Do you know what your containers are made of?

@mossnz

h*ck yeah containers

@alicegoldfuss says that
containers are *processes*
born from *tarballs*
anchored to *namespaces*
controlled by *cgroups*

Abstractions are friends

Today we'll be dealing with

- docker pull
- docker push
- docker build
- docker run
- docker-compose up

All good code needs a good problem

you have 1337 limes. definitely hax but ok.

let's solve this by uploading the limes to the cloud.

https://try.redis.io

```
> incr limes
(integer) 1
> incr limes
(integer) 2
```

https://hub.docker.com/_/redis/

$ docker run --name some-redis -d redis:latest

```
aa83e03d773a559f8a020e8d0b8068b797551f58060bb2a77476bb559e51b5ae
```

$ docker run -it --link some-redis:redis --rm redis redis-cli -h redis -p 6379

```
redis:6379> incr limes
(integer) 1
redis:6379> incr limes
(integer) 2
```

```
$ node index.js
```

```
const express = require('express');
const redis = require('redis');
```

```
express.put('/limes', (res) => {
  redis.incr('limes', (reply) => {
    const limes = reply || 0;
    res.json({
      limes,
    });
  });
});
```

```
GET /limes
{limes: "0"}

PUT /limes
{limes: 1}

POST /limes
Cannot POST /limes/
```

Dockerfile

```
FROM node:8 as base

WORKDIR /app
RUN npm install -g npm@6.4.1
COPY package*.json ./

FROM base as dependencies
RUN npm ci --production --loglevel http

FROM base as production
COPY --from=dependencies /app/node_modules ./node_modules
COPY . .
EXPOSE 1337
CMD ["node", "index.js"]
```

docker-compose.yml

```
services:
 app:
  image:          limes:local
 container_name: limes_app
 depends_on:
   - redis

redis:
 image:          redis:latest
 container_name: limes_redis
```

How to draw an Owl.

"A fun and creative guide for beginners"

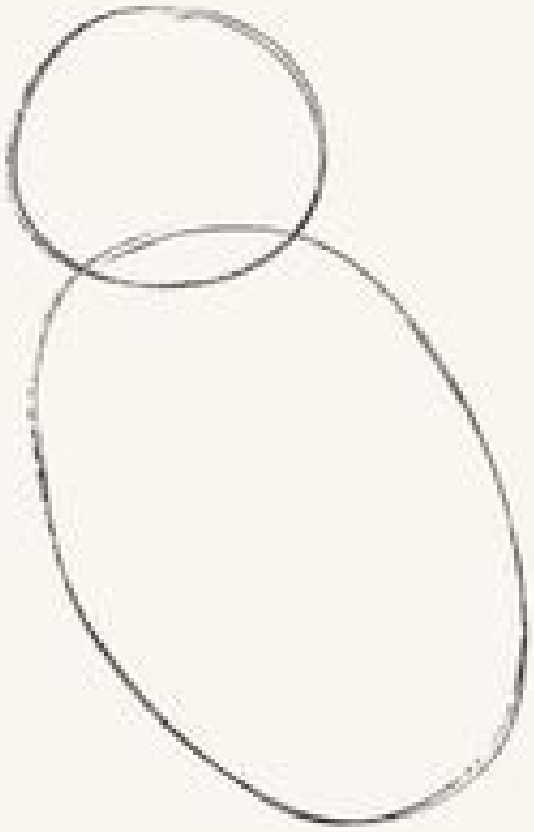


Fig 1. Draw two circles

Fig 2. Draw the rest of the damn Owl

Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

— Ken Thompson. 1984. (Turing Award Lecture)

1. Programs can write programs.

2. The C compiler is written in C

3. We can modify a compiler to deliberately alter programs

The actual bug I planted in the compiler would match code in the UNIX "login" command. The replacement code would miscompile the login command so that it would accept a particular known password.

3. We can modify a compiler to deliberately alter programs

... including the compiler, that compiles the compiler.

The moral of this story is you can't trust code you didn't build yourself.

If you wish to make an apple pie from scratch, you must first invent the universe.

— Carl Sagan. Cosmos.

Minimally Viable Pipeline

By introducing a CI Pipeline, we can make small steps towards trusting our containers

Dockerfile

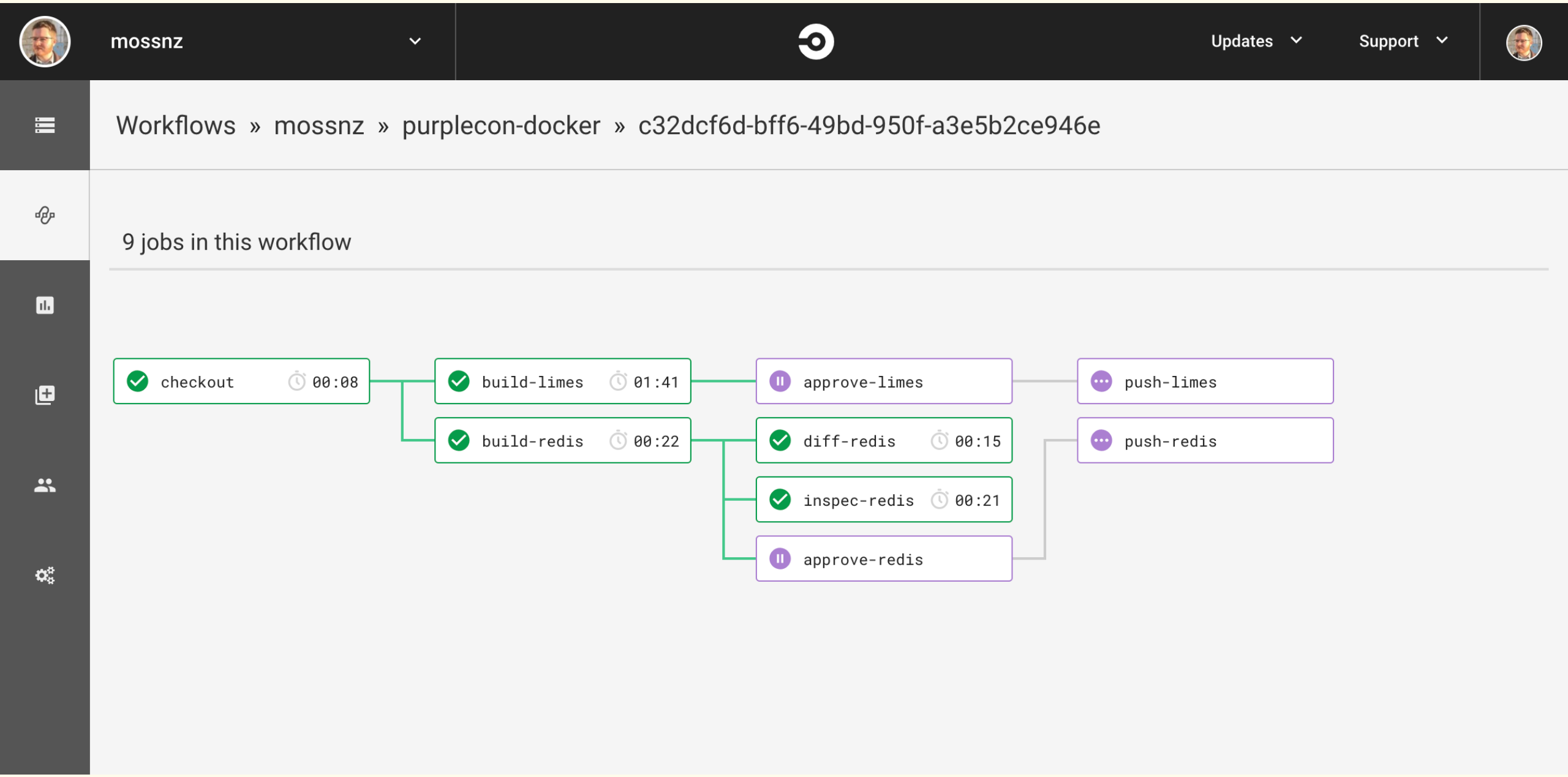
```
FROM redis:latest
```

```
$ docker build -t purplecon-redis:production
app/services/redis/.
```

CI Pipeline

After we build an image, we can gain trust by running

- **container-diff**, a Google open source tool
- **inspec**, Compliance as Code tooling
- **anchore** or **clair**, vulnerability scanners for Docker


mossnz
Updates
Support
Workflows » mossnz » purplecon-docker » c32dcf6d-bff6-49bd-950f-a3e5b2ce946e
9 jobs in this workflow
checkout 00:08
build-limes 01:41
approve-limes
push-limes
build-redis 00:22
diff-redis 00:15
inspec-redis 00:21
approve-redis
push-redis

https://github.com/GoogleContainerTools/container-diff

$ container-diff diff redis.tar remote://mossnz/

purple-redis:production --type=file

```
These entries have been added to /redis.tar: None

These entries have been deleted from /redis.tar: None

These entries have been changed between /redis.tar and purple-redis: None
```

https://www.inspec.io/

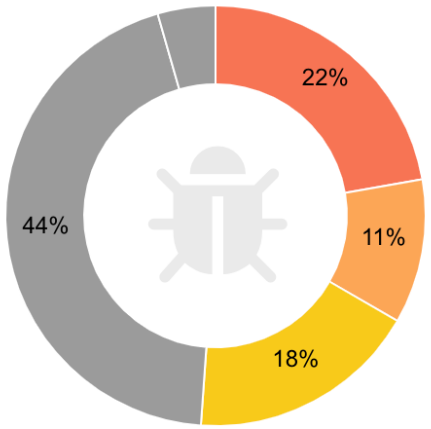
```
describe command('redis-cli ping') do
    its(:stdout) { should match(/PONG/) }
end
```

```
$ inspec exec test -t docker://purple-redis

  Command: `redis-cli ping`
     ✔  stdout should match /PONG/

Test Summary: 1 successful, 0 failures, 0 skipped
```

e8ca43ad4739



Quay Security Scanner has detected **45** vulnerabilities.

Patches are available for **1** vulnerabilities.

- **10** High-level vulnerabilities.
- **5** Medium-level vulnerabilities.
- **8** Low-level vulnerabilities.
- **20** Negligible-level vulnerabilities.
- **2** Unknown-level vulnerabilities.

Vulnerabilities

Filter Vulnerabilities... **Only show fixable**

CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN LAYER
CVE-2017-16997	9.3 / 10	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…
CVE-2017-8804	High	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…
CVE-2017-1000408	High	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…
CVE-2018-6551	High	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…
CVE-2018-6485	High	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…
CVE-2018-1000001	High	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…
CVE-2017-18269	High	glibc	2.24-11+deb9u3	(None)	ADD file:f8f26d117bc4a9289b7cd7447ca36e1a70b…

:latest is not your friend

The **:latest** tag isn't what it seems.

The **:latest** tag is applied to any build that *doesn't* have a tag.

Which breaks people's expectations and environments if you rely on it.

Dockerfile

```
- FROM redis:latest
+ FROM redis:5.0.1-alpine3.8
```

$ container-diff diff redis.tar remote://mossnz/purple-redis:production --type=file

```
These entries have been added to redis.tar: None
These entries have been deleted from redis.tar:
FILE
/bin
/bin/arch
/bin/base64
/bin/busybox
...
```

$ inspec exec test -t docker://purple-redis

```
Command: `redis-cli ping`
     ✔  stdout should match /PONG/

Test Summary: 1 successful, 0 failures, 0 skipped
```

Repositories

mossnz / purplecon-redis

Repository Tags

Compact Expanded

1 - 2 of 2

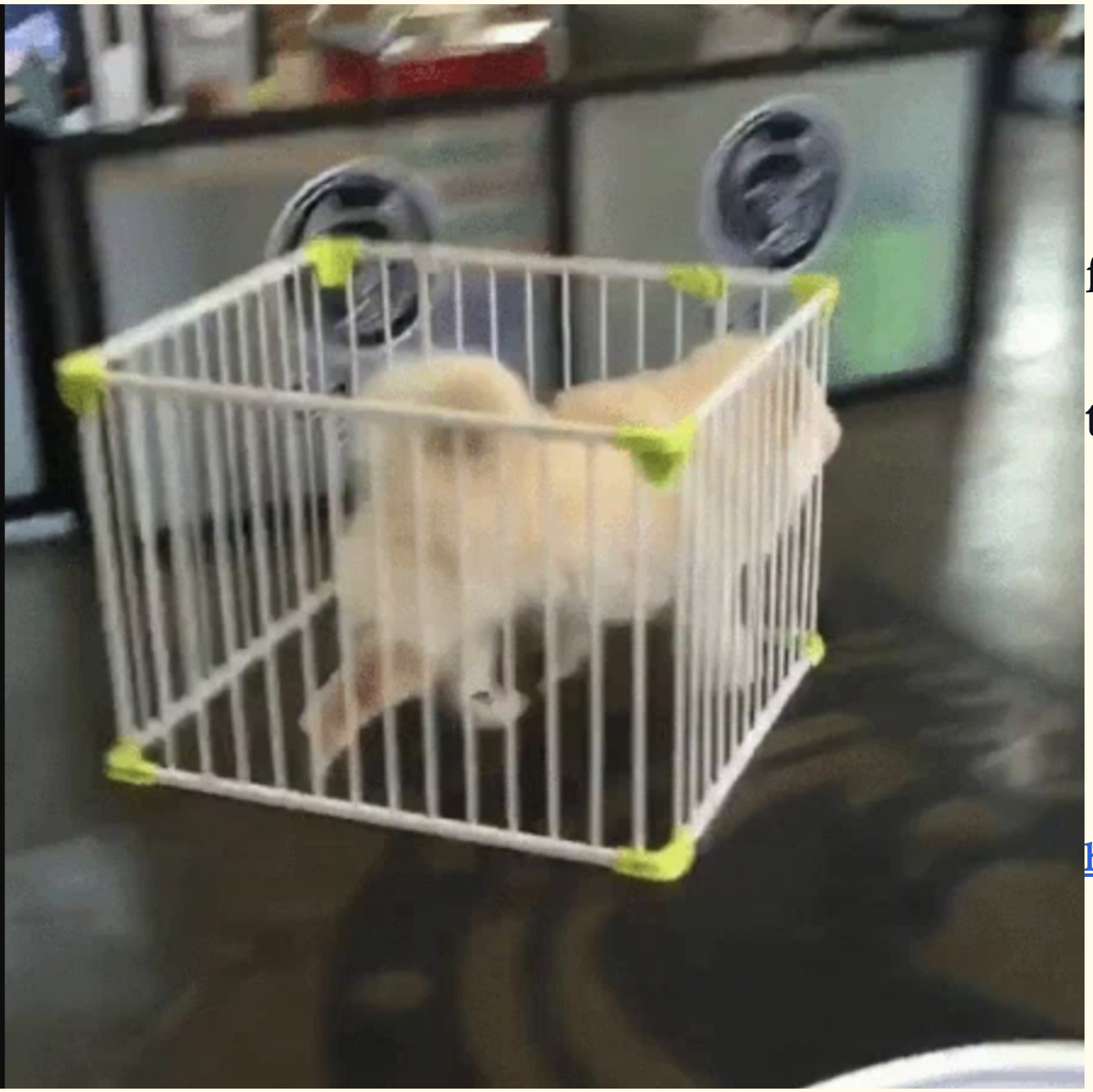
Filter Tags...

TAG	SECURITY SCAN	SIZE	EXPIRES	IMAGE
stage6	Passed	14.2 MB	Never	SHA256 1a0c599b422f
stage5	10 High • 1 fixable	33.6 MB	Never	SHA256 e8ca43ad4739

docker-compose.yml

```
services:
 app:
  image:          quay.io/mossnz/purplecon-limes:stage6
 container_name: limes_app
 depends_on:
   - redis

redis:
 image:          quay.io/mossnz/purplecon-redis:stage6
 container_name: limes_redis
```



from **redis:latest**

to **mossnz/purplecon-redis:stage6**

- Diffing
- Compliance as Code
- Vulnerability Scanning

https://github.com/mossnz/purplecon-docker