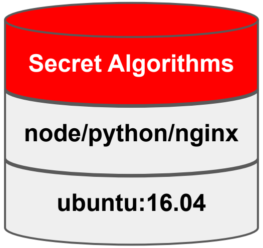


Image Spec

```
{
  "schemaVersion": 2,
  "config": {
 ...
  },
  "layers": [
    {
      "mediaType": "...tar+gzip+enc",
      "size": 32654,
      "digest": "sha256:ee34876d..."
      "annotations": {
        "org…enc.keys": "eefg9...",
      }
    },
    {
      "mediaType": "...tar+gzip+enc",
      "size": 16724,
      "digest": "sha256:ee3a4604..."
      "annotation": {
        "org…enc.keys": "bfc3a...,
                        da8be..."
        }
    },
    ...
  ],
}
```

Container Encryption Features

- Encrypt on layers means images can still benefit from deduplication of non-sensitive layers

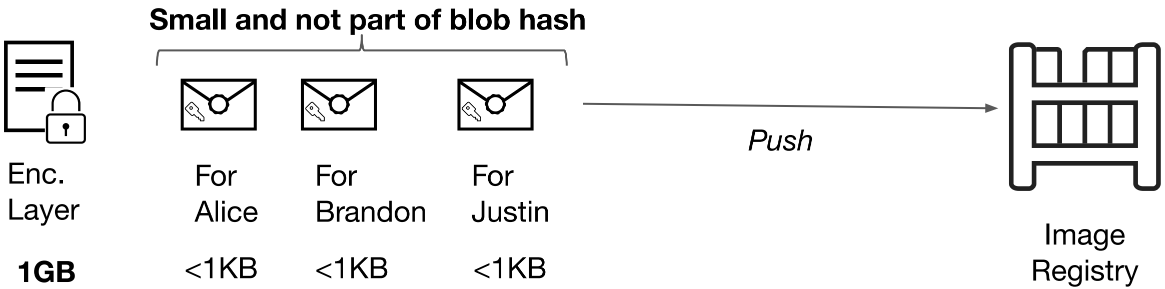


Secret Algorithms

node/python/nginx

ubuntu:16.04

- Encrypt once for multiple recipients. Registry deduplication on large encrypted data blob.



Small and not part of blob hash

Enc. Layer

1GB

For Alice

<1KB

For Brandon

<1KB

For Justin

<1KB

Push

Image Registry

Future Work

Encryption in OCI Image Spec

Issue: https://github.com/opencontainers/image-spec/issues/747
PR: https://github.com/opencontainers/image-spec/pull/775

Tighter Integration with ecosystem - k8s/docker/etc.

ImageDecryptSecrets KEP: https://github.com/kubernetes/enhancements/issues/1067
Moby: https://github.com/moby/moby/issues/38043,
https://github.com/moby/buildkit/issues/714
Cri-o: https://github.com/containers/image/issues/634

Hardware Encryption & FIPS compliance

Hardware Encryption: With TPMs, HSMs, TEEs and Key Management Key Wrapping Services
Support Enterprise FIPS: Ability to plugin FIPS compliant components for enterprise use