STL!/unix/usr/group

OMNITEC Corporation

# Let’s Encrypt and Containerized Websites

Main Meeting

13 May 2020

Lee Lammert
Original work by Chuck Doolittle

## Outline

- What exactly is Lets Encrypt?
- Obtaining a Certificate with LE
- Websites in Containers, ..
- Configuring LE
- Configuring a Website for LE
- Automating the process

## Lets Encrupt

From Wikipedia:

Let's Encrypt is a non-profit certificate authority run by Internet Security Research Group (ISRG) that provides X.509 certificates for Transport Layer Security (TLS) encryption at no charge.

*From a practical standpoint, five years ago the process required a somewhat convoluted shell script, but now it’s simple – install certtbot and get a cert!*

## Obtaining a new LE cert

Requirements:

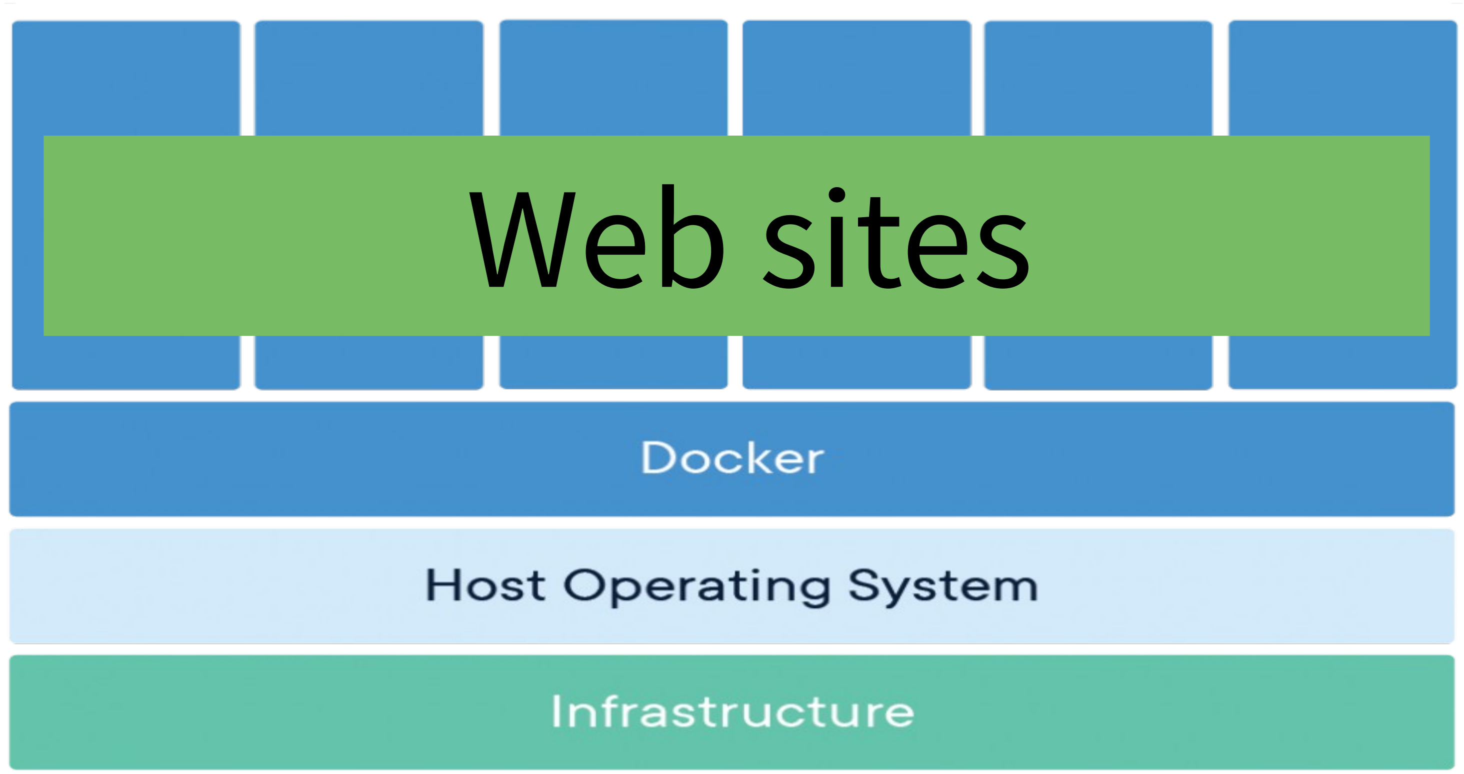
- Working FQDN (LE must retrieve the challenge)
- Configure  the server
- Certbot CAN configure automatically based on apache or nginx configuration, but it’s a lot simpler (and easier not to break anything) to do it by hand:

```
certbot certonly -d <fqdn>
```

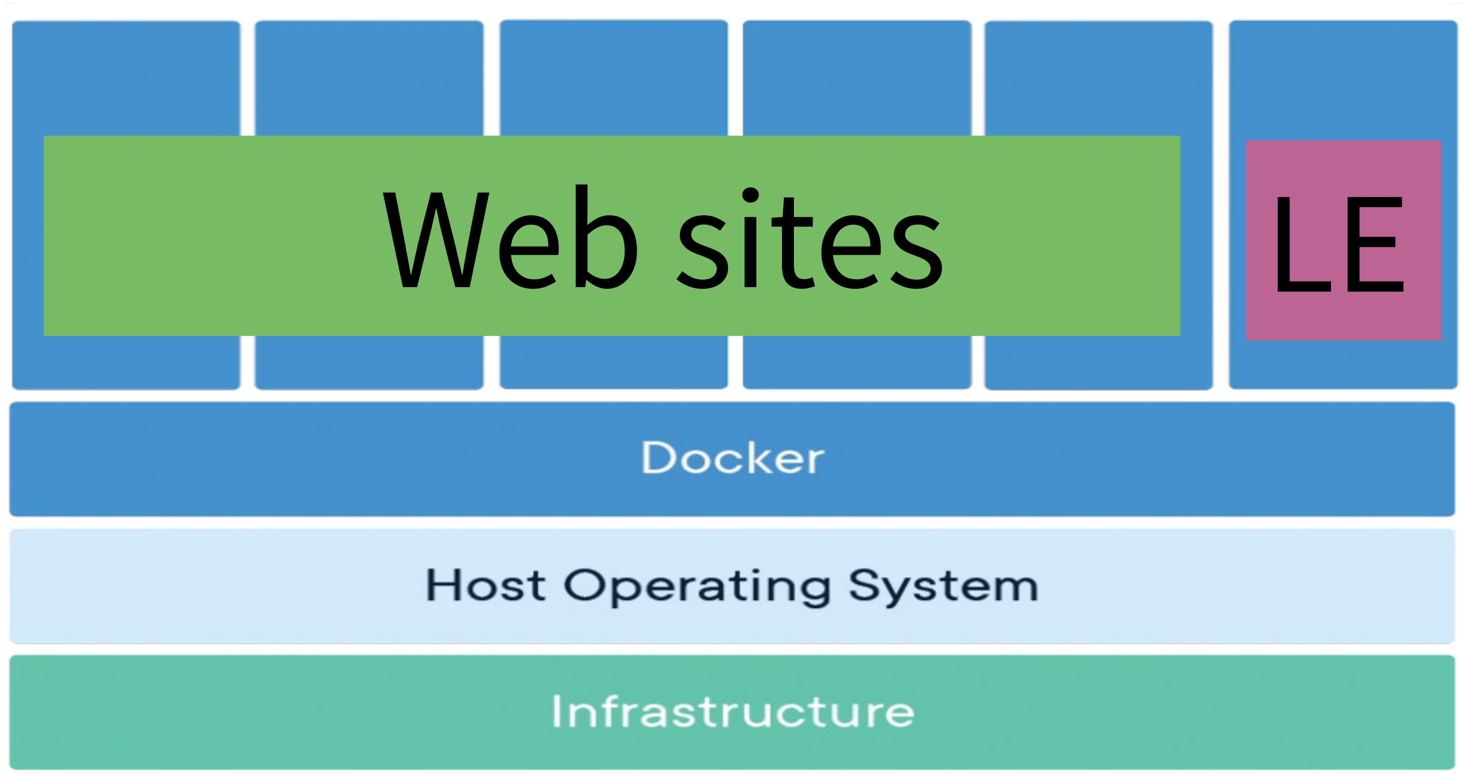
## Lets Encrypt data - /etc/letsencrypt

```
drwx-w----   3 root root   42 Aug 26  2019 accounts/
drwx------ 11 root root  218 Apr 16 11:13 archive/
drwxrwxr-x  2 root root 8192 Apr 16 11:13 csr/
drwx-w----  2 root root 8192 Apr 16 11:13 keys/
drwx------  9 root root  187 Apr 16 11:13 live/
-rw-rw-r--  1 root root 1143 Aug 27  2019 options-ssl-nginx.conf
drwxrwxr-x  2 root root  263 Apr 16 11:13 renewal/
drwxrwxr-x  5 root root   43 Aug 26  2019 renewal-hooks/
-rw-rw-r--  1 root root  424 Aug 27  2019 ssl-dhparams.pem
```

## So, .. what about Containers?



## Lets Encrypt is just another container!



## Instantiate an LE Container!

```
docker run --rm --name omni_certs \
        -v /u/SharedData/certbot/conf:/etc/letsencrypt \
        -v /u/SharedData/certbot/www:/var/www/certbot \
        -v /u/SharedData/certbot/log:/var/log \
        core:5000/omni_letsencrypt:1.0 \
        certonly --webroot -w /var/www/certbot \
        $DOMAIN \
        --rsa-key-size 4096 \
        --agree-tos \
        --non-interactive \
--expand \
        -m noc@omnitec.net
```

## Create persistant storage for LE & web sites

- **For certbot:**

  `/u/SharedData/certbot/conf:/etc/letsencrypt`

- **`For web site:`**

  `/u/SharedData/<website>/conf:/etc/letsencrypt`

  ***`or is it???`***

- **`Where is the cert required?7`**

## In the LoadBalancer!

- The cert isn't actually used in the website container, rather it's used in the LoadBalancer, where the external connection terminates!

```
server {
    listen 80;
    server_name ccsl.org www.ccsl.org;
    location / {
    return 301 https://ccsl.org$request_uri;
}
location /.well-known/acme-challenge/ {
            root /var/www/certbot;
    }
}
```

## Automating the process

- **Create a script to:**
  - Parse the domain(s)
  - Instantiate the container
  - If expired, renew
  - If a new cert, reload nginx

## Parse domain name(s)

```
#!/bin/bash
set -x

while [[ -n $1 ]]; do
DOMAIN="$DOMAIN -d $1"
shift
done
```

## Instantiate the container

```
docker run --rm --name omni_certs \
        -v /u/SharedData/certbot/conf:/etc/letsencrypt \
        -v /u/SharedData/certbot/www:/var/www/certbot \
        -v /u/SharedData/certbot/log:/var/log \
        core:5000/omni_letsencrypt:1.0 \
        certonly --webroot -w /var/www/certbot \
        $DOMAIN \
        --rsa-key-size 4096 \
        --agree-tos \
        --non-interactive \
--expand \
        -m noc@omnitec.net
```

## Check for an updated cert

```
LINES=`find /u/SharedData/certbot/conf/live -mmin -15|wc -l`

if [[ $LINES -gt 0 ]]; then
        docker exec loadbalancer nginx -s reload
fi
```

## Wait, .. there’s MORE!!!

Let’s do one for real:

- Verify DNS
- Create container config
- Obtain LE cert
- Go live!

Thanks for attending!

Lee Lammert

Omnitec Corporation