
SciLifeLab

PROTEGER LAS CARGAS DE TRABAJO DE KUBERNETES: DEL CÓDIGO AL CLÚSTER

XOPS CONFERENCE 2024

ÁLVARO REVUELTA

SciLifeLab

ABOUT ME

Alvaro Revuelta.

- Systems Developer
- Laboratory for Life Sciences - SciLifeLab

SciLifeLab

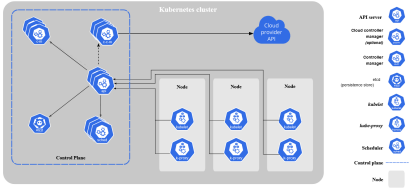
PRESENTATION STRUCTURE

1. Introduction
2. Inner Loop
3. Outer Loop
4. Other Security challenges
5. Other tools for security
6. Final

SciLifeLab

INTRODUCTION - ASSUMPTIONS

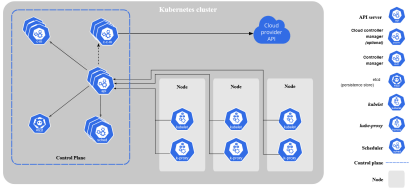
- Talk will cover both, developer and platform. But focus a bit more in security from a developer perspective.
- Assumes knowledge about containers and DevOps principles. Some knowledge about Kubernetes is also useful.



SciLifeLab

INTRODUCTION - WHY KUBERNETES

- Container management manually is difficult, leading to the rise of orchestration platforms like Kubernetes.
- It is a highly flexible tool, thus, it doesn't make any assumptions and the engineers have to be aware of all the possible vulnerabilities.

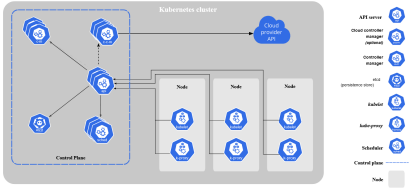


SciLifeLab

INTRODUCTION - WHY KUBERNETES

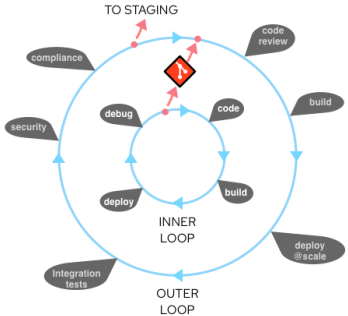
Perception on security issues in Kubernetes are underestimated.

- **53 %** of Organizations using Kubernetes experienced security issues; **55%** delayed deployments
  redhat.com/rhdc/managed-files/cl-state-of-kubernetes-security-report-2022-ebook-f31209-202205-en.pdf.
- **63%** of public code templates had improper configurations; **96%** of cloud applications had known vulnerabilities,
  unit42.paloaltonetworks.com/cloud-threat-report-2h-2021.



SciLifeLab

INTRODUCTION - INNER OUTER DEVELOPMENT LOOP



SOURCE: REDHAT

SciLifeLab

INNER LOOP

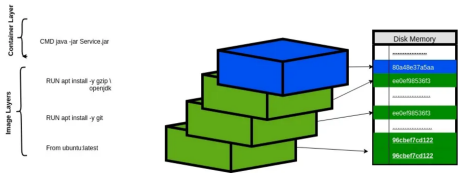
1. Write Code.
2. Build (containerize the application).
3. Debug Locally.

- This is where the Developer spends most of their time.

SciLifeLab

INNER LOOP: KEY CONSIDERATIONS

- Use a secure base image.
- Scan code dependencies for vulnerabilities (e.g., Trivy, Snyk).
- Optimize the application builds by doing good use of concepts such as the cache of layers.
- Increase productivity with Editor/IDE integration: VSCode Remote Containers



(**Source**: UnionFS : A File System of a Container)

SciLifeLab

INNER LOOP: SCAN IMAGE

trivy image imagetag:version

```
/ # trivy image rv0lt/flaskrediswebapp:basic | head
2023-04-21T18:04:09.297Z        INFO    Vulnerability scanning is en
2023-04-21T18:04:09.297Z        INFO    Secret scanning is enabled
2023-04-21T18:04:09.297Z        INFO    If your scanning is slow, pl
2023-04-21T18:04:09.297Z        INFO    Please see also https://aqua
ret detection
2023-04-21T18:04:11.456Z        INFO    Detected OS: debian
2023-04-21T18:04:11.456Z        INFO    Detecting Debian vulnerabili
2023-04-21T18:04:11.691Z        INFO    Number of language-specific
2023-04-21T18:04:11.691Z        INFO    Detecting python-pkg vulnera

rv0lt/flaskrediswebapp:basic (debian 10.13)
===========================================
Total: 146 (UNKNOWN: 5, LOW: 86, MEDIUM: 21, HIGH: 32, CRITICAL: 2)
```

(github.com/aquasecurity/trivy)

SciLifeLab

INNER LOOP: OPTIMIZE IMAGES

Minimizing the Number of Layers:

- Combine multiple RUN commands into one (e.g., chaining commands using &&).
- Removing unnecessary files during the build process (e.g., apt-get clean).
- Avoid using COPY to add files that won't be needed during runtime.

Using Smaller Base Images:

- Alpine Linux vs. Ubuntu: Trade-offs between size and functionality.
- Multi-stage builds: Using one image for building the application and another for the final image (stripped of development tools).

Caching Layers Efficiently:

- Take advantage of layer caching by ordering Dockerfile instructions logically.
- Place less frequently changed commands earlier in the Dockerfile to maximize caching.

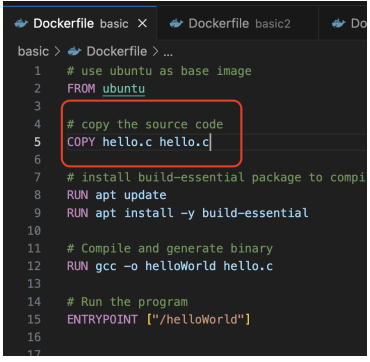
Removing Unnecessary Packages and Files:

- Remove temporary files, build dependencies, or logs to reduce image size.
- Clean up the file system after installations (e.g., rm -rf /var/lib/apt/lists/*).

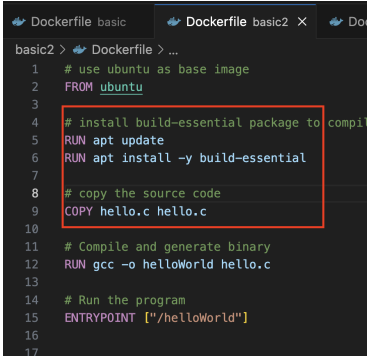
SciLifeLab

INNER LOOP: OPTIMIZE IMAGES

Subsequent building times: +60s



Subsequent building times: 1s



SciLifeLab

INNER LOOP: OPTIMIZE IMAGES

```
# use ubuntu as base image
FROM ubuntu as build-env

# install build-essential package to compile the source
RUN apt update && apt install -y build-essential

# copy the source code
COPY hello.c hello.c

# Compile and generate binary
RUN gcc -o helloWorld hello.c

# FROM alpine for an even greater size reduce
FROM ubuntu

# copy binary executable to new layer
COPY --from=build-env ./helloWorld ./helloWorld

# Run the program
ENTRYPOINT ["/helloWorld"]
```

WEIGHT IN MB REDUCED BY 4X TIMES

SciLifeLab

OUTER LOOP

1. Code Review
2. Automated testing
3. Production build
4. Compliance, and security checks,
5. Deployment to target environments.

- The platform engineer sets up the automation processes.

SciLifeLab

OUTER LOOP: CI - CD

Continuous Integration:
Build-Test-Merge.

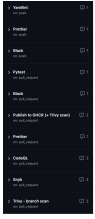
Continuous Delivery:
Someone decides when to push to prod.

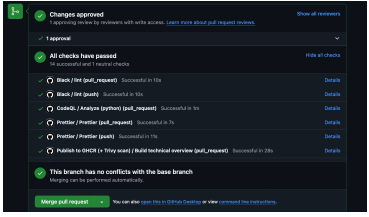
Continuous Deployment:
Automatically deploy to prod.

SciLifeLab

OUTER LOOP: SECURITY IN CI/CD

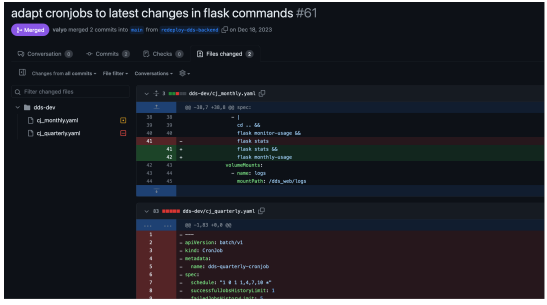
- Scan code and images during builds.
- Block deployments for high-severity vulnerabilities.

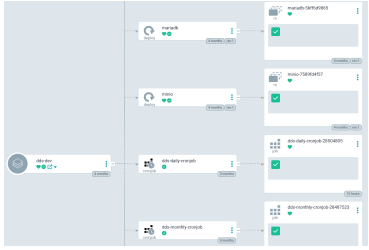




SciLifeLab

OUTER LOOP - GITOPS AND CD





ArgoCD

SciLifeLab

OUTER LOOP - SEALEDSECRETS

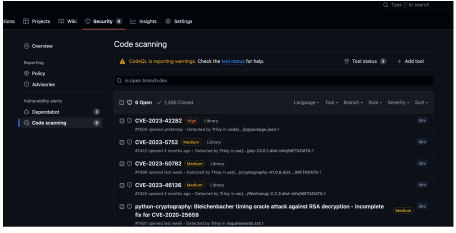
- The Kubernetes object Secrets, is not encrypted, it is encoded (base64)
- If we want to store our secrets in a repository, we can use sealed secrets.
- github.com/bitnami-labs/sealed-secrets
- *kubeseal -f secret.yaml -w sealsecret.yaml*

OUTER LOOP: CONTINUOUS VULNERABILITY MANAGEMENT

Why Continuous Scanning Matters

- The evolving threat landscape.
- New Vulnerabilities pop up quite often, CVE database

Integrate tools like Trivy into CI/CD pipelines to schedule scans of image repositories.



SciLifeLab

OTHER SECURITY CHALLENGES IN K8S

Default permissions

- By default, containers run as a root user.
- Malicious agents can exploit root access.
- We need to define security contexts to run as a non-privileged user and apply restrictions.

Communication between pods

- By default, all resources can communicate with each other.
- We need to ensure resource separation to provide better security in case of breach.
- The solution is to limit traffic flow for better control.

SciLifeLab

OTHER SECURITY CHALLENGES IN K8S

Service account tokens

- Containers are automatically created with a Token used to authenticate with the API.
- This means that, by default, containers can interact with the cluster and deploy resources.
- We need to disable mounting tokens if not needed (most cases).

```
[16]: ! mount | grep kubernetes

tmpfs on /run/secrets/kubernetes.io/serviceaccount type tmpfs (ro,relatime,size=12807532k)

[37]: ! cat /run/secrets/kubernetes.io/serviceaccount/token
```


eyJhbGciOiJSUzI1NiIsImtpZCI6IkhJRVBwZUpvQlVybWxQaXpXczc5QkpDOF9PZkVnUkRVT21iVmpZdGNfYlkifQ.eyJhdWQiOlsiaHR0cHM6Ly9r
dWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNzEzMDE5OTI3LCJpYXQiOjE2ODE0ODM5MjcsImlzcyI6Imh0dHBzOi8
va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJkZWZhdWx0IiwicG9kIjp7Im
5hbWUiOiJqdXAtNzhjNjg2NDk0ZC1mYmY0ciIsInVpZCI6IjJmMDkxMTdhLTQ2MWQtNDRmMy1hN2Q2LWRmZjhkYTk1ZmE4MCJ9LCJzZXJ2aWNlYWNjb
3VudCI6eyJuYW1lIjoiZGVmYXVsdCIsInVpZCI6IjM5MTMwNGE3LTk1MDYtNDQ0ZS05YzMwLWU2YzIzNjU0ODkyYiJ9LCJ3YXJuYWZ0ZXIiOjE2ODE0
ODc1MzR9LCJuYmYiOjE2ODE0ODM5MjcsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpkZWZhdWx0OmRlZmF1bHQifQ.KePNd-EJBUHidDncHiJM
NQSIWHyS0qmZe-ttUCaGJv9_RhatoychnioZdFjWWd3FeRdKDAnnoF-jZ7Y7JY0_80GknFgnUeZ5omUCTFs-ryvJlXDY34pOj1lY5NJEM8hNS5tljt6
xyH5FHWjnWn9WXHDNTX9ADLf_FaOSHoDncVwxN7_tp-hocpoHmC_KlQm8Ojies_znYXYShKBUN73YGtyIDAXWWanE1_0zOdTJdtHKMiVmOMJ_v_3ivs
pcgF1v8AdVGrPp049BEIQS1sabR_GasKC3624Kwjmxug6h_UIgqA16ESIZwTXEyVf2tJbNgU8C2e4WbhbcPQnre4Dv-g


```
[38]: ! cat /run/secrets/kubernetes.io/serviceaccount/namespace

default
```

VERY INTERESTING LECTURE

National Security Agency
Cybersecurity and Infrastructure Security Agency

Cybersecurity Technical Report

Kubernetes Hardening Guide

KUBERNETS HARDENING GUIDE

SciLifeLab

OTHER TOOLS FOR SECURITY

- Kube-hunter: Cluster Misconfiguration Detection; Pen-testing.
  - Same makers as Trivy
  - github.com/aquasecurity/kube-hunter
- Kubescape: similar than kube-hunter.
  - github.com/kubescape
- Kube-linter: Same as Trivy, but to find misconfiguration in k8s yaml files.
- Falco: Runtime Security.

Vulnerabilities
For further information about a vulnerability, search its ID in:
https://avd.aquasec.com/

ID	LOCATION	MITRE CATEGORY	VULNERABILITY	DESCRIPTION	EVIDENCE
None	Local to Pod (kube-hunter-n75v6)	Lateral Movement // ARP poisoning and IP spoofing	CAP_NET_RAW Enabled	CAP_NET_RAW is enabled by default for pods. If an attacker manages to compromise a pod, they could potentially take advantage of this capability to perform network attacks on other pods running on the same node	
KHV002	10.96.0.1:443	Initial Access // Exposed sensitive interfaces	K8s Version Disclosure	The kubernetes version could be obtained from the /version endpoint	v1.27.2

CNCF LANDSCAPE

OFFICIAL COMPLIANT KUBERNETES CNCF.IO

SciLifeLab

FINAL

CONTACT:
LINKEDIN - ALVARO REVUELTA MARTINEZ
ALVARO.REVUELTA@SCILIFELAB.UU.SE

SciLifeLab

FINAL

Q & A

SciLifeLab