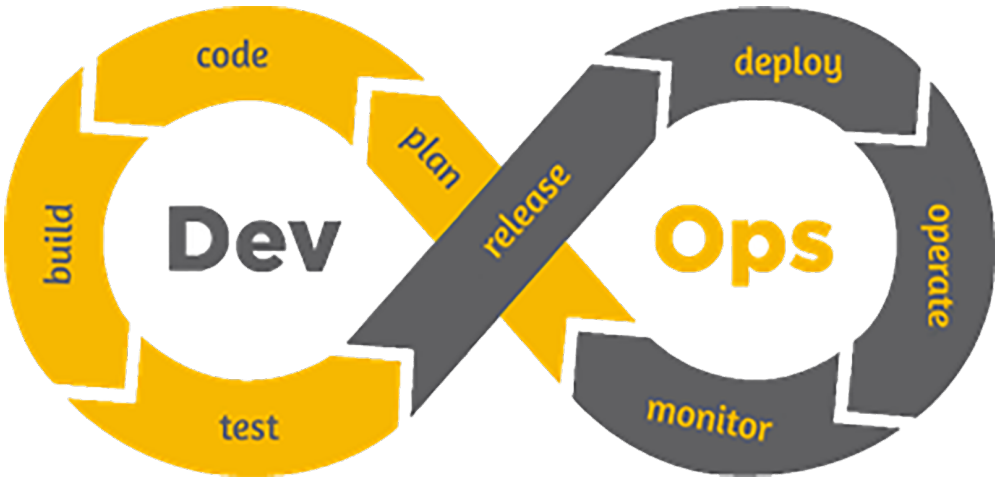
Red Team Operator by Open-Sec

DevOps para Red Teaming

Qué es DevOps ?

- Metodologías
  - CI/CD, telemetría, sistema de registros
- Tecnologías
  - Terraform, Ansible, Chef, Kubernetes, Jenkins
- Responsabilidad compartida
- Propiedad compartida



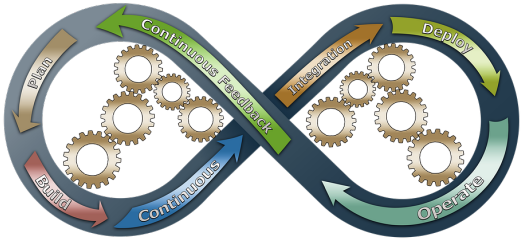
DevOps Shift Left

- Más rápido
- Más ágil
- Mejor calidad
- Ahorra tiempo

- Reduce los costos cuando hay que “re hacer”

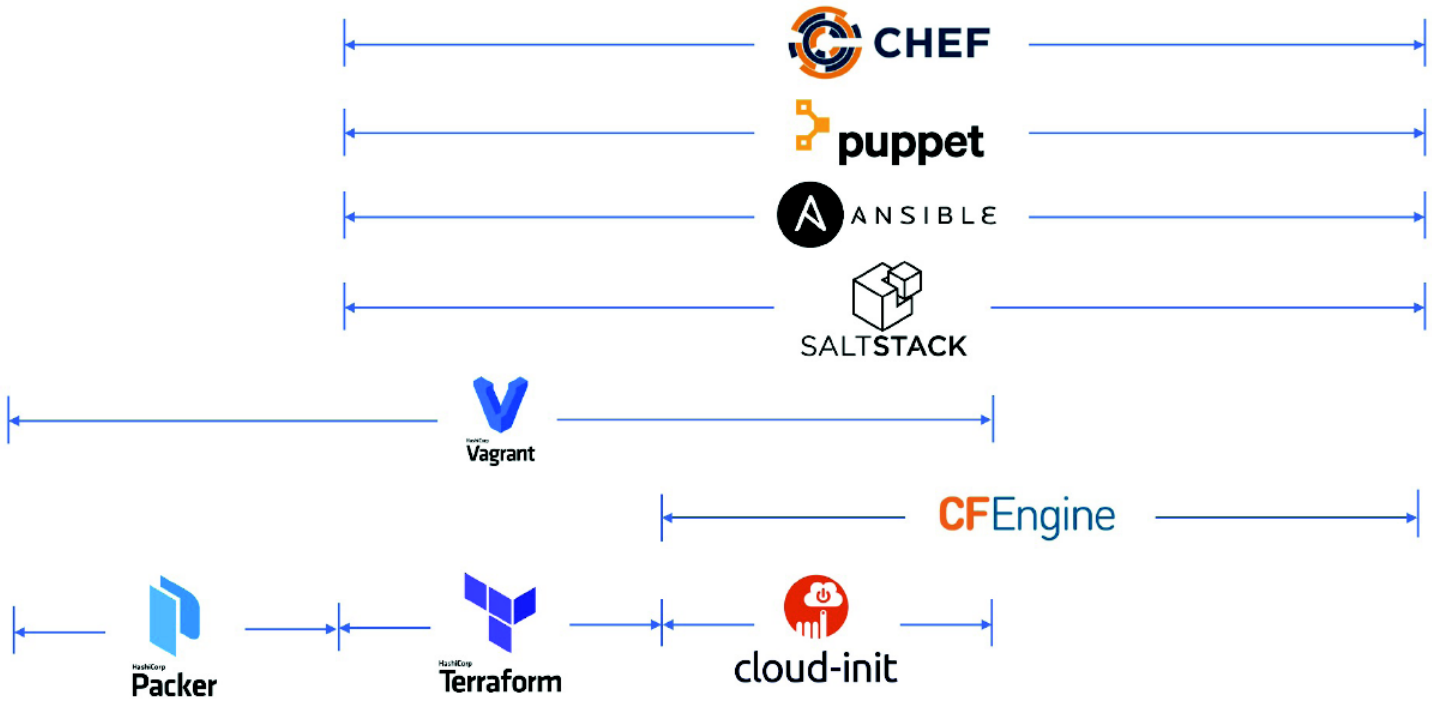
DevOps : Las 3 olas

- Principios de Flujo Downstream
- Principios de Feedback continuo
- Principios de Aprendizaje Continuo y Experimentación



Cómo se relaciona a Red Teaming ?

Infraestructura
Como
Servicio



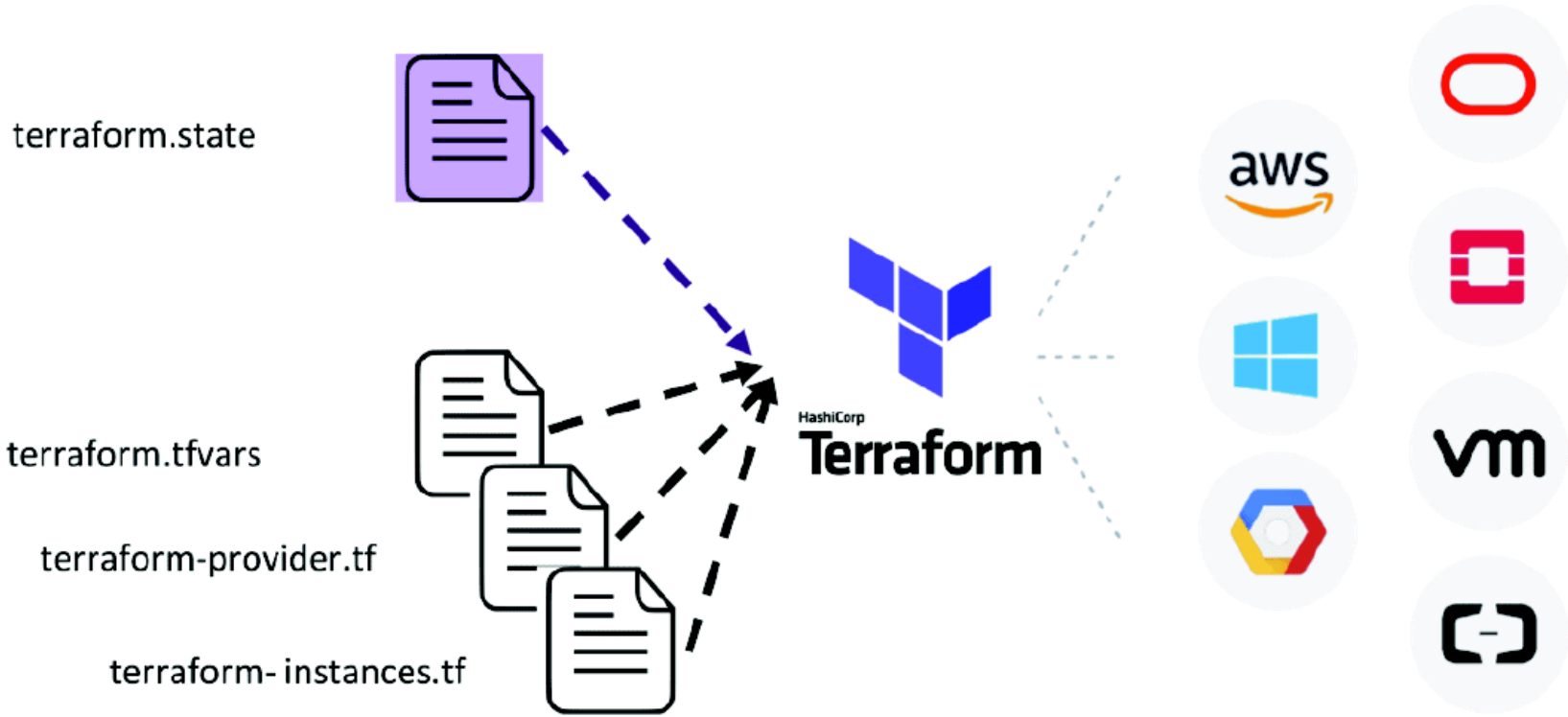
Plantillas Infraestructura

Manejo de Infraestructura

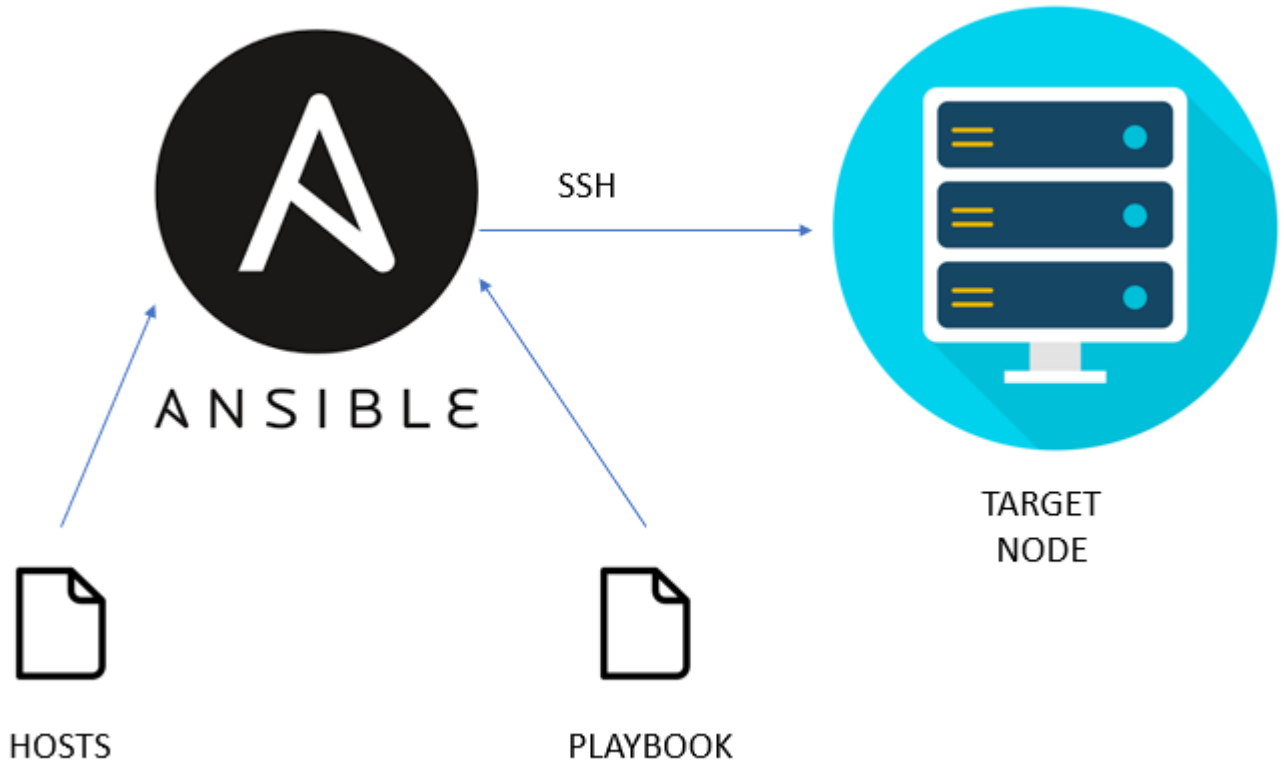
Instalar Aplicaciones y Configuraciones iniciales

Desplegar Configuraciones y Cambios post instalación

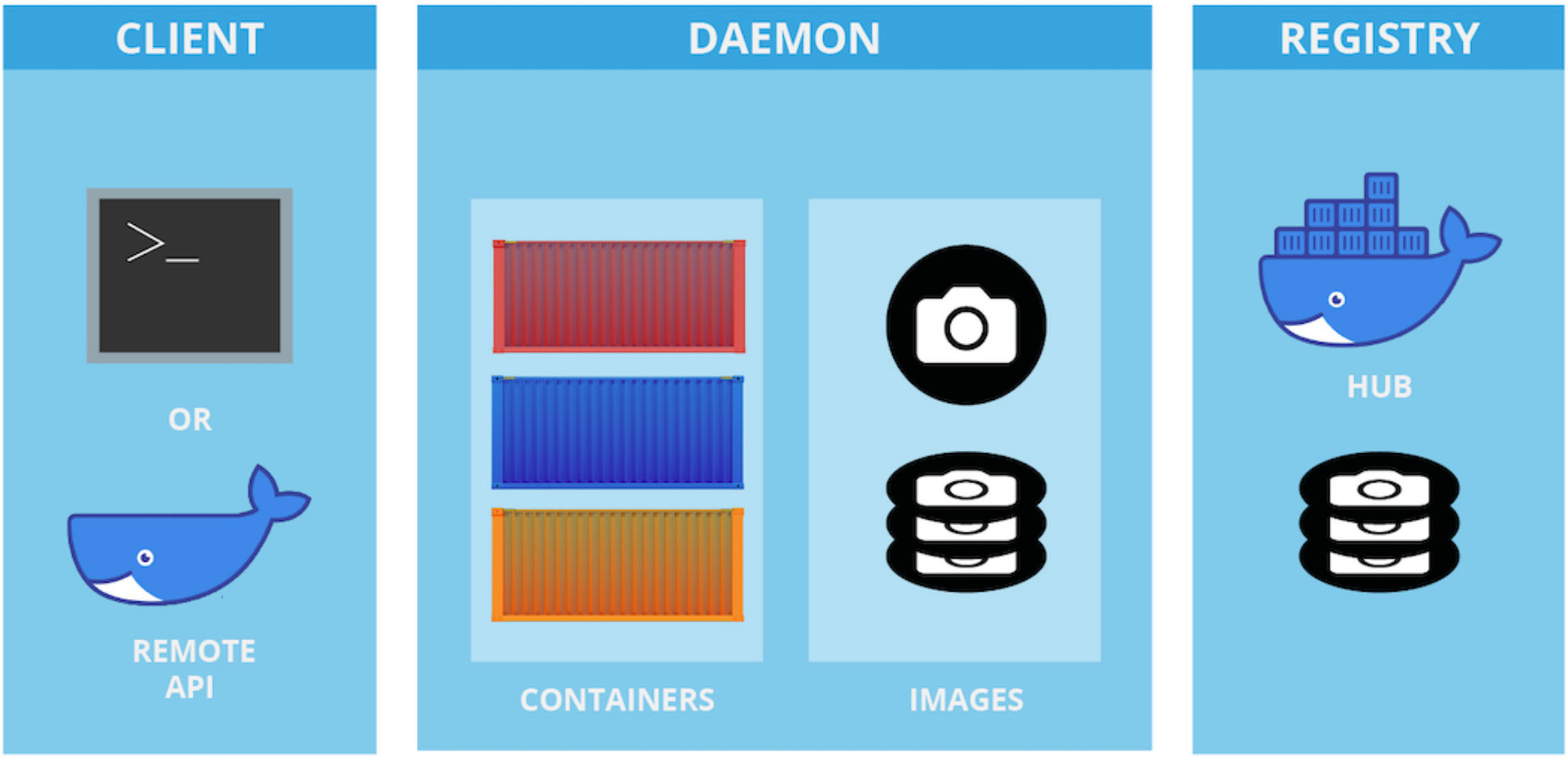
Terraform



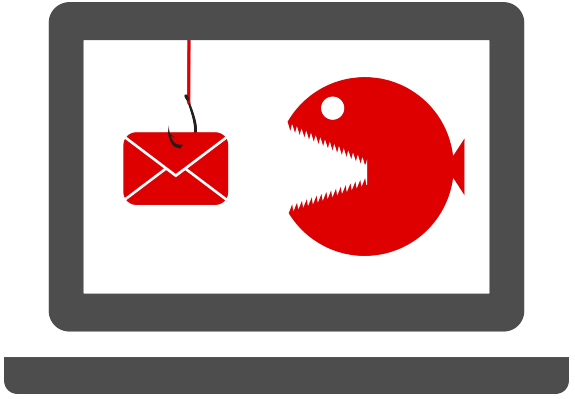
Ansible

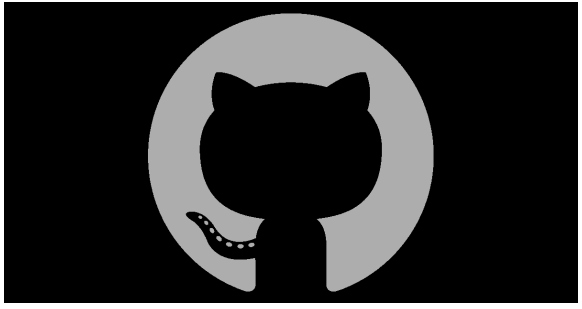


Docker


CLIENT
OR
REMOTE API
DAEMON
CONTAINERS
IMAGES
REGISTRY
HUB

Recursos y motivos





Recursos y motivos

Red Teamer

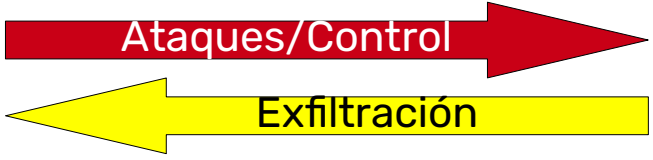
C2 Server

Proxy (mod_rewrite)

CDN
El endpoint es el proxy

Dominio “útil”
Re envía al CDN basado en el valor del campo Host del header

Implante
Envía request con el campo Host del header Apuntando al CDN.



Laboratorio

Provisionar un Linux con Kali

- Colocar las herramientas requeridas en un equipo ejecutando Linux puede ser complicado
- Podemos aplicar Docker
- Tomar como base una máquina virtual Ubuntu
  - Descargar desde https://drive.google.com/file/d/1i93CKKcWmdF2BzxfXe-ru9I764PSsWU2/view?usp=sharing
- Muchos pasos no están descritos aquí, deberá conocerlos previamente (conocimientos básicos) o buscar cómo hacerlo

Provisionar un Linux con Kali

- Luego de arrancar Ubuntu, hacer login usando las siguientes credenciales
  - Usuario : vagrant
  - Password : vagrant
- Obtener la dirección IP asignada a Ubuntu
- Ingresar vía SSH desde su equipo

Provisionar un Linux con Kali

- docker run --tty --interactive kalilinux/kali-rolling /bin/bash

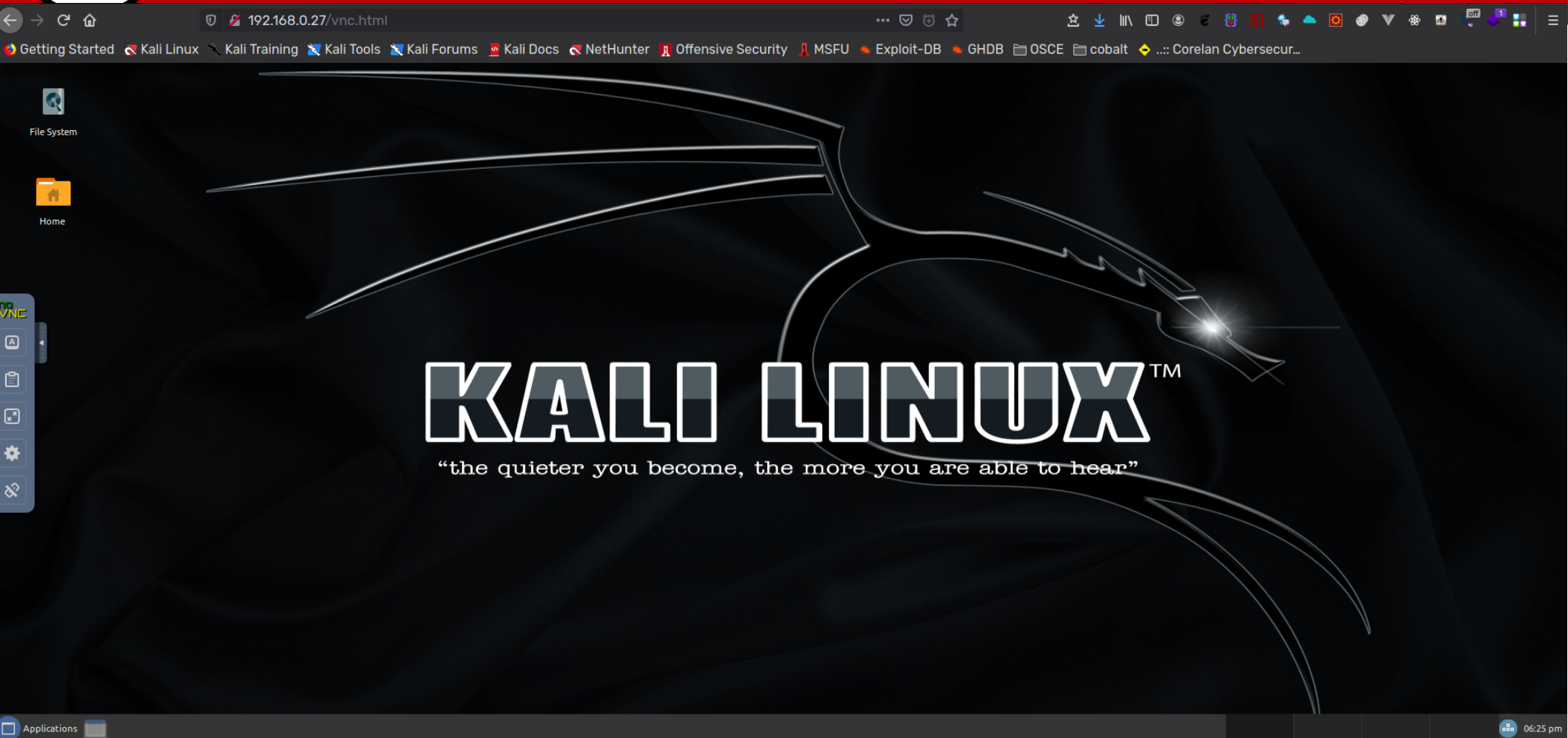
```
Unable to find image 'kalilinux/kali-rolling:latest' locally
latest: Pulling from kalilinux/kali-rolling
5d4efac9e69d: Pull complete
Digest: sha256:7568c28816d63399490a7a72c64be626d1fdaa0111bd7c0842bebe543bae6515
Status: Downloaded newer image for kalilinux/kali-rolling:latest
┌──(root㉿7200f9956f57)-[/]
└─# exit
exit
```

Provisionar un Linux con Kali

- docker run -it -p 80:6080 --cap-add=NET_ADMIN qeeqbox/kali:1.0

```
Unable to find image 'qeeqbox/kali:1.0' locally
1.0: Pulling from qeeqbox/kali
a4d7255c5bac: Pull complete
6e990e4d14ad: Pull complete
caa96ea73559: Pull complete
5e3bfc84c19d: Pull complete
f883477e9daa: Pull complete
144e7982db42: Pull complete
812c306f6afe: Pull complete
28345a4bc24f: Pull complete
2731a377382b: Pull complete
46ab8d13e39b: Pull complete
ba44c21cfa71: Pull complete
0bd1690dc3f4: Pull complete
fa40005fd91a: Pull complete
bf638493676d: Pull complete
9a3c76c41c71: Pull complete
3d077e26987c: Pull complete
3e9750955f47: Pull complete
c2b56fb0f9fc: Pull complete
d553ac96b2d5: Pull complete
cc0261e39dbd: Pull complete
Digest: sha256:a9c7287162dcd74a0654e4e1caa4e7c60523fd1fc70c0fbe39d8abc61a3a4708
Status: Downloaded newer image for qeeqbox/kali:1.0

https://github.com/qeeqbox/docker-images

Custom Kali distro accessible via VNC, RDP or web

root pass -> ^d&-MtG^T7
-----------------------
Username  -> xuser
Password  -> 8e;wbH0*v#
VNC pass  -> /xDiE!ol2u

Open-Se http://172.17.0.2:6080/index.html
```

Provisionar un Linux con Kali



Crear una imágen con Docker

- Como se ha visto, disponer de Kali usando Docker es bastante simple
- Esto no siempre cubrirá todo y a veces podemos requerir solo unas herramientas específicas
  - Por ejemplo, Impacket que está basado en Python y muchas veces resulta complicado lidiar con las versiones de este lenguaje
    - Se pueden usar “environments”, pero, vamos a ver como con Docker será más sencillo

Crear una imágen con Docker

- Imágen = Contenedor ?
- Crear un directorio Impacket en Ubuntu
- Crear un archivo llamado Dockerfile con el siguiente contenido :

```
FROM python:latest

RUN pip install --upgrade pip

RUN git clone https://github.com/SecureAuthCorp/impacket.git
RUN cd impacket && python setup.py install
ENTRYPOINT ["/bin/bash"]
```

Crear una imágen con Docker

- Crear la imágen
  - docker build -t impacket .

```
Sending build context to Docker daemon  257.5MB
Step 1/5 : FROM python:latest
latest: Pulling from library/python
1671565cc8df: Pull complete
3e94d13e55e7: Pull complete
fa9c7528c685: Pull complete
53ad072f9cd1: Pull complete
d6b983117533: Pull complete
d8092d56ded5: Pull complete
102b77c1d556: Pull complete
c65fe89c654d: Pull complete
18db97afaa16: Pull complete
Digest: sha256:5bbf8c1d6f7c0946e405587c502f316239916caba98b2dd55a31f0fd46510
Status: Downloaded newer image for python:latest
 ---> 4f9baf941f8e
Step 2/5 : RUN pip install --upgrade pip
 ---> Running in 69da9e994db0
Requirement already satisfied: pip in /usr/local/lib/python3.10/site-packages
WARNING: Running pip as the 'root' user can result in broken permissions and
virtual environment instead: https://pip.pypa.io/warnings/venv
```

```
Step 5/5 : ENTRYPOINT ["/bin/bash"]
 ---> Running in ab954a74833a
Removing intermediate container ab954a74833a
 ---> 4ad11a980aab
Successfully built 4ad11a980aab
Successfully tagged impacket:latest
```

“Ejecutar” la imágen

- docker run -it --rm impacket

```
root@b81f9738bd49:/# wmiexec.py
Impacket v0.10.1.dev1+20220720.103933.3c6713e3 - Copyright 2022 SecureAuth Corporation

usage: wmiexec.py [-h] [-share SHARE] [-nooutput] [-ts] [-silentcommand] [-debug] [-codec CODEC] [-shell-type {cmd,powershell}]
                  [-com-version MAJOR_VERSION:MINOR_VERSION] [-hashes LMHASH:NTHASH] [-no-pass] [-k] [-aesKey hex key] [-dc-ip ip address] [-A authfile]
                  [-keytab KEYTAB]
                  target [command ...]

Executes a semi-interactive shell using Windows Management Instrumentation.
```

Creando imágenes NO volátiles

- Como pueden haber comprobado anteriormente, lo hecho con Kali no es permanente, es decir, lo que se “grabó”, se perdió
- Vamos a crear un volúmen persistente

Creando imágenes NO volátiles

- git clone https://github.com/qeeqbox/docker-images

```
vagrant@vagrant:~$ git clone https://github.com/qeeqbox/docker-images
Cloning into 'docker-images'...
remote: Enumerating objects: 378, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 378 (delta 14), reused 9 (delta 9), pack-reused 360
Receiving objects: 100% (378/378), 891.49 KiB | 1.60 MiB/s, done.
Resolving deltas: 100% (88/88), done.
```

Creando imágenes NO volátiles

- git clone https://github.com/qeeqbox/docker-images

```
vagrant@vagrant:~$ git clone https://github.com/qeeqbox/docker-images
Cloning into 'docker-images'
remote: Enumerating objects
remote: Counting objects: 1
remote: Compressing objects
remote: Total 378 (delta 14
Receiving objects: 100% (37
Resolving deltas: 100% (88/
```

```
vagrant@vagrant:~$ cd docker-images/
vagrant@vagrant:~/docker-images$ ls -la
total 40
drwxrwxr-x 6 vagrant vagrant 4096 Sep  8 18:53 .
drwxr-xr-x 7 vagrant vagrant 4096 Sep  8 18:53 ..
-rw-rw-r-- 1 vagrant vagrant    9 Sep  8 18:53 changes.md
drwxrwxr-x 8 vagrant vagrant 4096 Sep  8 18:53 .git
-rw-rw-r-- 1 vagrant vagrant   79 Sep  8 18:53 info
drwxrwxr-x 4 vagrant vagrant 4096 Sep  8 18:53 kali
drwxrwxr-x 4 vagrant vagrant 4096 Sep  8 18:53 parrot
drwxrwxr-x 2 vagrant vagrant 4096 Sep  8 18:53 readme
-rw-rw-r-- 1 vagrant vagrant 4753 Sep  8 18:53 README.md
```

Creando imágenes NO volátiles

- **En el directorio creado, crear un archivo llamado docker-compose.yml con el siguiente contenido :**

```
version: "2.4"
services:
  kali_xfce:
    build: ./kali
    image: kali-html5
    container_name: kali-html5
    ports:
    - 80:6080
    environment:
    - VNC_PASSWORD=UQcyNFQ
    - ROOT_PASSWORD=UQcyNFQ
    - XUSER_PASSWORD=UQcyNFQ
    cap_add:
      - NET_ADMIN
    restart: always
    volumes:
    - ./home:/home/xuser
```

Creando imágenes NO volátiles

- Crear y ejecutar
  - docker-compose up

```
Building kali_xfce
Step 1/23 : FROM kalilinux/kali-rolling
 ---> 79fd822ddb90
Step 2/23 : MAINTAINER qeeqbox
 ---> Using cache
 ---> 97a275807833
Step 3/23 : ENV DEBIAN_FRONTEND=noninteractive
 ---> Using cache
 ---> 3d6cc3ce3032
Step 4/23 : RUN apt-get update && apt-get install -y aircrack-ng amap amass apt-utils arping arp-scan axel bash-completion binwalk bsdmainutils bulk-extractor
cewl commix crackmapexec creddump7 crunch cryptcat curl dirb dirbuster dmitry dnschef dnsenum dnsrecon dnsutils dos2unix enum4linux ethtool exiv2 expect explo
tdb fierce fping ftp gcc git gobuster golang hashcat hashdeep hashid hash-identifier hotpatch hping3 hydra iputils-ping john joomscan kpcli lbd libffi-dev mag
crescue make man-db masscan metasploit-framework mimikatz mlocate nasm nbtscan ncat ncrack netcat netcat-traditional netmask netsniff-ng net-tools ngrep nikto
nmap nodejs npm onesixtyone oscanner passing-the-hash patator php powershell powersploit proxychains proxychains4 ptunnel pwnat python2 python3 python3-pip py
hon3-setuptools python-dev python-setuptools rebind recon-ng responder ruby-dev samba samdump2 seclists set sipvicious skipfish sleuthkit smbclient smbmap smt
-user-enum snmp snmpcheck socat spike sqlmap ssh-audit sslscan sslsplit sslyze statsprocessor stunnel4 swaks tcpdump tcpick tcpreplay testssl.sh theharvester
nscmd10g tor udptunnel uniscan unix-privesc-check upx-ucl vim voiphopper wafw00f webshells weevely wfuzz wget whatweb whois windows-binaries winexe wordlists
pscan yersinia firefox-esr gosu armitage
 ---> Running in 674d438b95db
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/contrib amd64 Packages [109 kB]
Get:3 http://kali.download/kali kali-rolling/non-free amd64 Packages [221 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Packages [18.3 MB]
```

Creando imágenes NO volátiles

- Fallará

```
Package python-dev is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  python2-dev python2 python-dev-is-python3

E: Package 'netcat' has no installation candidate
E: Package 'python-dev' has no installation candidate
ERROR: Service 'kali_xfce' failed to build: The command '/bin/sh -c apt-get update && apt-get install -y aircrack-ng amap amass apt-utils arping arp-scan axel
bash-completion binwalk bsdmainutils bulk-extractor cewl commix crackmapexec creddump7 crunch cryptcat curl dirb dirbuster dmitry dnschef dnsenum dnsrecon dnsu
tils dos2unix enum4linux ethtool exiv2 expect exploitdb fierce fping ftp gcc git gobuster golang hashcat hashdeep hashid hash-identifier hotpatch hping3 hydra
iputils-ping john joomscan kpcli lbd libffi-dev magicrescue make man-db masscan metasploit-framework mimikatz mlocate nasm nbtscan ncat ncrack netcat netcat-tr
aditional netmask netsniff-ng net-tools ngrep nikto nmap nodejs npm onesixtyone oscanner passing-the-hash patator php powershell powersploit proxychains proxyc
hains4 ptunnel pwnat python2 python3 python3-pip python3-setuptools python-dev python-setuptools rebind recon-ng responder ruby-dev samba samdump2 seclists set
 sipvicious skipfish sleuthkit smbclient smbmap smtp-user-enum snmp snmpcheck socat spike sqlmap ssh-audit sslscan sslsplit sslyze statsprocessor stunnel4 swak
s tcpdump tcpick tcpreplay testssl.sh theharvester tnscmd10g tor udptunnel uniscan unix-privesc-check upx-ucl vim voiphopper wafw00f webshells weevely wfuzz wg
et whatweb whois windows-binaries winexe wordlists wpscan yersinia firefox-esr gosu armitage' returned a non-zero code: 100
```

Resolverlo es el reto de este laboratorio

Caso 5 : Mi C2 está lleno de contenedores !!!

PoshC2

Installing for Docker

You can also run PoshC2 using Docker, this allows more stable and running and enables PoshC2 to easily run on other operating systems.

The Docker install does not clone PoshC2 as the PoshC2 images on Docker Hub are used, so only a minimal install of some dependencies and scripts are performed.

To start with, install Docker on the host and then add the PoshC2 projects directory to Docker as a shared directory if required for your OS. By default this is /var/poshc2 on *nix and /private/var/poshc2 on Mac.

Kali based hosts

Install script:

```
*** PoshC2 Install script for Docker ***
Usage:
./Install-for-Docker.sh -b <git branch>

Default is the master branch
```

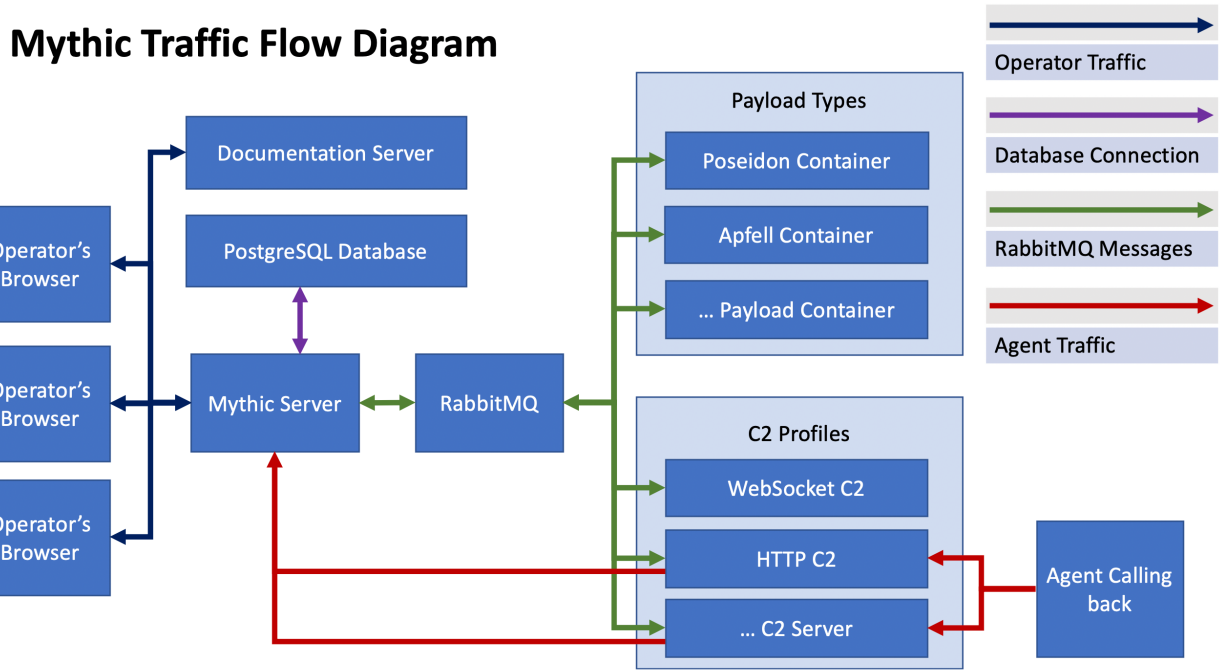
Elevated privileges are required as the install script performs script installations.

```
curl -sSL https://raw.githubusercontent.com/nettitude/PoshC2/master/Install-for-Docker.sh | sudo bas
```

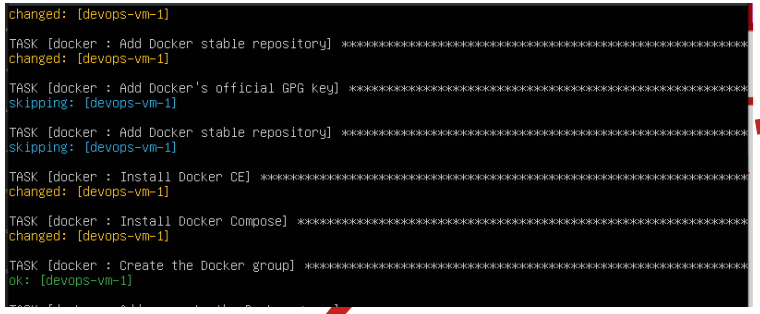
To use the `dev` or feature branches with Docker curl down the `Install-for-Docker.sh` on the appropriate branch and pass the branch name as an argument:

```
curl -sSL https://raw.githubusercontent.com/nettitude/PoshC2/BRANCHNAME/Install-for-Docker.sh | sudo
```

Caso 5 : Mi C2 está lleno de contenedores !!!

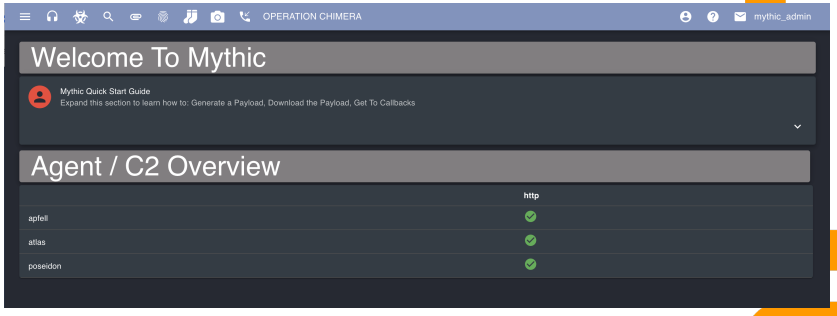


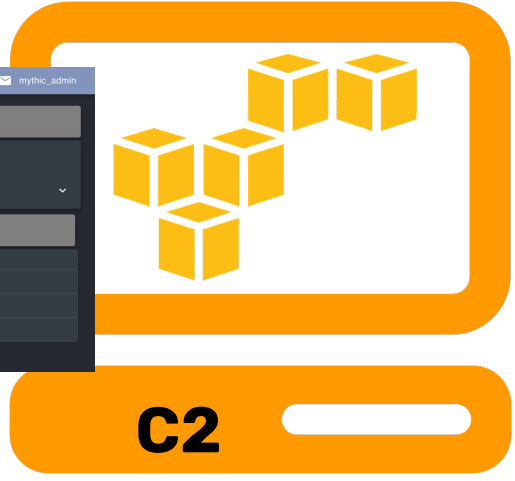
Caso 5 : Mi C2 está lleno de contenedores !!!



SSH

Ansible





C2