DevOps Security: Security Best Practices

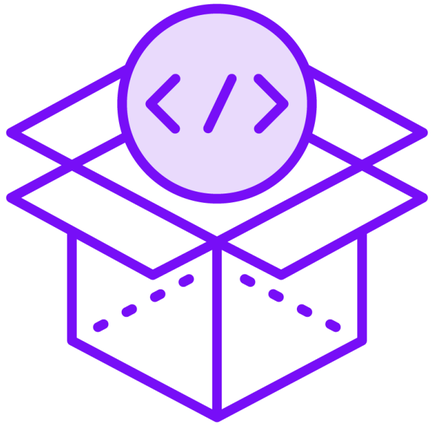
Implement Container Security Best Practices

Antonio J. Piedra
Senior DevOps Engineer

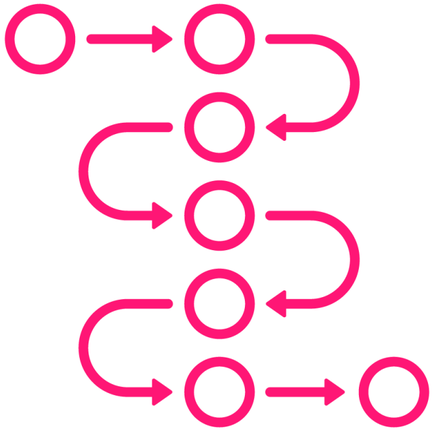
www.linkedin.com/in/ajpiedra

Secure Container Image Creation

Use official, minimal base images



Avoid installing unneeded packages, tools or dependencies



Use multi-stage builds

Sample Dockerfile

```
# Start with a minimal base image
FROM golang:1.24-alpine
```

Sample Dockerfile

```
# Start with a minimal base image
FROM golang:1.24-alpine

# Add app code
WORKDIR /app
COPY app .
```

Sample Dockerfile

```
# Start with a minimal base image
FROM golang:1.24-alpine

# Add app code
WORKDIR /app
COPY app .

# Build the application
RUN go build -o myapp .
```

Sample Dockerfile

```
# Start with a minimal base image
FROM golang:1.24-alpine

# Add app code
WORKDIR /app
COPY app .

# Build the application
RUN go build -o myapp .

# Install curl
RUN apk add --no-cache curl
```

Sample Dockerfile

```
# Start with a minimal base image
FROM golang:1.24-alpine

# Add app code
WORKDIR /app
COPY app .

# Build the application
RUN go build -o myapp .

# Install curl
RUN apk add --no-cache curl
```

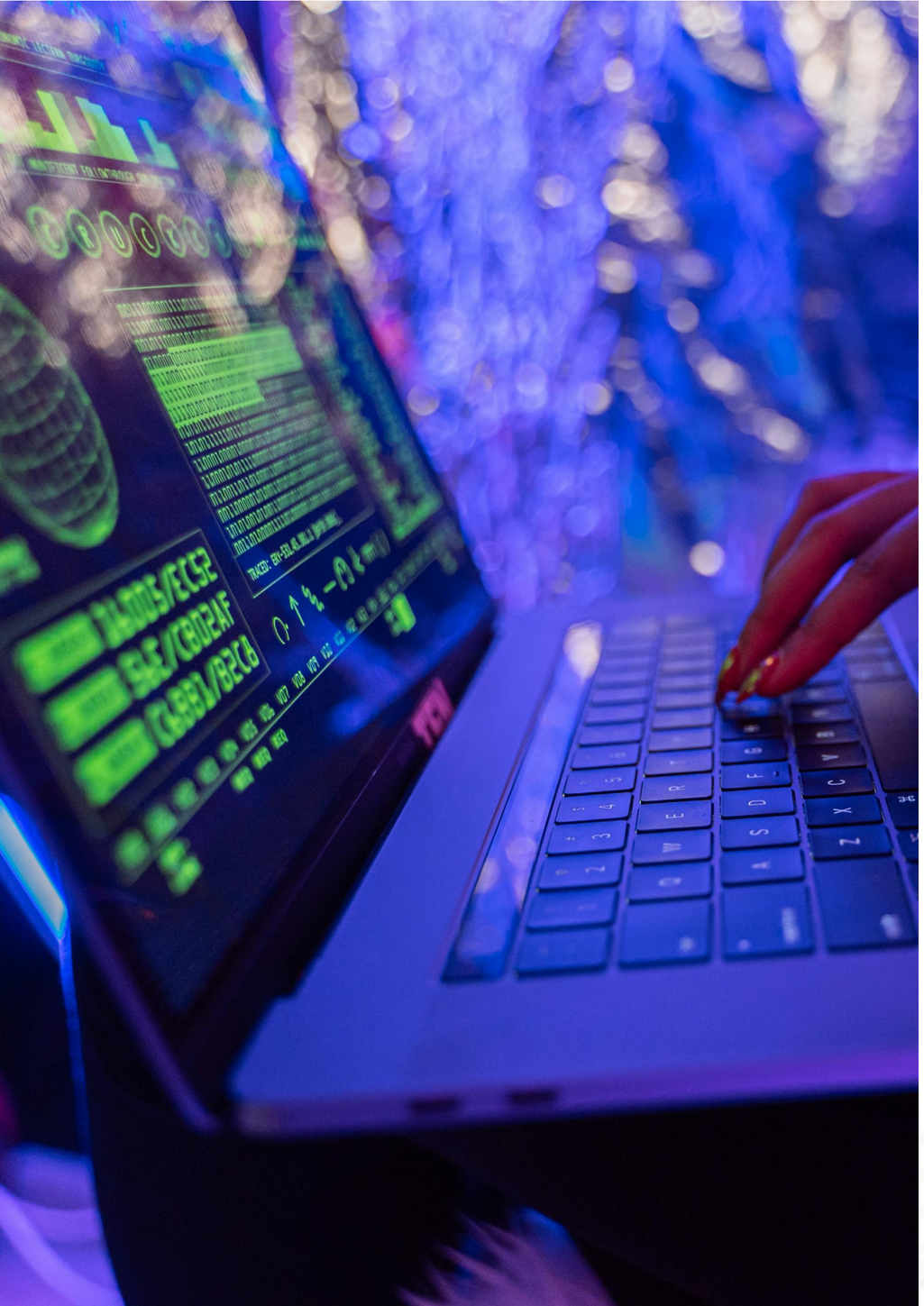
Sample Dockerfile

```
# Start with a minimal base image
FROM golang:1.24-alpine

# Add app code
WORKDIR /app
COPY app .

# Build the application
RUN go build -o myapp .

# Install curl
RUN apk add --no-cache curl

# Command to run the application
CMD ["./myapp"]
```

Multi-stage Dockerfile

```
# Stage 1: Build the application
FROM golang:1.24-alpine AS builder  # 76.76MB
WORKDIR /app
COPY app .
RUN go build -o myapp .

# Stage 2: Create the minimal final image
FROM alpine:3.21 # 3.6MB
WORKDIR /app
COPY --from=builder /app/myapp .
CMD ["./myapp"]
```

Smaller images mean reduced attack surface and faster pulls from registry.



Scan Container Images for Vulnerabilities

Use tools like Trivy, Grype, and Anchore

Scan as part of CI/CD pipelines

After deployment, keep monitoring

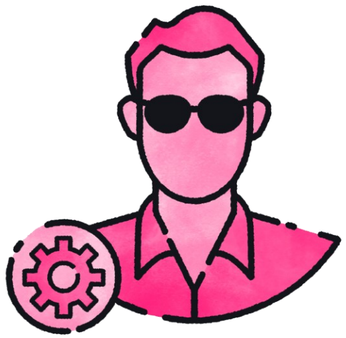
- New CVEs are discovered daily!

Runtime Security

Secure how your containers run.

Principle of least privilege

Drop capabilities and set filesystem to read-only



User namespaces or random UIDs/GIDs

Run containers as non-root users

Limiting resources used

Set memory and CPU boundaries

Demo: Scan Container Images with Trivy

Detect vulnerabilities before deployment

Demo: Create Secure Container Images

Reduce the risk of security breaches

Demo: Run Containers Securely – Drop Capabilities

Follow the principle of least privilege

Demo: Run Containers Securely – Read Only Filesystem

Prevent unauthorized changes in the container

Demo: Run Containers Securely – Define Resource Limits

Prevent resource exhaustion and ensure container stability

Demo: Run Containers Securely – Specify Running User

Define dedicates, unprivileged user for containers processes