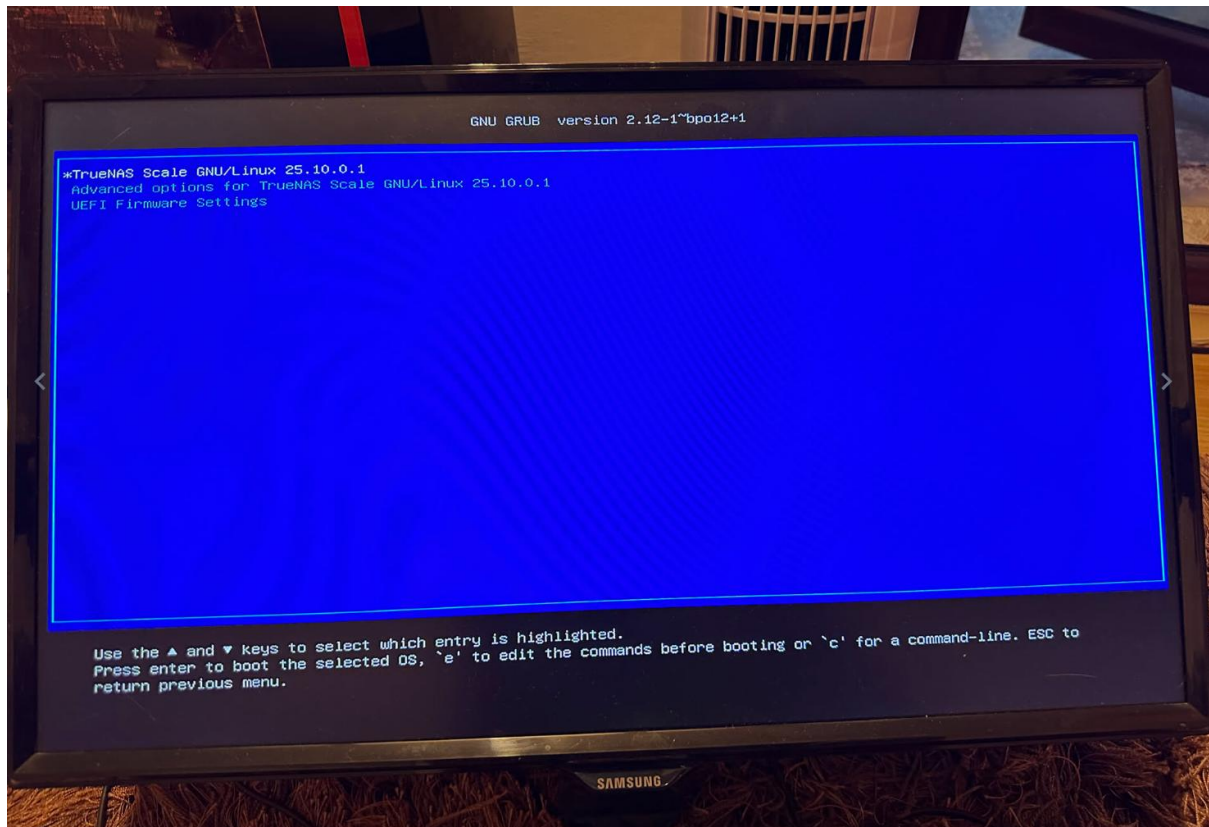
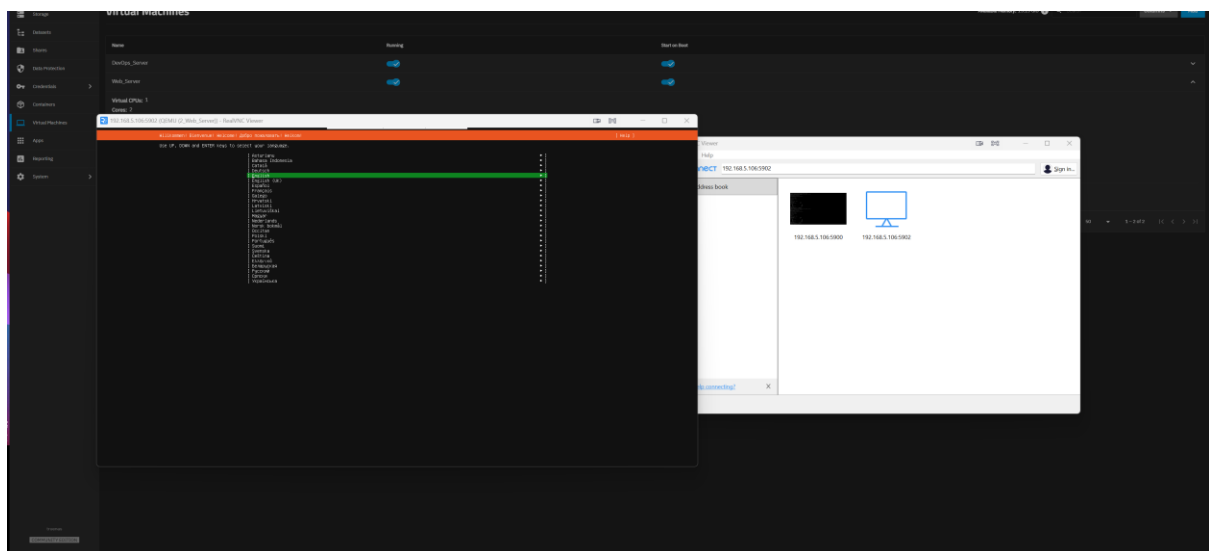
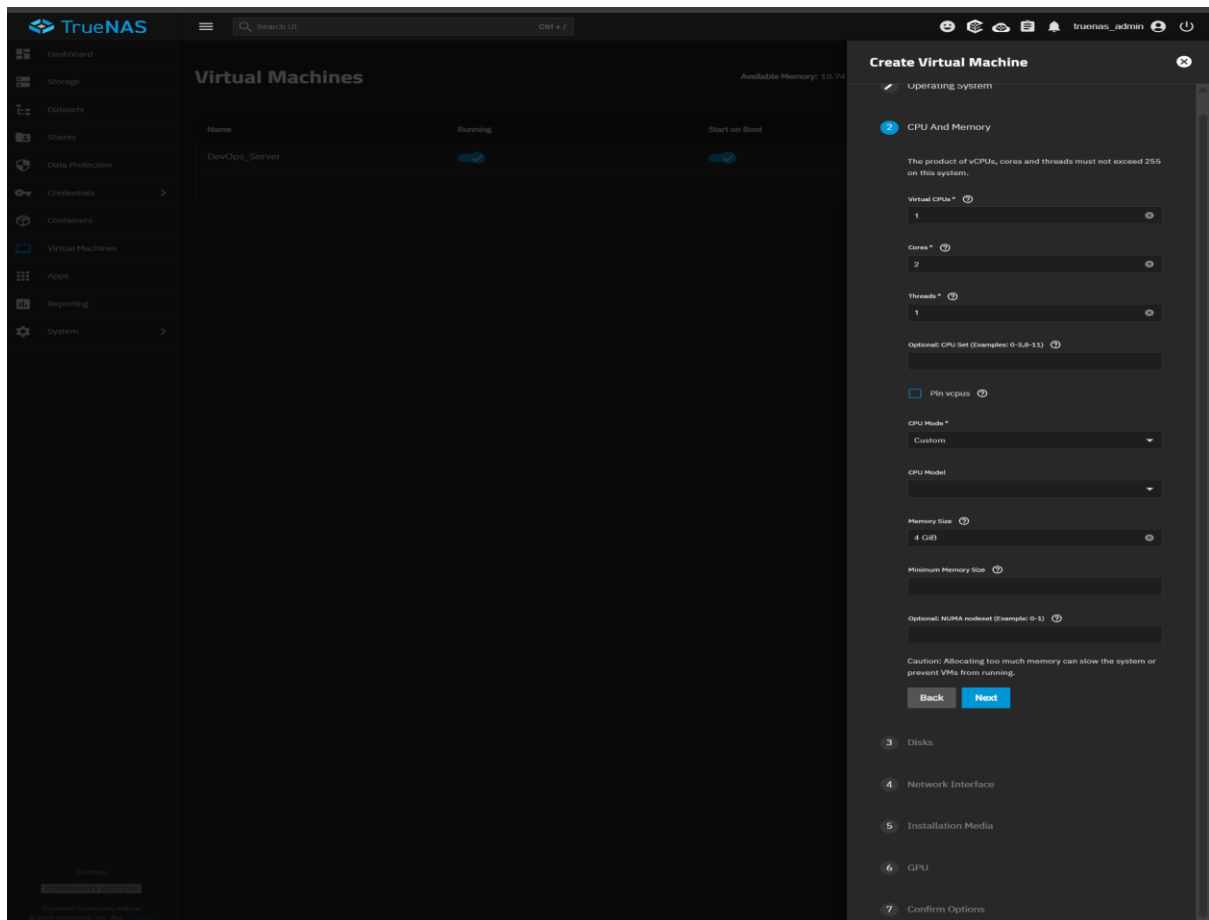


Pentru acest proiect de laborator am optat sa folosesc propriul VPS(un VM facut pe serverul meu propriu de TRUENAS SCALE) chiar daca ne era furnizat unul deja





Setupul pentru VM-ul de webserver ^^^

Pentru dev/devops stuff am facut alt VM separat ca sa fie ordine (si best practice din ce am citit)

Dupa logare:

```
WebServer (QEMU (2_Web_Server)) - RealVNC Viewer

0% 0.00 tasks: 27, 29 thr: 100 kth: 1 running
0.03 load average: 0.15 0.07 0.01
210M/3.31G uptime: 00:05:38
00/3.01G

Main [ 100%]
PID USER PR1 NI VIRT RES SHR S CPU/MEM% TIME+ Command
1 root 20 0 0 14920 16366 11792 S 0.0 0.5 0:01.00 /usr/lib/systemd/systemd --switched-root --system --deserialize=51
117 root 19 0 56392 11192 15688 S 0.0 0.5 0:00.11 /usr/lib/systemd/systemd-journald
840 root 11 0 272K 3644 7540 S 0.0 0.3 0:00.02 /usr/sbin/multipathd -d -s
860 root 20 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
861 root 11 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
862 root 11 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
863 root 11 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
864 root 11 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
865 root 11 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
866 root 11 0 272K 3644 0 S 0.0 0.3 0:00.00 /usr/sbin/multipathd -d -s
871 systemd-res 20 0 2740 14804 12100 S 0.0 0.4 0:00.04 /usr/lib/systemd/systemd-resolved
878 root 20 0 3460 13536 9284 S 0.0 0.4 0:00.16 /usr/lib/systemd/systemd-udevd
1268 systemd-res 20 0 21432 12176 10624 S 0.0 0.4 0:00.04 /usr/lib/systemd/systemd-networkd
1305 root 20 0 2080 2012 1092 S 0.0 0.1 0:00.01 /bin/sh /usr/lib/systemd/scripts/chronyd-starters.sh -n -F 1
1307 root 20 0 4472 2152 2916 S 0.0 0.1 0:00.00 /usr/sbin/cron -F
1308 systemd-logd 20 0 1132 5076 4000 S 0.0 0.2 0:00.06 /sbin/dmccn --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
1318 root 20 0 44092 23056 14696 S 0.0 0.3 0:00.00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
1320 polkitd 20 0 2396 9452 7640 S 0.0 0.2 0:00.03 /usr/lib/polkit-1/polkitd --no-debug --log-level=notice
1326 root 20 0 10620 9904 8576 S 0.0 0.3 0:00.04 /usr/lib/systemd/systemd-logind
1329 root 20 0 5308 14984 12612 S 0.0 0.4 0:00.03 /usr/libexec/udisks2/udisksd
1332 root 20 0 5308 14984 0 S 0.0 0.4 0:00.00 /usr/libexec/udisks2/udisksd
1336 root 20 0 5308 14984 0 S 0.0 0.4 0:00.00 /usr/libexec/udisks2/udisksd
1340 root 20 0 5308 14984 0 S 0.0 0.4 0:00.00 /usr/libexec/udisks2/udisksd
1347 root 20 0 8760 5612 4744 S 0.0 0.2 0:00.02 login -- bogdan
1423 chrony 20 0 26644 12000 7044 S 0.0 0.3 0:00.23 /usr/sbin/chronyd -n -F 1
1445 root 20 0 1306 3144 10020 S 0.0 0.2 0:00.07 /usr/bin/python3 /usr/share/anaconda/updates/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
1437 polkitd 20 0 2396 9452 0 S 0.0 0.2 0:00.00 /usr/lib/polkit-1/polkitd --no-debug --log-level=notice
1438 polkitd 20 0 2396 9452 0 S 0.0 0.2 0:00.00 /usr/lib/polkit-1/polkitd --no-debug --log-level=notice
1439 polkitd 20 0 2396 9452 0 S 0.0 0.2 0:00.00 /usr/lib/polkit-1/polkitd --no-debug --log-level=notice
1452 chrony 20 0 10996 2456 1012 S 0.0 0.1 0:00.00 /usr/sbin/chronyd -n -F 1
1455 syslog 20 0 2196 6032 4208 S 0.0 0.2 0:00.01 /usr/sbin/rsyslogd -n -iNONE
1471 root 20 0 3028 14216 11976 S 0.0 0.4 0:00.05 /usr/sbin/ModemManager
1470 root 20 0 5308 14984 0 S 0.0 0.4 0:00.00 /usr/libexec/udisks2/udisksd
1473 root 20 0 5308 14984 0 S 0.0 0.4 0:00.00 /usr/libexec/udisks2/udisksd
1502 syslog 20 0 2196 6032 0 S 0.0 0.2 0:00.00 /usr/sbin/rsyslogd -n -iNONE
1503 syslog 20 0 2196 6032 0 S 0.0 0.2 0:00.00 /usr/sbin/rsyslogd -n -iNONE
1504 syslog 20 0 2196 6032 0 S 0.0 0.2 0:00.00 /usr/sbin/rsyslogd -n -iNONE
1513 root 20 0 3028 14216 0 S 0.0 0.4 0:00.00 /usr/sbin/ModemManager
1520 root 20 0 3028 14216 0 S 0.0 0.4 0:00.00 /usr/sbin/ModemManager
1525 root 20 0 5308 14984 0 S 0.0 0.4 0:00.00 /usr/libexec/udisks2/udisksd
1527 root 20 0 3028 14216 0 S 0.0 0.4 0:00.00 /usr/sbin/ModemManager
1531 root 20 0 1306 35544 0 S 0.0 0.5 0:00.00 /usr/bin/python3 /usr/share/anaconda/updates/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
1567 root 20 0 10740 7020 5656 S 0.0 0.2 0:00.00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
1640 root 20 0 4776 40144 39008 S 0.1 0.9 0:00.99 /usr/libexec/ftpd/ftpd
1647 root 20 0 4776 40144 0 S 0.0 1.4 0:00.01 /usr/libexec/ftpd/ftpd
1648 root 20 0 4776 40144 0 S 0.0 1.4 0:00.00 /usr/libexec/ftpd/ftpd
1649 root 20 0 4776 40144 0 S 0.0 1.4 0:00.00 /usr/libexec/ftpd/ftpd
1651 root 20 0 3116 10996 5268 S 0.0 0.3 0:00.02 /usr/libexec/ufw/ufw
1654 root 20 0 3116 10996 0 S 0.0 0.3 0:00.00 /usr/libexec/ufw/ufw
1655 root 20 0 3116 10996 0 S 0.0 0.3 0:00.00 /usr/libexec/ufw/ufw
1656 root 20 0 3116 10996 0 S 0.0 0.3 0:00.00 /usr/libexec/ufw/ufw
1702 bogdan 20 0 20020 12796 10640 S 0.0 0.4 0:00.06 /usr/lib/systemd/systemd --user
1706 bogdan 20 0 20004 4092 2076 S 0.0 0.1 0:00.00 (sd-pam)
1805 bogdan 20 0 0056 6180 4500 S 0.0 0.2 0:00.04 -bash
1951 bogdan 20 0 9526 6256 4004 T 0.0 0.2 0:00.05 http
1974 bogdan 20 0 9560 6256 4288 R 0.0 0.2 0:00.01 http
```

m-am conectat apoi prin ssh ca sa nu mai am inca o aplicatia in plus pe pc

Name	Date modified	Type	Size
 bogdandevops	26.05.2026 23:29	File	1 KB
 bogdandevops.pub	26.05.2026 23:29	PUB File	1 KB

```
Windows PowerShell
PS C:\Users\ripi> ssh -i .\bogdandevops bogdan@192.168.5.110
```

```
bogdan@192.168.5.110's password:
Welcome to Ubuntu 26.04 LTS (GNU/Linux 7.0.0-15-generic x86_64)

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

System information as of Tue May 26 08:41:49 PM UTC 2026

System load:  0.0      Processes:    170
Usage of /:   14.2% of 55.75GB   Users logged in:  0
Memory usage: 2%      IPv4 address for ens3: 192.168.5.110
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue May 26 20:38:49 2026 from 192.168.5.10
bogdan@bogdan:~$
```

Am instalat nexus repository manager pe VM

sonatype nexus repository
Community Edition

Welcome

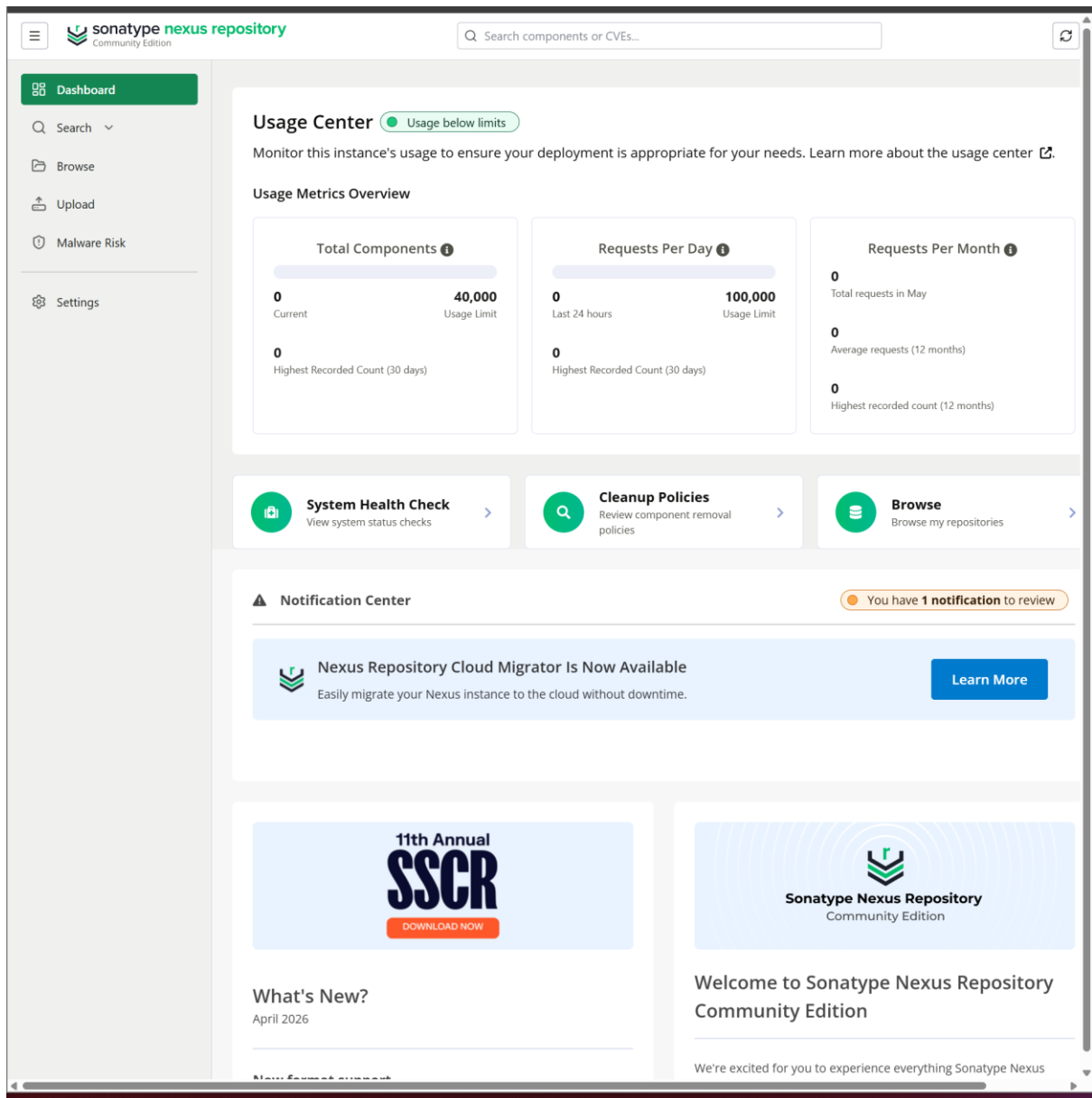
Login to continue

To log in for the first time, use the generated admin password located at:
/home/nexus/sonatype-work/nexus3/admin.password

Username

Password

```
bogdan@bogdan: ~  
Proto Recv-Q Send-Q Local Address      Foreign Address    State       PID/Program name  
tcp        0      0 0.0.0.0:22         0.0.0.0:*          LISTEN      -  
tcp        0      0 127.0.0.53:53      0.0.0.0:*          LISTEN      -  
tcp        0      0 127.0.0.54:53      0.0.0.0:*          LISTEN      -  
tcp        0      0 192.168.5.110:22   192.168.5.10:55180 ESTABLISHED -  
tcp6       0      0 :::22              :::*                LISTEN      -  
tcp6       0      0 :::8081            :::*                LISTEN      -  
udp        0      0 127.0.0.54:53      0.0.0.0:*          -           -  
udp        0      0 127.0.0.53:53      0.0.0.0:*          -           -  
udp        0      0 192.168.5.110:68   0.0.0.0:*          -           -  
udp        0      0 127.0.0.1:323      0.0.0.0:*          -           -  
udp6       0      0 :::1:323           :::*                -           -  
udp6       960    0 fe80::2a0:98ff:fe4c:546 :::*                -           -  
bogdan@bogdan:~$
```



Am facut si cont pe nexus, urmatorul pas acum este sa transformam nexusul intr un docker registry privat, adica ce se builduieste cu github actions sa se stockeze imaginile de docker (backend si frontend) in el ca mai apoi sa fie "servite" pe celalalt VM de web

Name: A unique identifier for this repository

Online: ☒ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our [documentation](#) for which connector is appropriate for your use case. For information on scaling the repositories see our [scaling documentation](#).

☐ **Path based routing**

Use the following URL with your Docker client to pull images from this repository:
`company.com/repo-name/image:tag`

☒ **Other Connectors**

Select subdomain, ports, or leave blank if using a reverse proxy.

HTTP:

Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.

☒

HTTPS:

Create an HTTPS connector at specified port. Normally used if the server is configured for https.

☐

Allow Anonymous Docker Pulls for This Repository:

☒ Allow anonymous Docker pulls for this repository (Global Anonymous Access and Docker Bearer Token Realm required)

i Anonymous access to Docker repositories requires configuration in two places: globally on the Security → Anonymous Access page and within each Docker repository's configuration form. [Learn more in our Docker help documentation](#).

Docker Registry API Support

Enable Docker V1 API:

☐ Allow clients to use the V1 API to interact with this repository

Storage

Blob store:

Blob store used to store repository contents

Strict Content Type Validation:

☒ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

Deployment policy:

Controls if deployments of and updates to artifacts are allowed

Aici: 8082 portul prin care comunica docker cu nexus, ca nexus e pe 8081, si allow anonymous docker pull ca sa poata sa traga web serveru imaginile fara sa faca login

Realms

Manage the active security realms and their order

Active Realms

Available

Filter

+ Transfer All

+ Conan Bearer Token Realm

+ Default Role Realm

+ LDAP Realm

+ npm Bearer Token Realm

+ NuGet API-Key Realm

+ Pub Bearer Token Realm

+ Rut Auth Realm

+ Terraform Token Realm

8 items available

Active

Filter

x Remove All

x Local Authenticating R... ↑ ↓

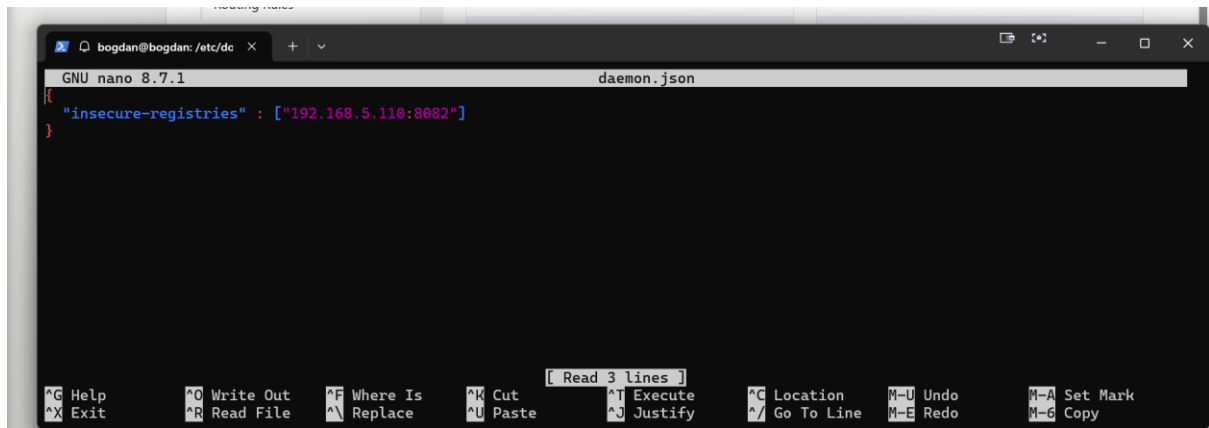
x Docker Bearer Token R... ↑ ↓

2 items transferred

Discard

Save

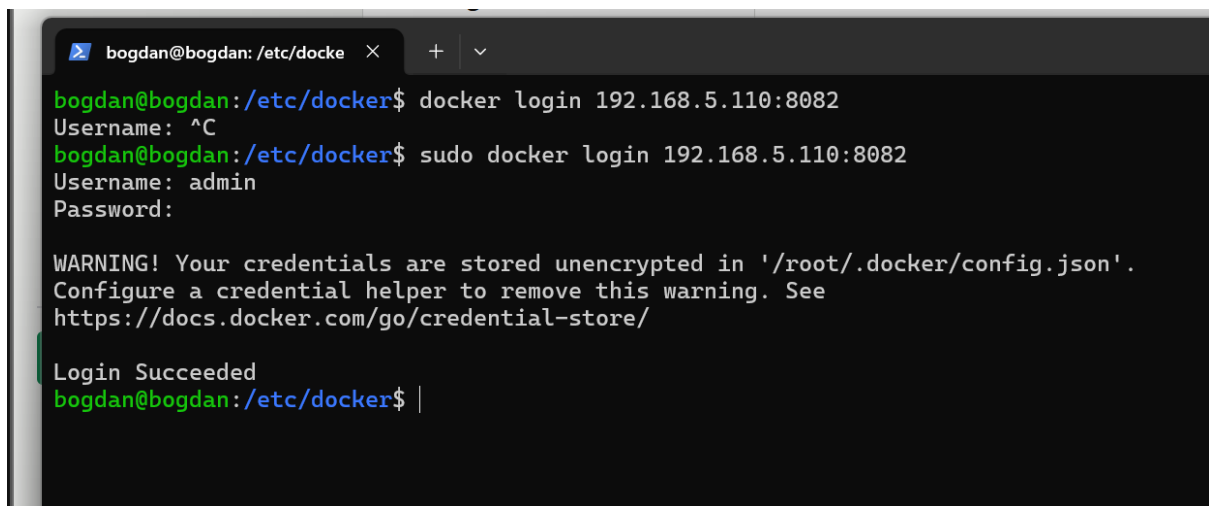
Pt ca docker push si login sa functioneze corect, nexus trb sa stie cum sa valideze credidientialele docker



```
GNU nano 8.7.1 daemon.json
{"insecure-registries": ["192.168.5.110:8082"]}

[ Read 3 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File   ^\ Replace    ^U Paste      ^J Justify
^C Location   ^-_ Undo      ^-A Set Mark
^_/ Go To Line ^-E Redo      ^-6 Copy
```

ca sa stie unde sa se conecteze prin http la nexus(ideea e ca docker de baza refuza conexiunile nesecurizate)

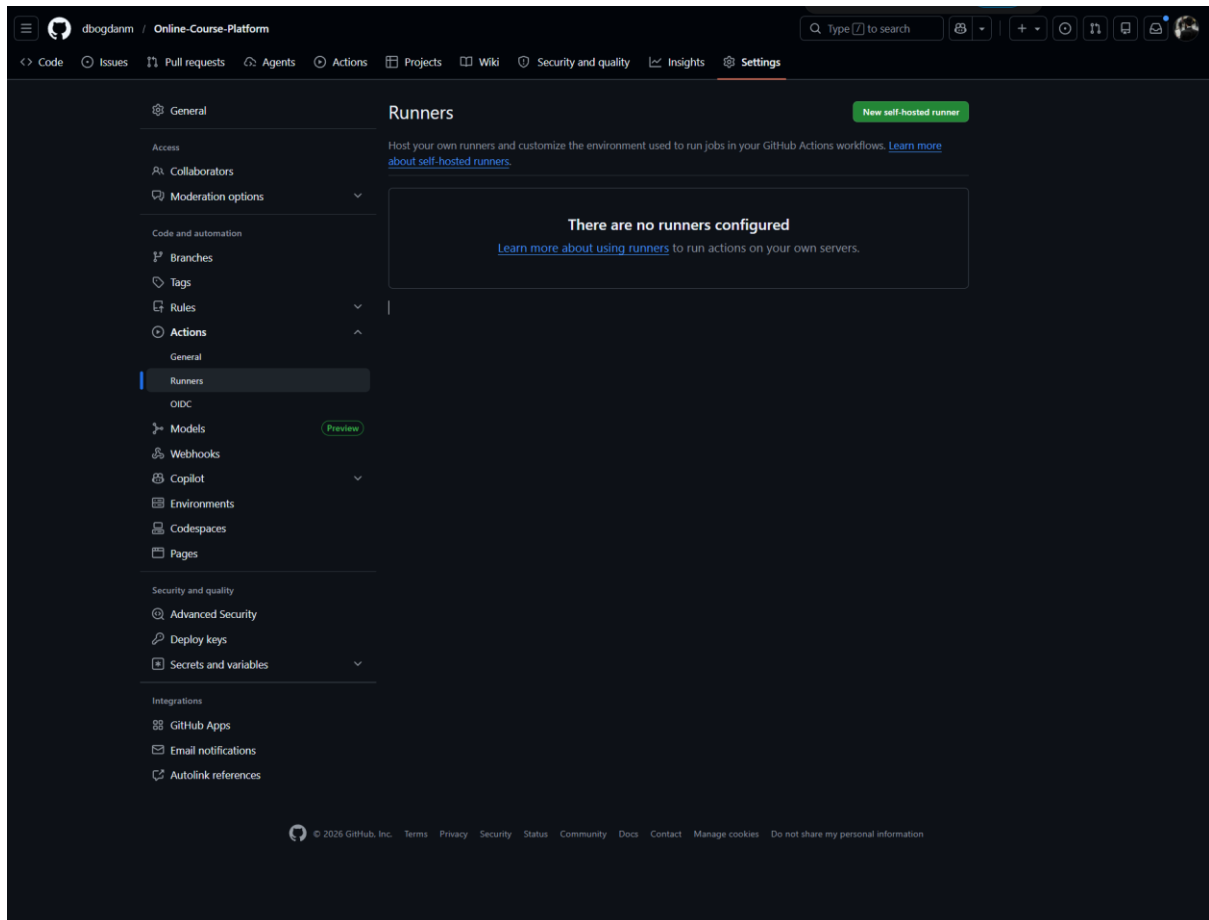


```
bogdan@bogdan: /etc/docke
bogdan@bogdan:/etc/docker$ docker login 192.168.5.110:8082
Username: ^C
bogdan@bogdan:/etc/docker$ sudo docker login 192.168.5.110:8082
Username: admin
Password:

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
bogdan@bogdan:/etc/docker$
```


Acum trebuie sa instalam agentul de github(github runner, self hosted normal)



Am facut si alt user (din motive de securitate in mare parte aici), aici ne ghideaza github singur:

Warning

Using self-hosted runners in public repositories is not recommended. Forks of your public repository can potentially run dangerous code on your self-hosted runner by creating a pull request. [Learn more about security hardening for self-hosted runners.](#)

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. If you do not already have an existing volume licensing agreement for your GitHub purchases, by downloading and configuring the GitHub Actions Runner, you agree to the [GitHub Customer Agreement](#).

Runner image

macOS

Linux

Windows

Architecture

x64

Download

```
# Create a folder
$ mkdir actions-runner && cd actions-runner

# Download the latest runner package
$ curl -o actions-runner-linux-x64-2.334.0.tar.gz -L
https://github.com/actions/runner/releases/download/v2.334.0/actions-runner-linux-x64-2.334.0.tar.gz

# Optional: Validate the hash
$ echo "actions-runner-linux-x64-2.334.0.tar.gz" | shasum -a 256 -c

# Extract the installer
$ tar xzf ./actions-runner-linux-x64-2.334.0.tar.gz
```

Configure

```
# Create the runner and start the configuration experience
$ ./config.sh --url https://github.com/dbogdanm/Online-Course-Platform --token R...

# Last step, run it!
$ ./run.sh
```

Using your self-hosted runner

```
# Use this YAML in your workflow file for each job
runs-on: self-hosted
```

For additional details about configuring, running, or shutting down the runner, please check out our [product docs](#).

```
github-runner@bogdan:~$ cat /etc/passwd | grep github-runner
github-runner:x:1000:1000:github-runner:/home/github-runner:/bin/bash
github-runner$ echo "0" | shasum -a 256 -c
actions-runner-linux-x64-2.334.0.tar.gz: OK
github-runner$ tar xzf ./actions-runner-linux-x64-2.334.0.tar.gz
github-runner$ ./config.sh --url https://github.com/dbogdanm/Online-Course-Platform --token:

-----
      G I T H U B   A C T I O N S
-----
                Self-hosted runner registration
-----

# Authentication

✓ Connected to GitHub

# Runner Registration

Enter the name of the runner group to add this runner to: [press Enter for Default]
```

Dupa ce am terminat tutorialul de la github, trebuie ca runnerul sa fie setat sa mearga in permanenta, nu doar o singura data (cand zice la final ca dam ./run.sh simplu doar), deci il facem ca serviciu

```
Enter name of work folder: [press Enter for _work]
/ Settings Saved.

github-runner@bogdan:~/actions-runner$ su bogdan
Password:
su: Authentication failure
github-runner@bogdan:~/actions-runner$ su bogdan
Password:
bogdan@bogdan:/home/github-runner/actions-runner$ sudo ./svc.sh install github-runner
sudo ./svc.sh start
Creating launch runner in /etc/systemd/system/actions.runner.dbogdanm-Online-Course-Platform.devops-server.service
Run as user: github-runner
Run as uid: 1002
gid: 1002
Created symlink '/etc/systemd/system/multi-user.target.wants/actions.runner.dbogdanm-Online-Course-Platform.devops-server.service' + '/etc/systemd/s
ystem/actions.runner.dbogdanm-Online-Course-Platform.devops-server.service'.

/etc/systemd/system/actions.runner.dbogdanm-Online-Course-Platform.devops-server.service
● actions.runner.dbogdanm-Online-Course-Platform.devops-server.service - GitHub Actions Runner (dbogdanm-Online-Course-Platform.devops-server)
   Loaded: loaded (/etc/systemd/system/actions.runner.dbogdanm-Online-Course-Platform.devops-server.service; enabled; preset: enabled)
   Active: active (running) since Tue 2026-05-26 21:54:40 UTC; 7ms ago
 Invocation: 9e25398da853431e90e63c0665b82c6b
   Main PID: 8226 ((runsv, sh))
    Tasks: 1 (Limit: 15215)
  Memory: 1.7M (peak: 1.7M)
     CPU: 2ms
   CGroup: /system.slice/actions.runner.dbogdanm-Online-Course-Platform.devops-server.service
           └─8226 "(runsv, sh)"

May 26 21:54:40 bogdan systemd[1]: Started actions.runner.dbogdanm-Online-Course-Platform.devops-server.service - GitHub...-server).
Hint: Some lines were ellipsized, use -l to show in full.
bogdan@bogdan:/home/github-runner/actions-runner$ |
```

Abia acum e pregatit VM-ul de devops, ca rezumat: instalare de docker, nexus cu repository privat si runner pe githubs

Pregatirea VM-ului de web(initial configurariile prin ssh, instalari de docker si configurarea de insecure registry, la fel ca mai inainte)

```
bogdan@bogdanweb:~/ssh$ sudo apt install netstat
[sudo: authentication] Password:
Error: Unable to locate package netstat
bogdan@bogdanweb:~/ssh$ netstat
Command 'netstat' not found, but can be installed with:
sudo apt install net-tools
bogdan@bogdanweb:~/ssh$ sudo apt install net-tools
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14  linux-headers-7.0.0-14-generic  linux-image-unsigned-7.0.0-14-generic  linux-main-modules-7.0.0-14-generic  linux-modules-7.0.0-14-generic  linux-tools-7.0.0-14  linux-tools-7.0.0-14-generic
Use 'sudo apt autoremove' to remove them.

Installing:
  net-tools

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 4
  Download size: 151 kB
  Space needed: 680 kB / 29.2 GB available

Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 2.10-2ubuntu1 [151 kB]
Fetched 151 kB in 0s (1,290 kB/s)
Selecting previously unselected package net-tools.
(Reading database... 139635 files and directories currently installed.)
Preparing to unpack ./net-tools_2.10-2ubuntu1_amd64.deb...
Unpacking net-tools (2.10-2ubuntu1)...
Setting up net-tools (2.10-2ubuntu1)...
Processing triggers for man-db (2.13.1-1build1)...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
bogdan@bogdanweb:~/ssh$ ifconfig
bogdan@bogdanweb:~/ssh$ ifconfig -a
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.5.111 netmask 255.255.255.0  broadcast 192.168.5.255
    inet6 fe80::2a0:9eff:fe53:72f0 prefixlen 64  scopeid 0x20<link>
    ether 08:00:26:53:72:f0  txqueuelen 1000  (Ethernet)
    RX packets 394  bytes 218926 (21.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 785  bytes 78600 (76.6 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128  scopeid 0x10<host>
    loop  txqueuelen 1000  (local loopback)
    RX packets 125  bytes 12343 (12.3 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 125  bytes 12343 (12.3 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

bogdan@bogdanweb:~/ssh$
```

```
bogdan@bogdanweb: ~
| . B * = B + = . o |
| * = O O * . o |
| + + O + o . = . |
| o = . o . o |
+----[SHA256]-----+

ripip@Bogdan-PC MINGW64 ~/.ssh
$ ssh bogdan@192.168.5.111
The authenticity of host '192.168.5.111 (192.168.5.111)' can't be established.
ED25519 key fingerprint is SHA256:Yl3h5SVA8l8fonCMEOf1MaLMwF0wtLoQLmBTP/uVRKI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.5.111' (ED25519) to the list of known hosts.
bogdan@192.168.5.111's password:
Welcome to Ubuntu 26.04 LTS (GNU/Linux 7.0.0-15-generic x86_64)

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

System information as of Tue May 26 10:05:16 PM UTC 2026

System load:  0.09          Processes:      136
Usage of /:   19.4% of 36.07GB  Users logged in: 0
Memory usage: 8%           IPv4 address for ens3: 192.168.5.111
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

bogdan@bogdanweb:~$
```

```
bogdan@bogdanweb: ~/.ssh x + v
containerd:
  Version:      v2.2.4
  GitCommit:    193637f77ee8ae5f5aa5248f49e7baa3e6164966e
runc:
  Version:      1.3.5
  GitCommit:    v1.3.5-0-g488fc13e
docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0

=====

To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:

    dockerd-rootless-setuptool.sh install

Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/

WARNING: Access to the remote API on a privileged Docker daemon is equivalent
to root access on the host. Refer to the 'Docker daemon attack surface'
documentation for details: https://docs.docker.com/go/attack-surface/

=====
```

La fel ca mai inainte:

```
bogdan@bogdanweb: /etc/di x + v
GNU nano 8.7.1 daemon.json
{
  "insecure-registries" : ["192.168.5.110:8082"]
}
```

Si iar setarile sa meraga ca serviciu...

```

+-----[SHA256]-----+
github-runner@bogdan:~$ ssh-copy-id bogdan@192.168.5.111
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/github-runner/.ssh/id_ed25519.pub"
The authenticity of host '192.168.5.111 (192.168.5.111)' can't be established.
ED25519 key fingerprint is: SHA256:Yl3h5SVA8l8fonCME0f1MaLMWF0wtLoQLmBTP/uvRKI
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
bogdan@192.168.5.111's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'bogdan@192.168.5.111'"
and check to make sure that only the key(s) you wanted were added.

github-runner@bogdan:~$ ssh bogdan@192.168.5.111
Welcome to Ubuntu 26.04 LTS (GNU/Linux 7.0.0-15-generic x86_64)

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

System information as of Tue May 26 10:18:45 PM UTC 2026

System load:  0.0               Processes:    142
Usage of /:   20.8% of 36.07GB  Users logged in: 0
Memory usage: 10%              IPv4 address for ens3: 192.168.5.111
Swap usage:  0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

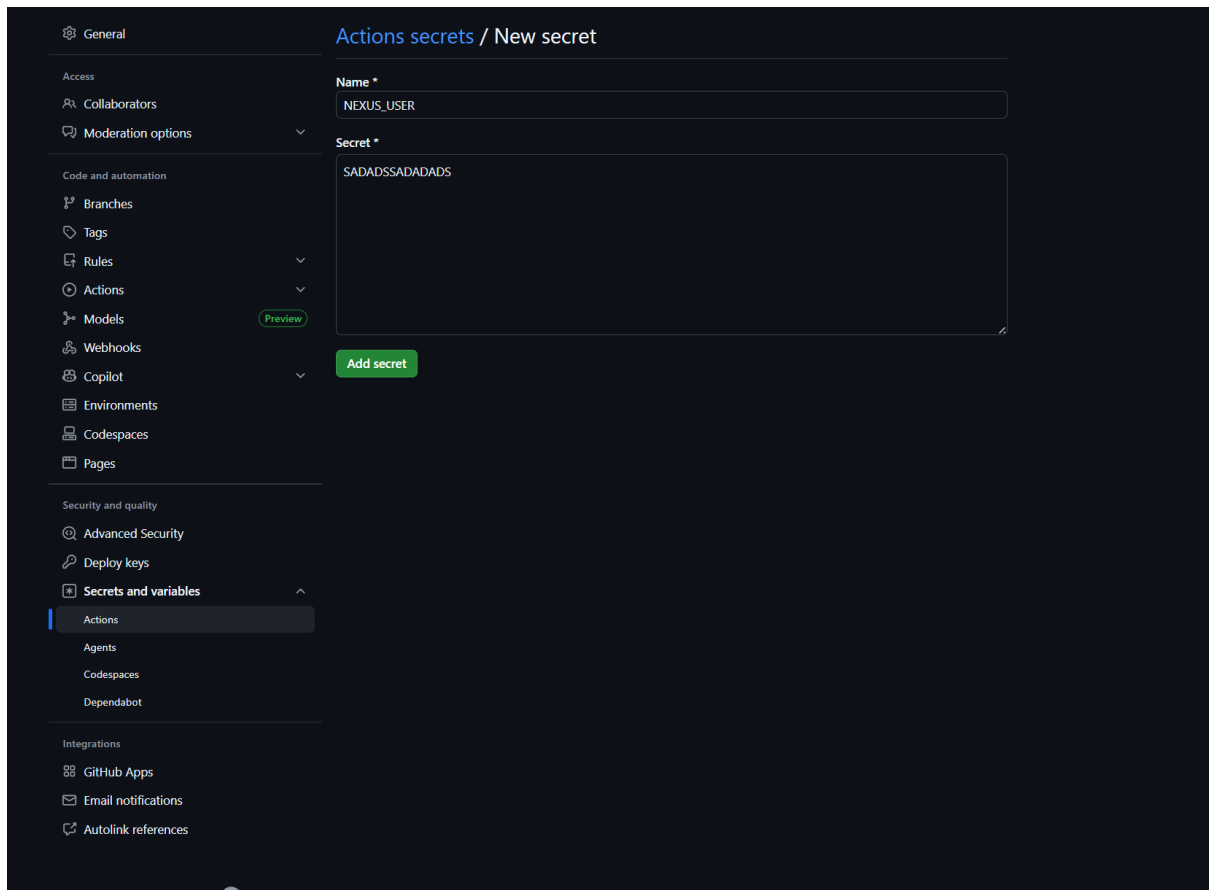
5 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

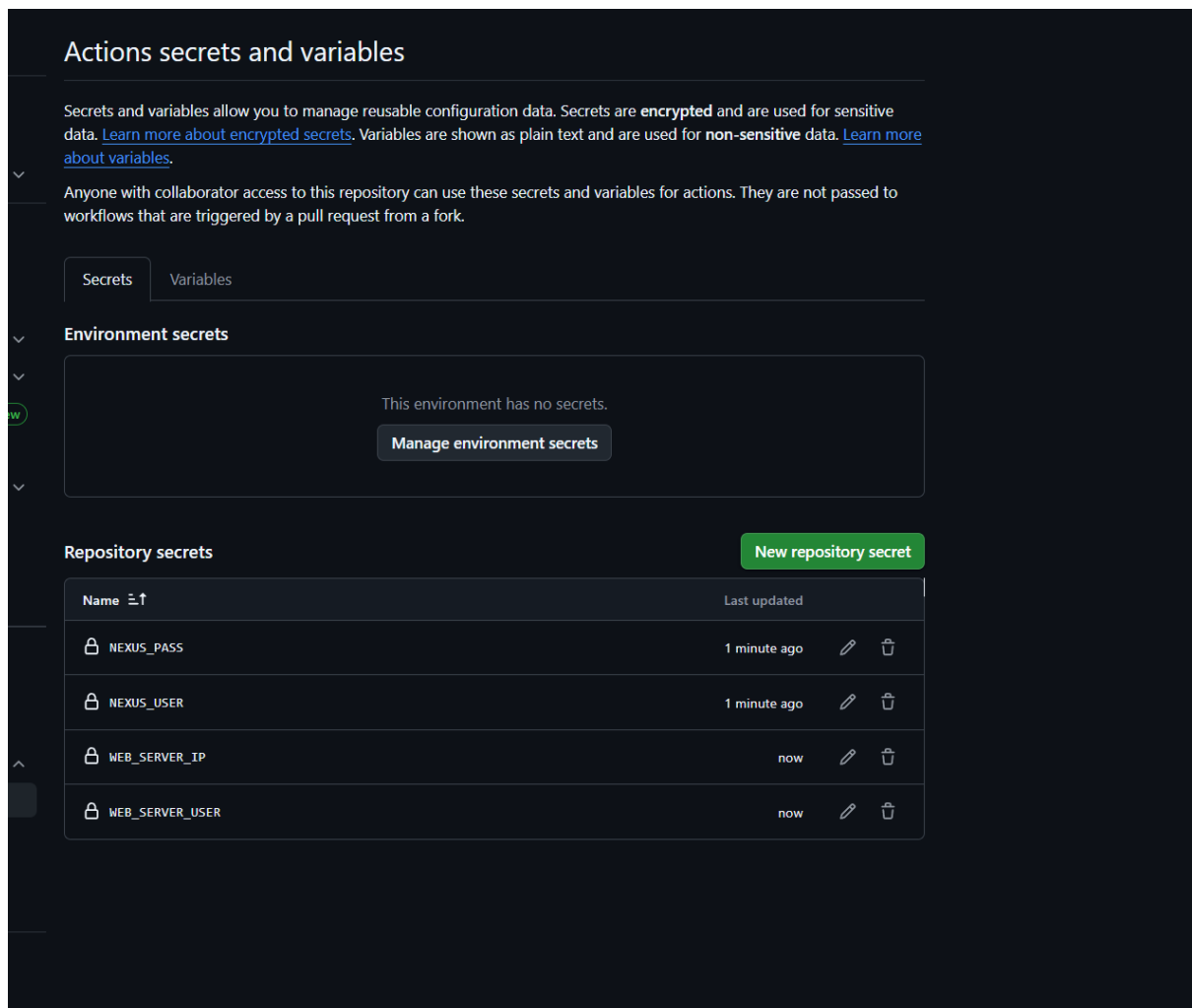
Last login: Tue May 26 22:07:51 2026 from 192.168.5.10
bogdan@bogdanweb:~$ |

```

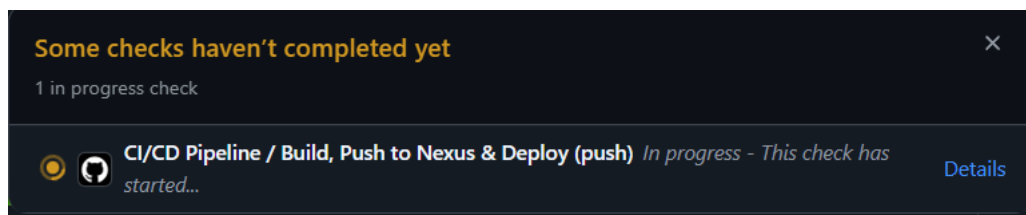
Aici am pus cheia ssh ca sa aiba acces VM-urile intre ele, adica din cel de devops sa pot da ssh in cel de web



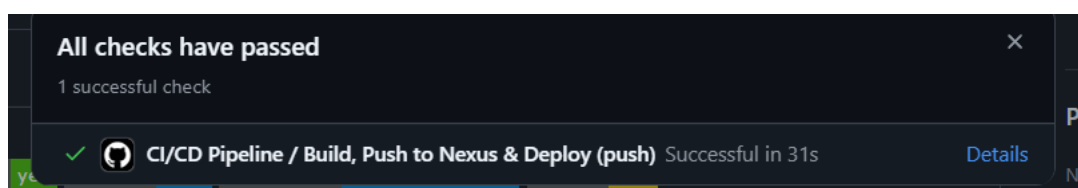
Acum vine si partea interesanta, parerea mea, ca sa facem pipelineul functional, trebuie sa mai facem cv modificari la proiect, adica sa nu lasam parole la vedere



Partea mai grea acum este scrierea fisierelor deploy.yml si docker-compose.prod.yml (nu am facut eu acest lucru singur)



Si pushul final



← CI/CD Pipeline

✓ ci: fix docker-compose command syntax in deploy step #3 Re-run all jobs ...

Summary

All jobs

Build, Push to Nexus & Deploy

Run details

Usage

Workflow file

Annotations

1 warning

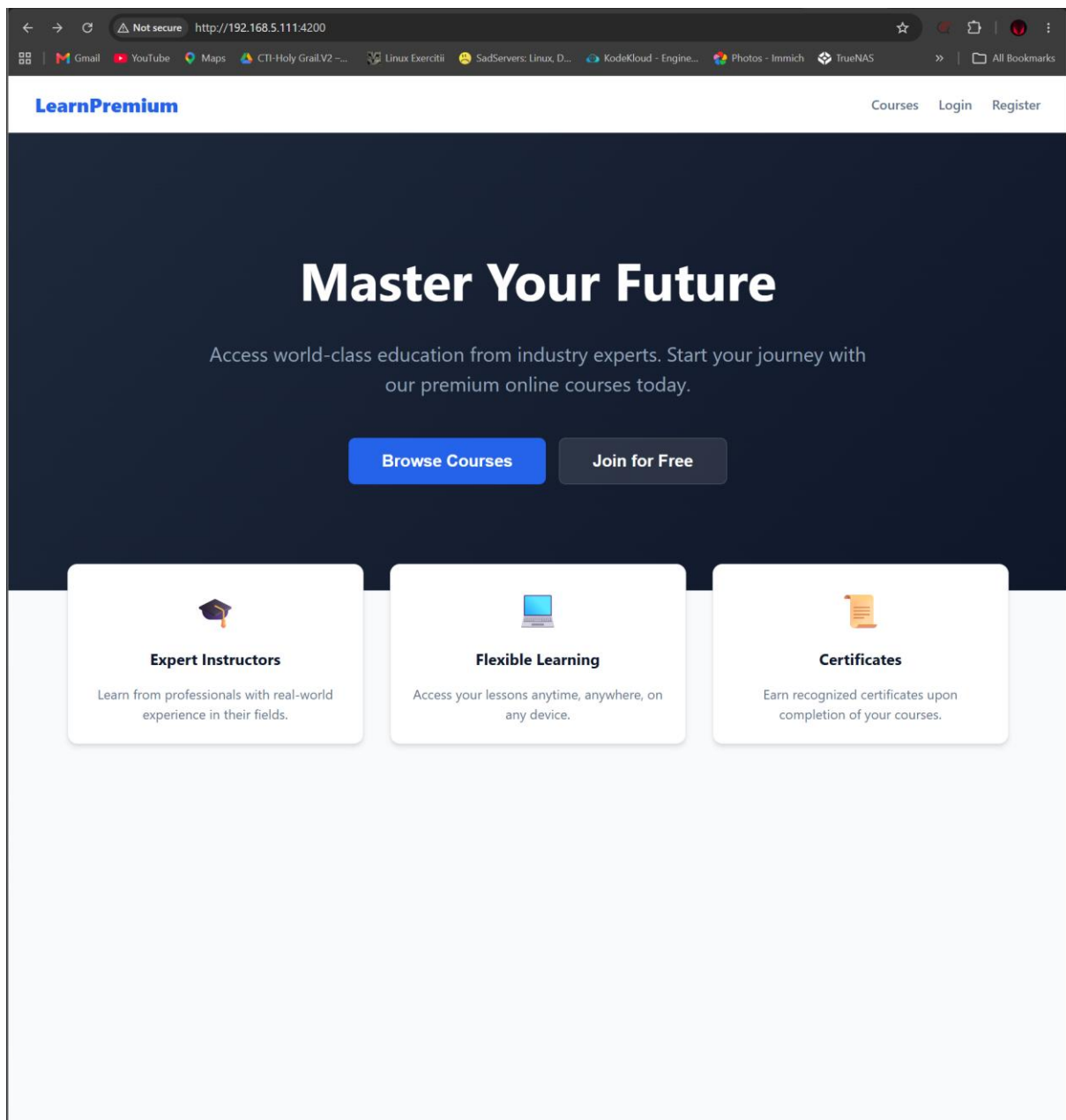
Build, Push to Nexus & Deploy

succeeded now in 31s

Search logs

- > ✓ Set up job 1s
- > ✓ Checkout Code 1s
- > ✓ Login to Nexus Docker Registry 0s
- > ✓ Build and Push Backend Image 12s
- > ✓ Build and Push Frontend Image 2s
- > ✓ Deploy to Web Server 11s
- > ✓ Post Checkout Code 0s
- > ✓ Complete job 0s

Si aplicatia finala in web browser:



BONUS: am instalat tailscale pe serverul web (ca acolo e nginx)



02:14



LearnPremium Courses Login Register

Master Your Future

Access world-class education from industry experts. Start your journey with our premium online courses today.

Browse Courses

Join for Free



Not Secure — 1.16.66



Q. Cu ce ne ajuta Docker concret la deploy-ul proiectului?

A. Exemplu concret: daca lucram fara docker, trebuia sa intru pe sv de web si sa instalez manual runtimeul de .net 10, baza de date, si alte chestii, daca trebuia sa trec pe viitor la .net 11, trebuia iar sa fac actualizarile astea manual, asa cu docker am instalat doar pe el, deja venind impachetat cu .net 10 inauntru. Inca un exemplu ar mai fi evitarea de conflicte ale versionarii aplicatiilor sau toolurilor, daca copiam direct fisierele de "acasa" pe server, poate lipsea ceva librerie si nu mergea, cu docker, cand am dat docker build pe serverul de devops, codul a fost pus intr-un container care contine propriul mini sistem de operare, acel container ruleaza pe serverul de web ,exact acelasi mediu in care a fost construit

Q. De ce folosim .env în loc să scriem parolele direct în docker-compose.yml?

A. Din motive de securitate, nu se hardcodeaza parole care apoi or sa fie pushate pe git niciodata, parolele se pun in .env , iar .env se adauga in .gitignore ca sa nu ajunge pe git, ramane doar pe local, eu nu am folosit .env aici din 2 motive: primul ar fi ca am un runner pe github si e "semi-bresa" de securitate sa ai un runner private pe repo public, si al doilea este ca nu am vrut sa ma "complic", si asa nu puteam sa pun repo-ul public si nu aveam de ce sa lucrez cu .env

Q. Ce rol are nginx reverse proxy în fața containerelor voastre (de ce nu expuneți direct portul aplicației la internet)?

A. Se foloseste pt a nu expune direct serverele de aplicatie la internet, asa mai ai un strat in plus de securitate, terminare de SSL si load balancing

Q. Ce ați învățat azi nou și ce vi s-a părut cel mai interesant?

A. Am invatat ca nu stiu (inca!) sa scriu docker files sau fisiere de configurare YAML care sunt vitale pt pipeline, legat de partea non-coding a proiectului sa zic asa, majoritatea chestiilor le invatasem singur deja(mai putin asta cu scrisul de fisiere de configurare), nu stiam precis niste cai de configurare prin nexus,cum se cheama de exemplu calea precisa din care poti porni nexusul, mi se pare foarte interesant felul in care functioneaza docker, modularizeaza totul si face chiar error proof aplicatia