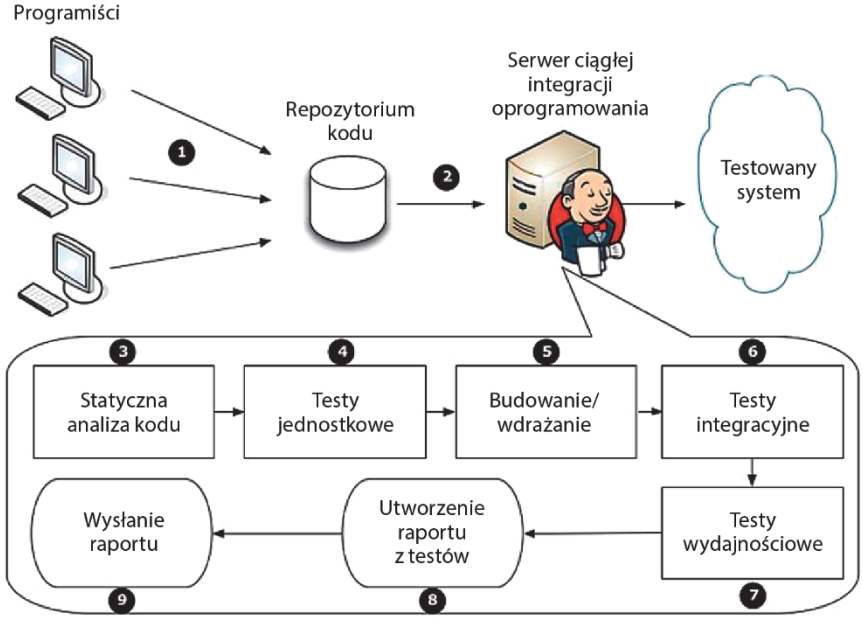
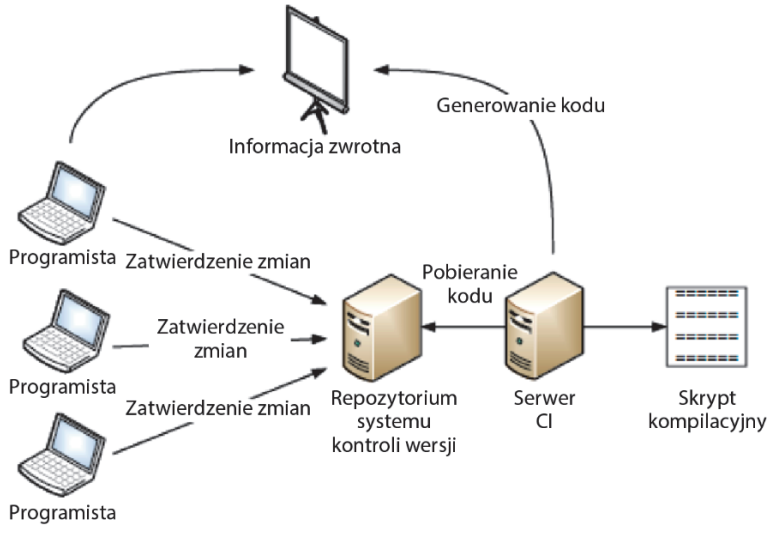
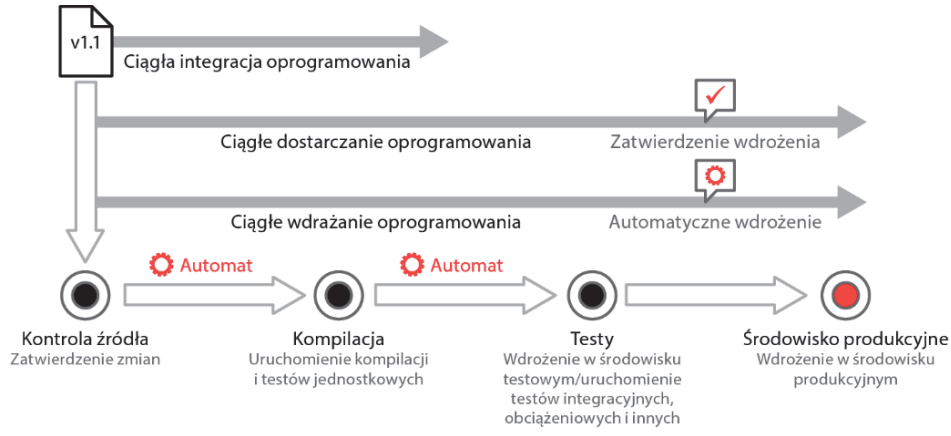


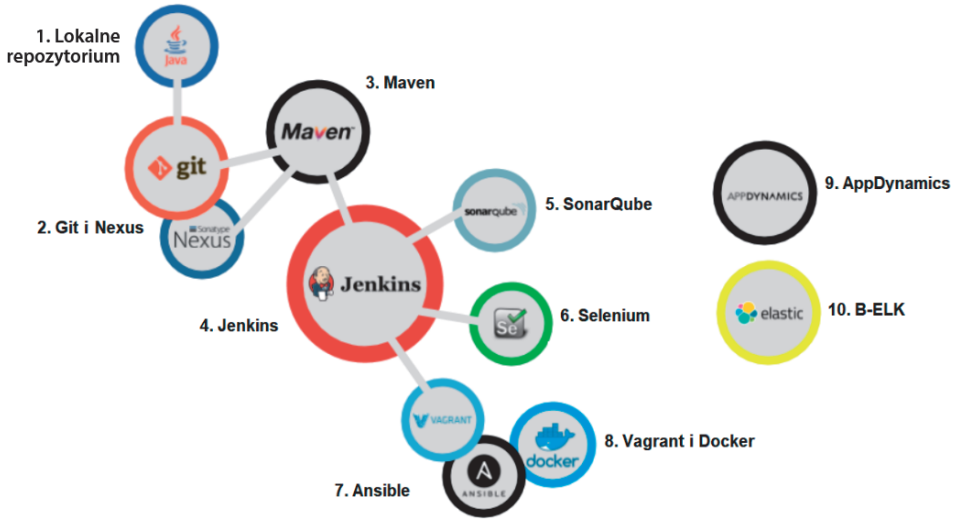
Rysunek 1.3. *Proces ciągłego dostarczania oprogramowania*



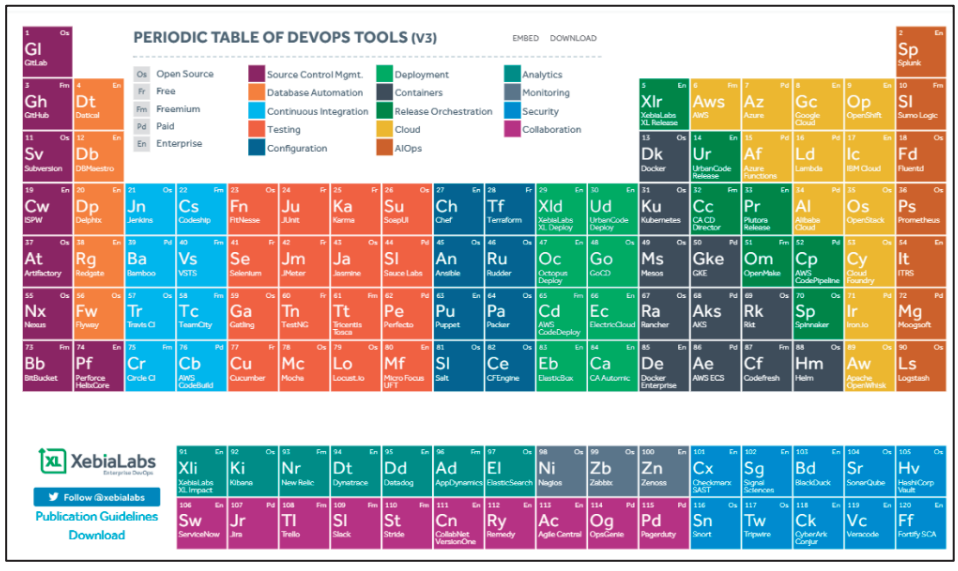
Rysunek 1.4. *Proces ciągłej integracji oprogramowania*



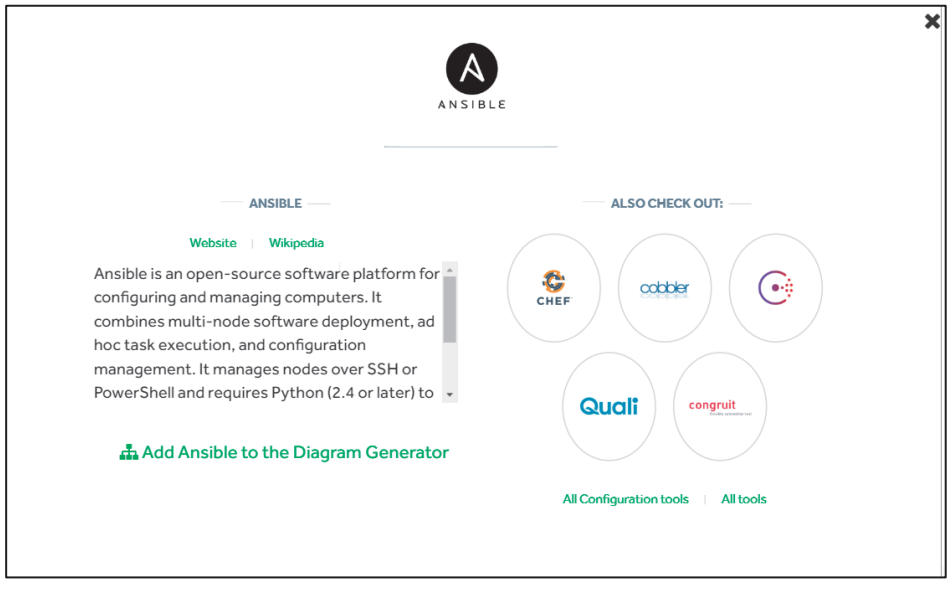
Rysunek 1.5. *Kanał dostarczania oprogramowania*



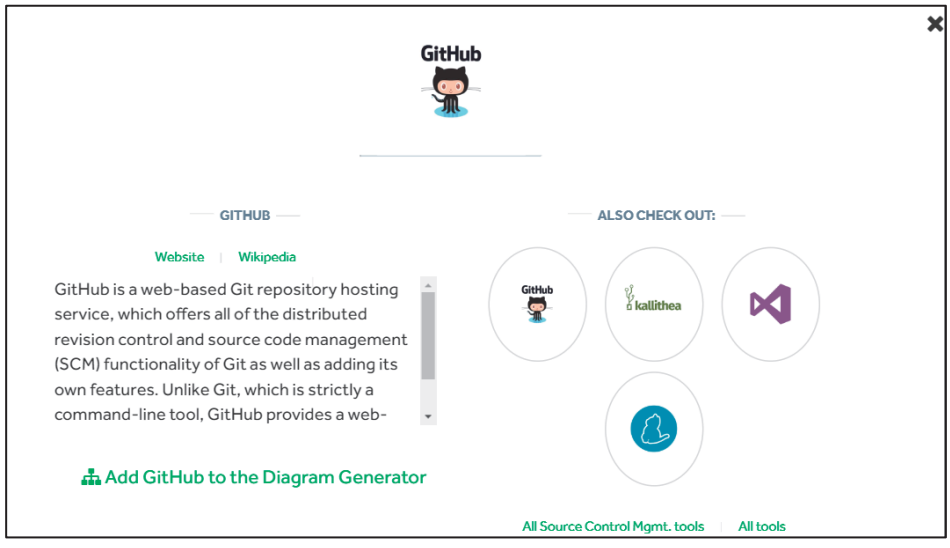
Rysunek 1.6. *Narzędzia DevOps*



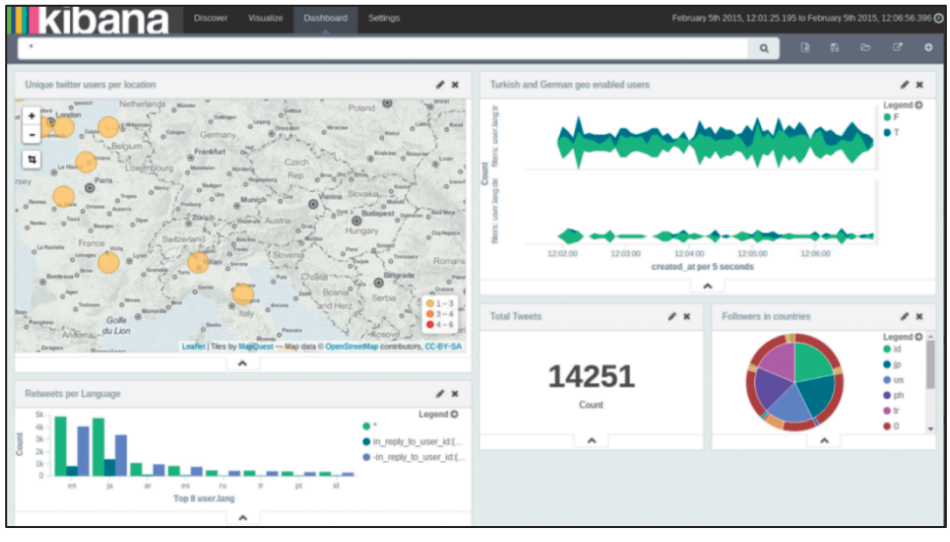
Rysunek 1.7. *Układ okresowy narzędzi DevOps*



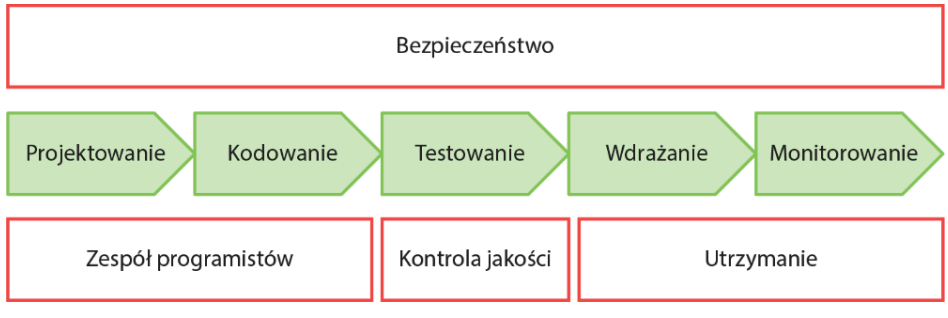
Rysunek 1.8. *Opis narzędzia Ansible*



Rysunek 1.9. *Opis narzędzia GitHub*



Rysunek 1.10. *Narzędzie Kibana do analizy danych*



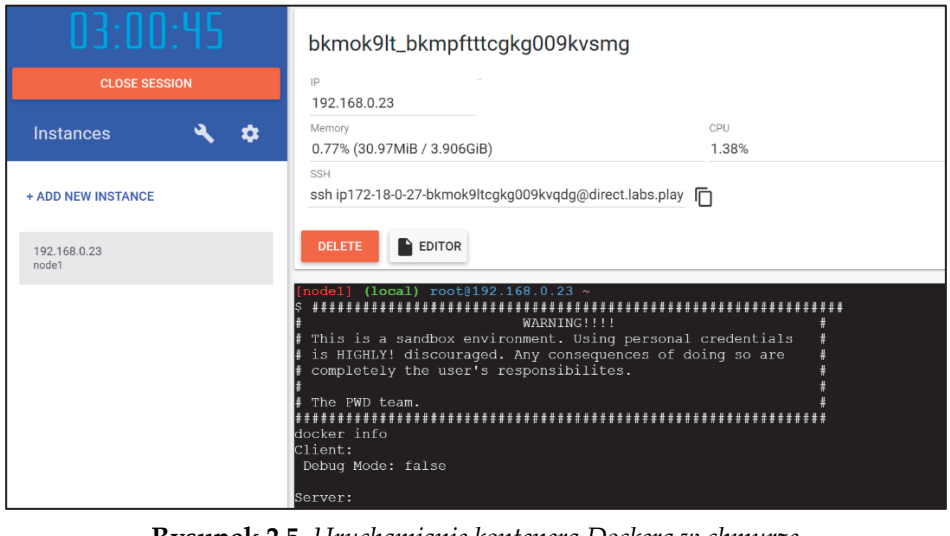
Rysunek 1.11. *Cykl DevSecOps*

```
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5b7339215d1d: Pull complete
14ca88e9f672: Pull complete
a31c3b1caad4: Pull complete
b054a26005b7: Pull complete
Digest: sha256:9b1702dcfe32c873a770a32cfd306dd7fc1c4fd134adfb783db68defc8894b3c
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[node1] (local) root@192.168.0.23 ~
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              4c108a37151f        3 weeks ago         64.2MB
```

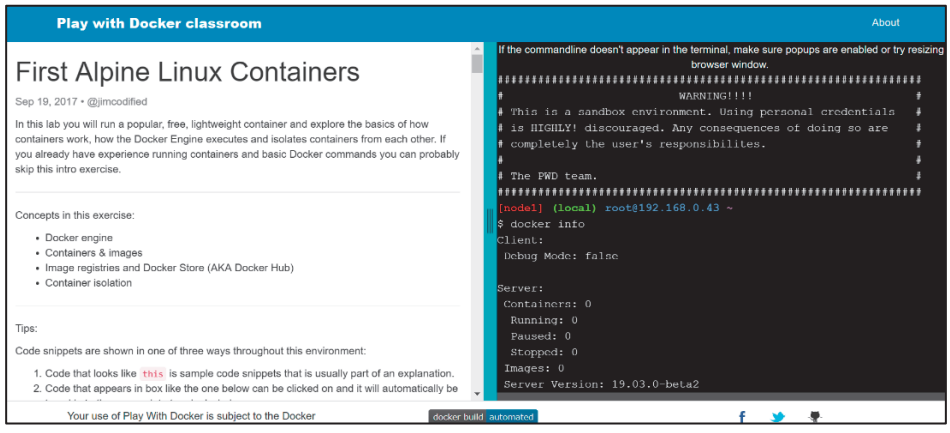
Rysunek 2.3. *Przykład użycia polecenia docker pull*

```
$ docker run -i -t ubuntu  /bin/bash
root@760dca7304c6:/# ls -l
total 4
drwxr-xr-x    2 root root 4096 Jun 12 16:55 bin
drwxr-xr-x    2 root root    6 Apr 24  2018 boot
drwxr-xr-x    5 root root  360 Jul 16 09:10 dev
drwxr-xr-x    1 root root   66 Jul 16 09:10 etc
drwxr-xr-x    2 root root    6 Apr 24  2018 home
drwxr-xr-x    8 root root   96 May 23  2017 lib
drwxr-xr-x    2 root root   34 Jun 12 16:55 lib64
drwxr-xr-x    2 root root    6 Jun 12 16:54 media
drwxr-xr-x    2 root root    6 Jun 12 16:54 mnt
drwxr-xr-x    2 root root    6 Jun 12 16:54 opt
dr-xr-xr-x 1179 root root    0 Jul 16 09:10 proc
drwx------    2 root root   37 Jun 12 16:55 root
drwxr-xr-x    1 root root   21 Jun 18 22:51 run
drwxr-xr-x    1 root root   21 Jun 18 22:51 sbin
drwxr-xr-x    2 root root    6 Jun 12 16:54 srv
dr-xr-xr-x   13 root root    0 Jul 15 01:26 sys
drwxrwxrwt    2 root root    6 Jun 12 16:55 tmp
drwxr-xr-x    1 root root   18 Jun 12 16:54 usr
drwxr-xr-x    1 root root   17 Jun 12 16:55 var
```

Rysunek 2.4. *Wewnątrz kontenera*



Rysunek 2.5. *Uruchamianie kontenera Dockera w chmurze*



Rysunek 2.6. *Serwis Play with Docker*

```
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILA
BILITY         MANAGER STATUS      ENGINE VERSION
vz79cf4danb0njb3mg9p8vwpu *   node1               Ready               Active
               Leader              19.03.0-beta2
trmx4medo7ffaxy0e3z5xkhvt     node2               Ready               Active
                                   19.03.0-beta2
[node1] (local) root@192.168.0.37 ~
$
```

```
# completely the user's responsibilites.                    #
#                                                           #
# The PWD team.                                             #
#############################################################
[node2] (local) root@192.168.0.38 ~
$ docker swarm join --token SWMTKN-1-1u4oi5o8db4831ipn37cyof8attas3h0xt2bf2y0fdf2my0id5-6xje7khf
2.168.0.37:2377
This node joined a swarm as a worker.
```

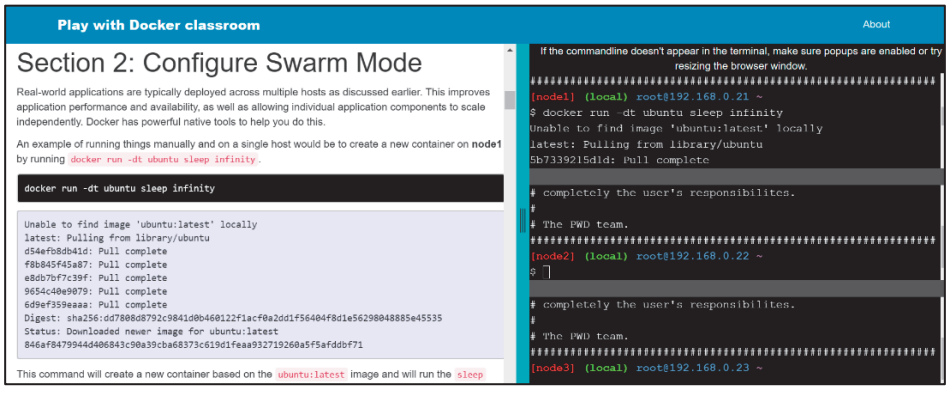
Rysunek 2.8. *Dodanie węzła roboczego do klastra*

```
$ docker service create --replicas 1 --name helloworld alpine ping www.google.com
4b5umtqhuhjxj88xv3yk91520
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged
[node1] (local) root@192.168.0.23 ~
$ docker service ls
ID                  NAME                MODE                REPLICAS            IMA
GE                PORTS
4b5umtqhuhjx        helloworld          replicated          1/1                 alp
ine:latest
```

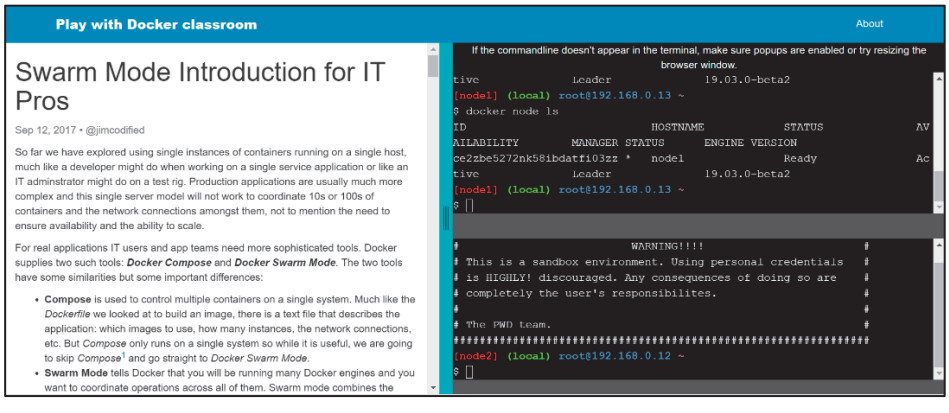
Rysunek 2.9. *Tworzenie repliki usługi w klastrze Dockera*

```
$ docker service ps helloworld
ID                  NAME                IMAGE               NODE                DESIRED STATE
      CURRENT STATE               ERROR               PORTS
mf33ajplktg3        helloworld.1        alpine:latest       node2               Running
      Running about a minute ago
[node1] (local) root@192.168.0.23 ~
$ docker service scale helloworld=4
helloworld scaled to 4
[node1] (local) root@192.168.0.23 ~
$ docker service ps helloworld
ID                  NAME                IMAGE               NODE                DESIRED STATE
      CURRENT STATE           ERROR               PORTS
mf33ajplktg3        helloworld.1        alpine:latest       node2               Running
      Running 2 minutes ago
tidzecyit0z8        helloworld.2        alpine:latest       node2               Running
      Running 7 seconds ago
fqmov6k5j2wa        helloworld.3        alpine:latest       node1               Running
      Running 7 seconds ago
3qx814c75j1v        helloworld.4        alpine:latest       node1               Running
      Running 7 seconds ago
```

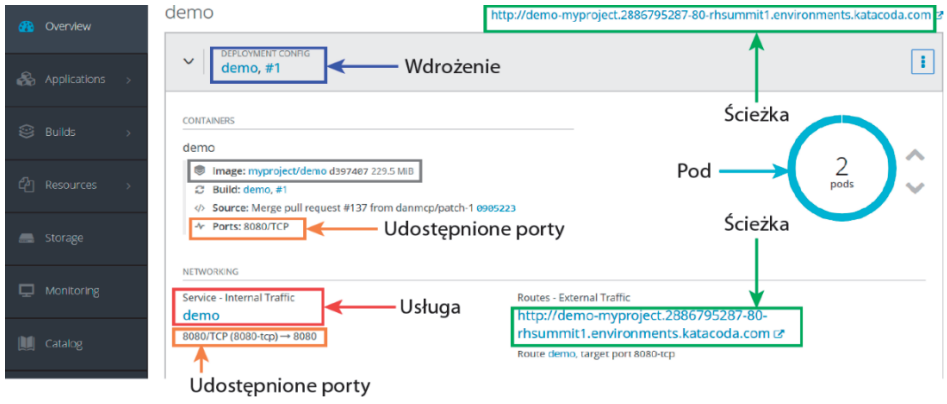
Rysunek 2.10. *Zwiększenie liczby replik usługi w klastrze*



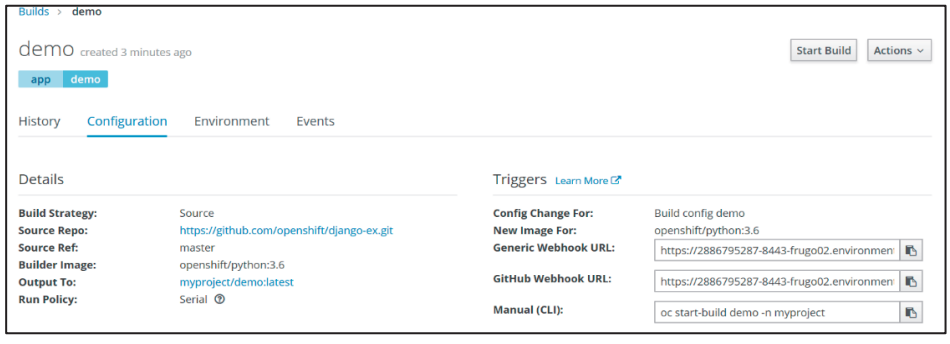
Rysunek 2.11. *Konfiguracja trybu Swarm*



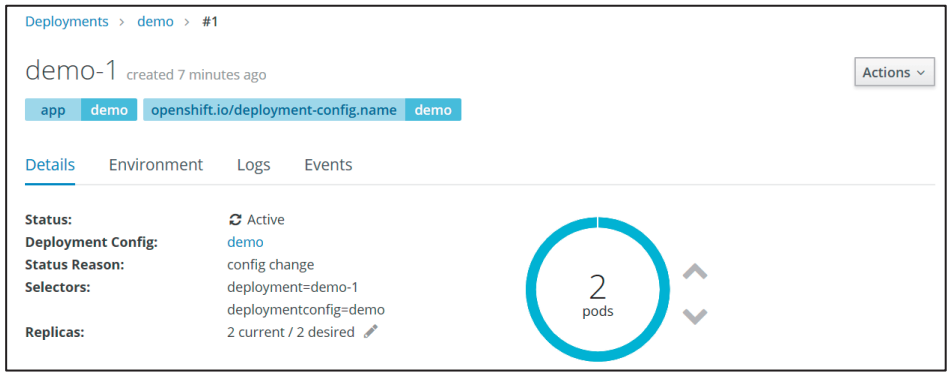
Rysunek 2.12. *Wprowadzenie do trybu Swarm dla profesjonalistów IT*



Rysunek 2.13. *Projekt OpenShift*

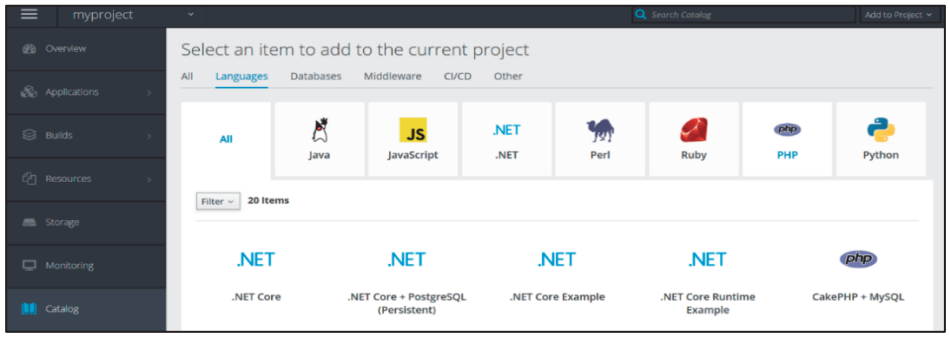


Rysunek 2.14. *Konfiguracja wdrożenia*

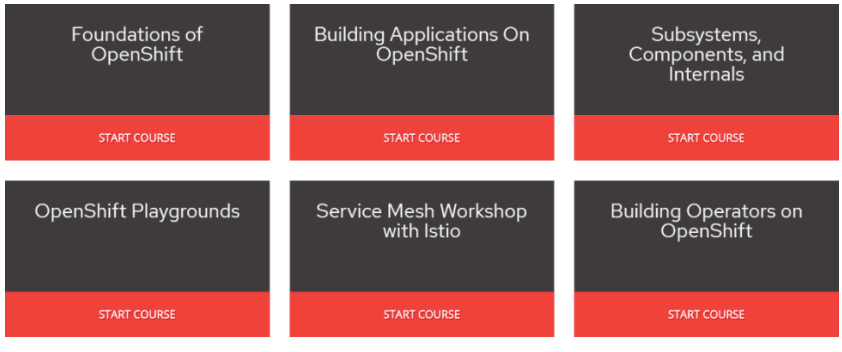


Rysunek 2.15. *Szczegóły konfiguracyjne wdrożenia*

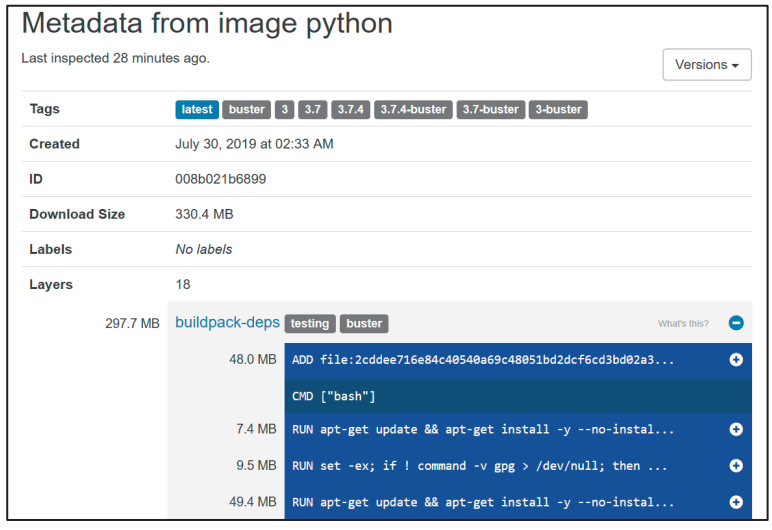
Rysunek 2.16. *Lista uruchomionych podów*



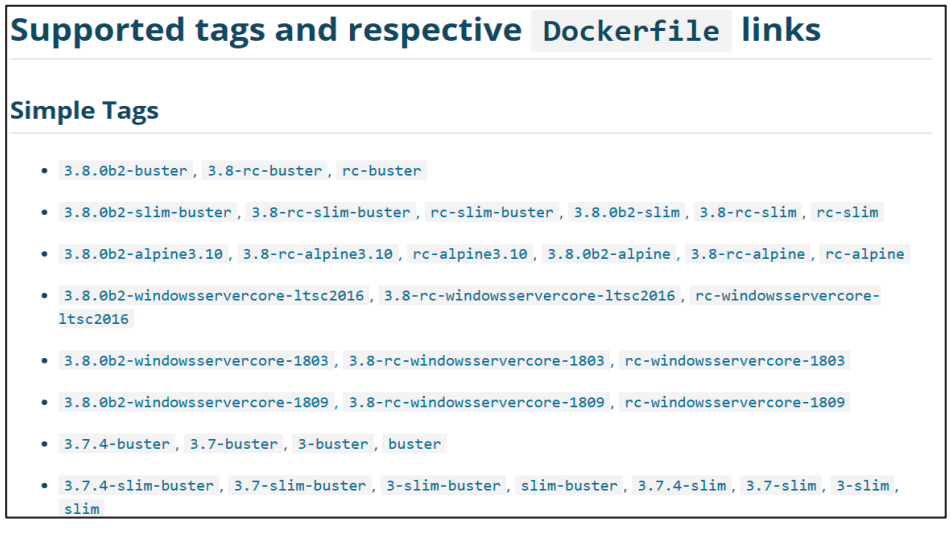
Rysunek 2.17. *Wbudowane szablony platformy OpenShift*



Rysunek 2.18. *Scenariusze szkoleniowe platformy OpenShift*



Rysunek 3.3. *Strona usługi MicroBadger prezentująca metadane obrazu zawierającego środowisko Pythona*



Rysunek 3.4. *Etykiety oficjalnego obrazu zawierającego środowisko Pythona, umieszczonego w serwisie Docker Hub*

```
$ docker image pull python:3.8-rc-alpine3.10
3.8-rc-alpine3.10: Pulling from library/python
050382585609: Pull complete
dac2222ca532: Pull complete
a5a8a13f5210: Pull complete
48ed6fe4c480: Pull complete
f5c21fef32f5: Pull complete
Digest: sha256:e686f6b5cf95f23bbb19d4f38a9f541abfdca0f7f6be6b74fbf862db068793be
Status: Downloaded newer image for python:3.8-rc-alpine3.10
docker.io/library/python:3.8-rc-alpine3.10
[node1] (local) root@192.168.0.43 ~
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python              latest              14a2caeca327        17 hours ago        918MB
python              3.8-rc-alpine3.10   70da12d86711        17 hours ago        109MB
```

Rysunek 3.5. *Etykieta pobranego obrazu*

```
$ docker build -t fedora_image .
Sending build context to Docker daemon  1.747MB
Step 1/4 : FROM fedora:latest
 ---> 2b74bf3d2430
Step 2/4 : MAINTAINER maintainer
 ---> Using cache
 ---> 471a4d43a7c5
Step 3/4 : RUN echo "This container was built on $(date)." > /tmp/built.txt
 ---> Using cache
 ---> 2c90489b36b6
Step 4/4 : ENTRYPOINT ["cat","/tmp/built.txt"]
 ---> Using cache
 ---> 8a83454955f9
Successfully built 8a83454955f9
Successfully tagged fedora_image:latest
[node1] (local) root@192.168.0.43 ~
$ docker run fedora_image
This container was built on Tue Jul 30 18:19:06 UTC 2019.
```

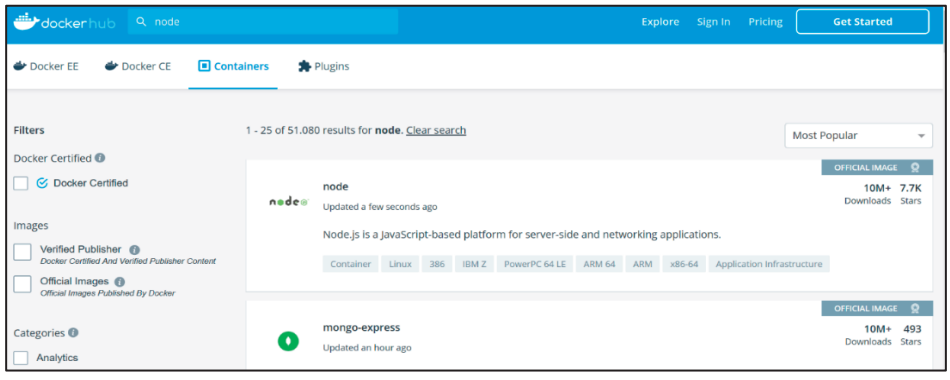
Rysunek 3.6. *Przykład użycia poleceń docker build i docker run*

```
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
test_dockerfile     latest              e22b016f25f7        3 seconds ago       176MB
ubuntu              latest              3556258649b2        3 days ago          64.2MB
[node1] (local) root@192.168.0.33 ~/ubuntutest
$ docker run -dti --name mycontainer e22b016f25f7
aca0408912343d042b3352146dd87e891179941498237e220a5d254175288625
[node1] (local) root@192.168.0.33 ~/ubuntutest
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
           PORTS               NAMES
aca040891234        e22b016f25f7        "bash"              12 seconds ago      Up 10 sec
onds                            mycontainer
```

Rysunek 3.8. *Polecenia docker images, docker run i docker ps*

```
$ docker exec -i -t mycontainer /bin/bash
root@aca040891234:/# dstat
You did not select any stats, using -cdngy by default.
--total-cpu-usage-- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai stl| read  writ| recv  send|  in   out | int   csw
 31  17  52   0   0| 174k 2591k|   0     0 |   0     0 |  12k   47k
 28  27  45   0   0|   0     0 |   0     0 |   0     0 |  27k   87k
 31  28  41   0   0|   0    32k|   0     0 |   0     0 |  28k  100k
 28  25  47   0   0|   0  4096B|   0     0 |   0     0 |  28k   89k
 30  30  39   0   0|   0   396k|   0     0 |   0     0 |  28k  100k
 29  26  45   0   0|   0   296k|   0     0 |   0     0 |  28k   91k
 31  29  41   0   0|   0  4096B|   0     0 |   0     0 |  27k   99k
 28  26  45   0   0|   0    48k|   0     0 |   0     0 |  27k   89k
```

Rysunek 3.9. *Przykład użycia polecenia dstat wewnątrz kontenera*



Rysunek 3.11. *Wyszukiwanie obrazu w serwisie Docker Hub*

```
$ docker pull gcr.io/distroless/python3
Using default tag: latest
latest: Pulling from distroless/python3
e8d8785a314f: Pull complete
e005d777a298: Pull complete
3e010093287c: Pull complete
609f69c3154c: Pull complete
Digest: sha256:b83bd4dc7c34d1c3a1b8400474163fccfe5b9110d0d1cb12b48e1786473d5ba2
Status: Downloaded newer image for gcr.io/distroless/python3:latest
gcr.io/distroless/python3:latest
[node1] (local) root@192.168.0.43 ~
$ docker pull python:alpine
alpine: Pulling from library/python
050382585609: Pull complete
dac2222ca532: Pull complete
29a7fe408caa: Pull complete
6ad337b9b53f: Pull complete
31d663a76478: Pull complete
Digest: sha256:d22196e0ced4a0fd44916e3ff4aea00565260f66a3d0d26f5551b8fdbd833423
Status: Downloaded newer image for python:alpine
docker.io/library/python:alpine
```

Rysunek 3.20. *Pobranie pełnego obrazu oraz jego okrojonej wersji zawierającej środowisko Pythona i system operacyjny Alpine*

```
1 FROM python:3-slim AS build-env
2 ADD . /app
3 WORKDIR /app
4
5 FROM gcr.io/distroless/python3
6 COPY --from=build-env /app /app
7 WORKDIR /app
```

Rysunek 3.22. *Przykładowy plik Dockerfile wykorzystywany do tworzenia okrojonego obrazu zawierającego środowisko Pythona 3*

```
$ docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
python3                       latest              f6d85470e9f6        21 minutes ago      52.5MB
<none>                        <none>              801bdc56f2ed        23 minutes ago      52.5MB
python                        3-slim              ca7f9e245002        2 weeks ago         143MB
python                        latest              a4cc999cf2aa        2 weeks ago         929MB
gcr.io/distroless/python3     latest              b31fedb42763        49 years ago        50.9MB
```

```
$ docker run -it python bash
root@a4c6043a9fef:/# exit
exit
[node1] (local) root@192.168.0.8 ~
$ docker run -it python3 bash
/usr/bin/python3.5: can't open file 'bash': [Errno 2] No such file or directory
[node1] (local) root@192.168.0.8 ~
```

Rysunek 3.23. *Uruchomienie okrojonego obrazu zawierającego środowisko Pythona 3*

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS                 NAMES
79b33d74c408        mysql               "docker-entrypoint.s…"   20 seconds ago      Up 19 seconds
3306/tcp, 33060/tcp   mysql
[node1] (local) root@192.168.0.38 ~
$ docker exec mysql touch /opt/filename
touch: cannot touch '/opt/filename': Read-only file system
```

Rysunek 4.4. *Uruchomienie kontenera z bazą MySQL i woluminem*

```
$ docker pull python@sha256:35ff9f44818f8850f1d318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
sha256:35ff9f44818f8850f1d318aa69c2e7ba61d85e3b93283078c10e56e7d864c183: Pulling from library/python
c5e155d5a1d1: Already exists
221d80d00ae9: Already exists
4250b3117dca: Already exists
3b7ca19181b2: Already exists
425d7b2a5bcc: Already exists
dc3049ff3f44: Pull complete
472a6afc6332: Pull complete
5f79c90f8d7c: Pull complete
1051ee813012: Pull complete
Digest: sha256:35ff9f44818f8850f1d318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
Status: Downloaded newer image for python@sha256:35ff9f44818f8850f1d318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
docker.io/library/python@sha256:35ff9f44818f8850f1d318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
[node1] (local) root@192.168.0.28 ~
$ docker pull python@sha256:35ff9f44818f8850f1d318aa69c2e7ba61d85e3b93283078c10e56e7d864c1831
invalid checksum digest length
```

Rysunek 4.19. *Pobranie obrazu Dockera zawierającego środowisko Pythona i sprawdzenie jego sumy kontrolnej*

```
$ export DOCKER_CONTENT_TRUST=1
[node1] (local) root@192.168.0.28 ~
$ docker pull jmortegac/linux_tweet_app:1.0
Error: remote trust data does not exist for docker.io/jmortegac/linux_tweet_app: notary.docker.io does not have trust
 data for docker.io/jmortegac/linux_tweet_app
[node1] (local) root@192.168.0.28 ~
$ export DOCKER_CONTENT_TRUST=0
[node1] (local) root@192.168.0.28 ~
$ docker pull jmortegac/linux_tweet_app:1.0
1.0: Pulling from jmortegac/linux_tweet_app
bc95e04b23c0: Pull complete
a21d9ee25fc3: Pull complete
9bda7d5afd39: Pull complete
e64d34b6ad71: Pull complete
a7e018a2b8ff: Pull complete
Digest: sha256:db9f2e75b91780c804c283081123fbe1b2b4080fe124eeb040f8ad1bfd70fe38
Status: Downloaded newer image for jmortegac/linux_tweet_app:1.0
docker.io/jmortegac/linux_tweet_app:1.0
```

Rysunek 4.20. *Pobieranie obrazu przy włączonej i wyłączonej funkcjonalności DCT*

```
ENV GPG_KEY E3FF2839C048B25C084DEBE9B26995E310250568
ENV PYTHON_VERSION 3.8.0a4

RUN set -ex \
        \
        && wget -O python.tar.xz "https://www.python.org/ftp/python/${PYTHON_VERSION%%[a-z]*}/Python-$PYTHON_VERSION.tar.xz" \
        && wget -O python.tar.xz.asc "https://www.python.org/ftp/python/${PYTHON_VERSION%%[a-z]*}/Python-$PYTHON_VERSION.tar.xz.asc" \
        && export GNUPGHOME="$(mktemp -d)" \
        && gpg --batch --keyserver ha.pool.sks-keyservers.net --recv-keys "$GPG_KEY" \
        && gpg --batch --verify python.tar.xz.asc python.tar.xz \
        && { command -v gpgconf > /dev/null && gpgconf --kill all || :; } \
        && rm -rf "$GNUPGHOME" python.tar.xz.asc \
        && mkdir -p /usr/src/python \
        && tar -xJC /usr/src/python --strip-components=1 -f python.tar.xz \
        && rm python.tar.xz \
        \
```

Rysunek 4.21. *Bezpieczne pobieranie obrazu z wykorzystaniem pliku Dockerfile*

```
$ docker run -d -p 5000:5000 --restart=always --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
c87736221ed0: Pull complete
1cc8e0bb44df: Pull complete
54d33bcb37f5: Pull complete
e8afc091c171: Pull complete
b4541f6d3db6: Pull complete
Digest: sha256:77a8fb00c00b99568772a70f0863f6192ff2635e4af4e22e4d9c622edeb5f2de
Status: Downloaded newer image for registry:2
d47e0b90b502870403e3f634632ffd4012c792b02aba11f4264cc5eed56c7089
[node1] (local) root@192.168.0.28 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
                    NAMES
d47e0b90b502        registry:2          "/entrypoint.sh /etc…"   2 minutes ago       Up 2 minutes        0.0.0.
0:5000->5000/tcp    registry
```

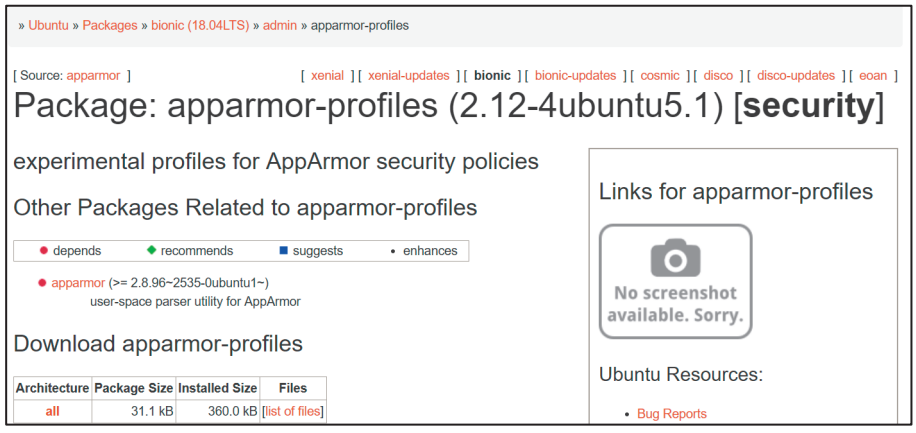
Rysunek 4.23. *Pobranie kontenera w celu utworzenia lokalnego rejestru*

Open

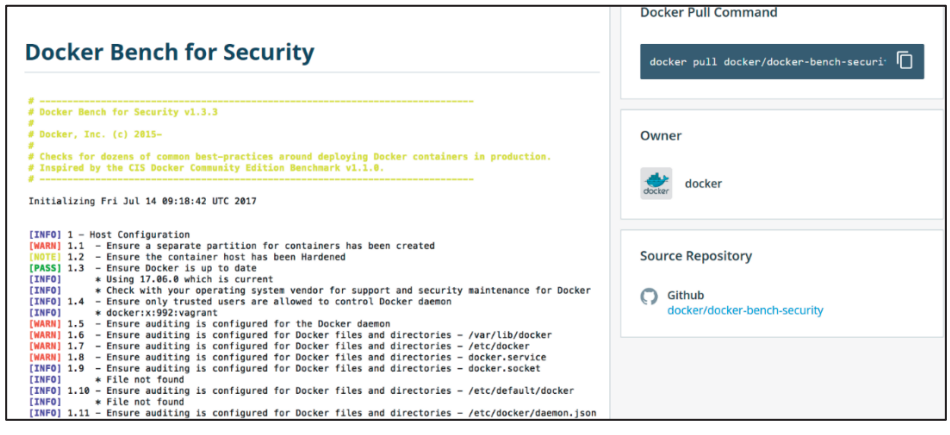
audit.rules
/etc/audit/rules.d

```
## Rules
-w /usr/bin/docker -k docker
-w /var/lib/docker -k docker
-w /etc/docker -k docker
-w /usr/lib/systemd/system/docker.service -k docker
-w /usr/lib/systemd/system/docker.socket -k docker
-w /etc/default/docker -k docker
-w /etc/docker/daemon.json -k docker
-w /usr/bin/docker-containerd -k docker
-w /usr/bin/docker.runc -k docker
```

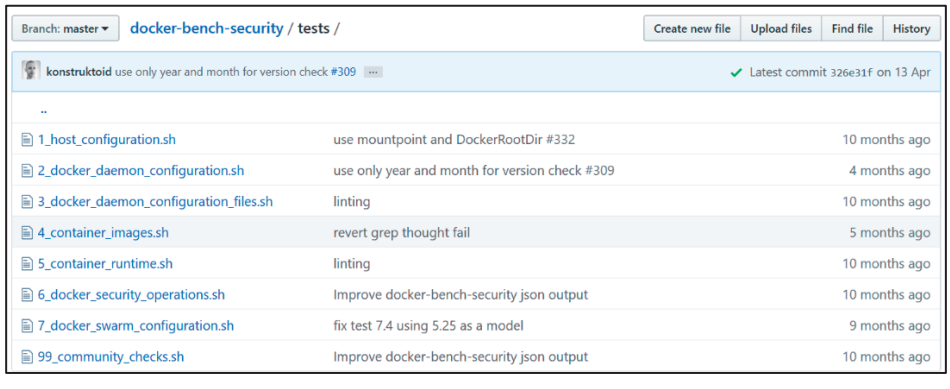
Rysunek 5.4. *Definiowanie reguł audytu*



Rysunek 5.6. *Pakiet modułu AppArmor dla systemu Ubuntu*



Rysunek 5.10. *Obraz Docker Bench for Security w serwisie Docker Hub*



Rysunek 5.17. *Repozytorium GitHub*

```
desc_5_1="Ensure AppArmor Profile is Enabled"
check_5_1="$id_5_1  - $desc_5_1"
starttestjson "$id_5_1" "$desc_5_1"

totalChecks=$((totalChecks + 1))

fail=0
no_apparmor_containers=""
for c in $containers; do
  policy=$(docker inspect --format 'AppArmorProfile={{ .AppArmorProfile }}' "$c")

  if [ "$policy" = "AppArmorProfile=" ] || [ "$policy" = "AppArmorProfile=[]" ] || [ "$policy" = "AppArmorProfile=<no value>" ]; then
    # If it's the first container, fail the test
    if [ $fail -eq 0 ]; then
      warn "$check_5_1"
      warn "     * No AppArmorProfile Found: $c"
      no_apparmor_containers="$no_apparmor_containers $c"
      fail=1
    else
      warn "     * No AppArmorProfile Found: $c"
      no_apparmor_containers="$no_apparmor_containers $c"
    fi
  fi
```

Rysunek 5.18. *Skrypt 5_container_runtime.sh sprawdzający profil AppArmor*

```
  desc_5_2="Ensure SELinux security options are set, if applicable"
  check_5_2="$id_5_2  - $desc_5_2"
  starttestjson "$id_5_2" "$desc_5_2"

  totalChecks=$((totalChecks + 1))

  fail=0
  no_securityoptions_containers=""
  for c in $containers; do
    policy=$(docker inspect --format 'SecurityOpt={{ .HostConfig.SecurityOpt }}' "$c")

    if [ "$policy" = "SecurityOpt=" ] || [ "$policy" = "SecurityOpt=[]" ] || [ "$policy" = "SecurityOpt=<no value>" ]; then
      # If it's the first container, fail the test
      if [ $fail -eq 0 ]; then
        warn "$check_5_2"
        warn "     * No SecurityOptions Found: $c"
        no_securityoptions_containers="$no_securityoptions_containers $c"
        fail=1
      else
        warn "     * No SecurityOptions Found: $c"
        no_securityoptions_containers="$no_securityoptions_containers $c"
      fi
    fi
```

Rysunek 5.19. *Filtr SecurityOpt w skrypcie 5_container_runtime.sh*

```
for c in $containers; do
  user=$(docker inspect --format 'User={{.Config.User}}' "$c")

  if [ "$user" = "User=0" ] || [ "$user" = "User=root" ] || [ "$user" = "User=" ] || [ "$user" = "User=[]" ] || [ "$user" = "User=<no v
    # If it's the first container, fail the test
    if [ $fail -eq 0 ]; then
      warn "$check_4_1"
      warn "     * Running as root: $c"
      root_containers="$root_containers $c"
      fail=1
    else
      warn "     * Running as root: $c"
      root_containers="$root_containers $c"
    fi
  fi
done
```

Rysunek 5.20. *Skrypt 4_container_images.sh sprawdzający konto wykorzystywane wewnątrz kontenera*

```
desc_2_6="Ensure TLS authentication for Docker daemon is configured"
check_2_6="$id_2_6  - $desc_2_6"
starttestjson "$id_2_6" "$desc_2_6"

totalChecks=$((totalChecks + 1))
if [ grep -i 'tcp://' "$CONFIG_FILE" 2>/dev/null 1>&2 ] || \
  [ $(get_docker_cumulative_command_line_args '-H' | grep -vE '(unix|fd)://') >/dev/null 2>&1 ]; then
  if [ $(get_docker_configuration_file_args '"tlsverify":' | grep 'true') ] || \
      [ $(get_docker_cumulative_command_line_args '--tlsverify' | grep 'tlsverify') >/dev/null 2>&1 ]; then
    pass "$check_2_6"
    resulttestjson "PASS"
    currentScore=$((currentScore + 1))
  elif [ $(get_docker_configuration_file_args '"tls":' | grep 'true') ] || \
      [ $(get_docker_cumulative_command_line_args '--tls' | grep 'tls$') >/dev/null 2>&1 ]; then
    warn "$check_2_6"
    warn "     * Docker daemon currently listening on TCP with TLS, but no verification"
    resulttestjson "WARN" "Docker daemon currently listening on TCP with TLS, but no verification"
    currentScore=$((currentScore - 1))
```

Rysunek 5.21. *Fragment skryptu sprawdzający szyfrowanie TLS*

```
FIND=$(grep "^FROM" ${AUDIT_FILE} | sed 's/ /:space:/g')
for I in ${FIND}; do
    IMAGE=$(echo ${I} | sed 's/:space:/ /g' | awk '{ if ($1=="FROM") { print $2 }}')
    TAG=$(echo ${IMAGE} | cut -d':' -f2)
    Display --indent 2 --text "Found image:" --result "${IMAGE}"

    IS_DEBIAN=$(echo ${IMAGE} | grep -i debian)
    IS_FEDORA=$(echo ${IMAGE} | grep -i fedora)
    IS_UBUNTU=$(echo ${IMAGE} | grep -i ubuntu)
    IS_ALPINE=$(echo ${IMAGE} | grep -i alpine)
    IS_LATEST=$(echo ${TAG} | grep -i latest)

    if [ -n "${IS_DEBIAN}" ]; then IMAGE="debian"; fi
    if [ -n "${IS_FEDORA}" ]; then IMAGE="fedora"; fi
    if [ -n "${IS_UBUNTU}" ]; then IMAGE="ubuntu"; fi
    if [ -n "${IS_ALPINE}" ]; then IMAGE="alpine"; fi
    if [ -n "${IS_LATEST}" ]; then
        ReportWarning "dockerfile" "latest TAG used. Specifying a targeted OS image and version is better for reproducible results."
    fi
```

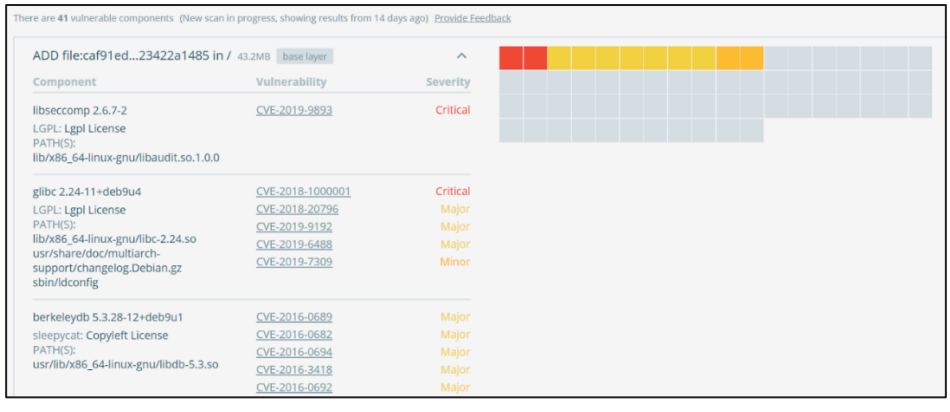
Rysunek 5.29. *Kod uzyskujący typ obrazu*

```
if [ ${FILE_DOWNLOAD} -eq 1 ]; then

    SSL_USED_FIND=$(egrep "(https)" ${AUDIT_FILE})

    if HasData "${SSL_USED_FIND}"; then
        SSL_USED="YES"
        COLOR="GREEN"
    else
        SSL_USED="NO"
        COLOR="RED"
        ReportSuggestion "Use SSL downloads when possible to increase security (DNSSEC, HTTPS, validation of domain, avoid MitM)"
    fi
    Display --indent 2 --text "Integrity testing performed" --result "${SSL_USED}" --color ${COLOR}
    HASHING_USED=$(egrep "(sha1sum|sha256sum|sha512sum)" ${AUDIT_FILE})
    Display --indent 2 --text "Hashing" --result "${HASHING_USED}"
    KEYS_USED=$(egrep "(apt-key adv)" ${AUDIT_FILE}| sed 's/RUN apt-key adv//g'| sed 's/--keyserver/Key Server:/g' | sed 's/--recv/Key
    Display --indent 2 --text "Signing keys used" --result "${KEYS_USED}"
    Display --indent 2 --text "All downloads properly checked" --result "?"
else
    Display --indent 2 --text "No files seems to be downloaded in this Dockerfile"

fi
```

Rysunek 5.30. *Sprawdzenie pobranych pakietów*

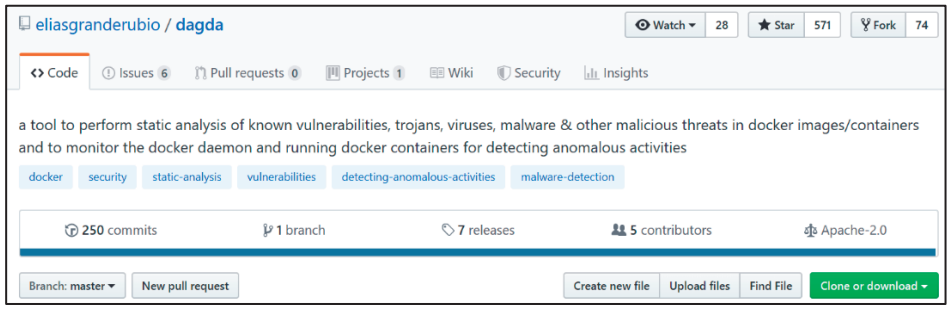
```
[root@dockerlab001 ~]# more myreport.txt
Dockscan Report

-[ Medium ]-
=Docker running with IPv4 forwarding enabled=
Description:
Docker daemon reports it is running daemon with IPv4 forwarding enabled.
This is not recommended for production as it forwards network packets without rules.
Output:
Docker daemon reports it is running with automatic IPv4 forwarding.
Solution:
It is recommended to disable IPv4 forwarding by default.


-[ Low ]-
=Container have higher number of changed files=
Description:
Container have high number of changed files which is not recommended practice.
This is not recommended for production as data can be lost. It can also mean successful break in attempt.
Output:
475bed6f82c3124e9602e5992a2d546e1e80aab08364184440a389893703505e (/traefik_proxy_1 ) with IP:  has more than 5 file changes: 9
/etc
/etc/traefik
/etc/traefik/traefik.toml
/run
/run/secrets
/tmp
/var
/var/run
/var/run/docker.sock


Solution:
It is recommended to have minimal number of changed files inside container and do not store data inside container. It is recommended to use volumes.

=Docker registries are not mirrored=
Description:
Docker daemon reports it is running configuration without registry mirrors.
If you set up local mirror, your docker host does not have to go directly to internet if not needed.
Output:
Docker daemon reports it does not have mirror registries.
Offending registry indexes:
docker.io


Solution:
It is recommended to setup mirror registry.
```

Rysunek 5.33. *Raport opisujący luki w bezpieczeństwie hosta platformy Docker*

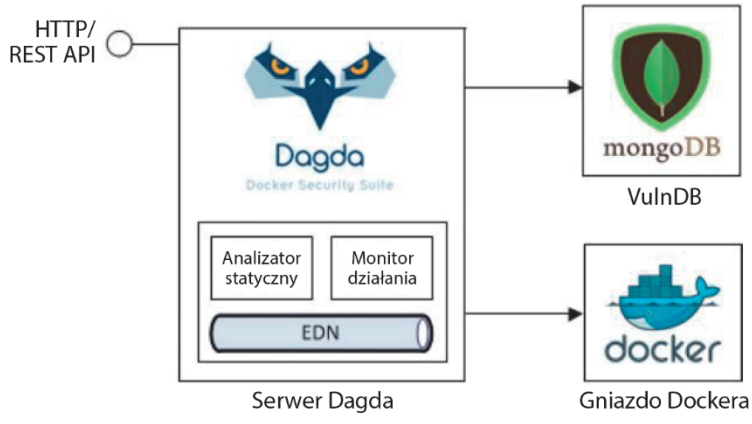


Rysunek 6.1. *Analiza warstw obrazu Dockera*

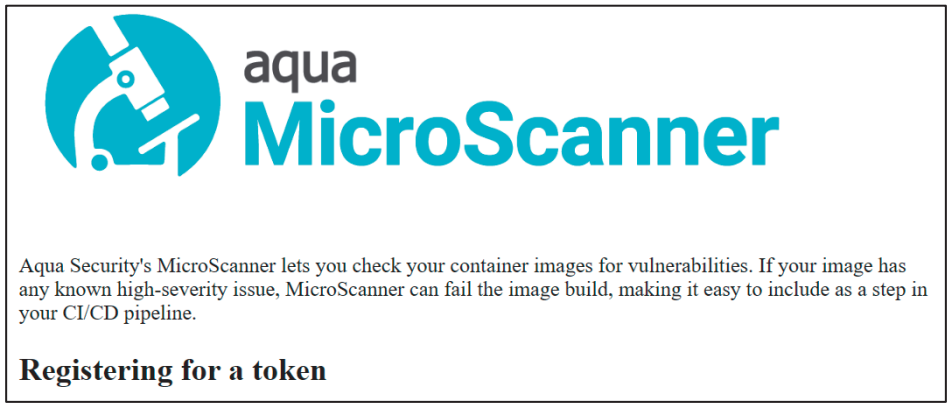
Rysunek 6.2. *Szczegółowe informacje na temat zagrożenia CVE obrazu Dockera*



Rysunek 6.3. *Narzędzie Dagda w repozytorium GitHub*



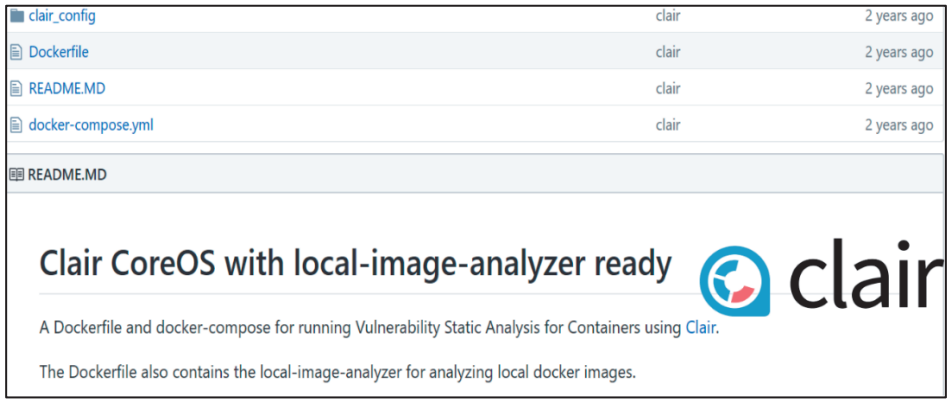
Rysunek 6.4. *Architektura narzędzia Dagda*



Rysunek 6.15. *Strona narzędzia MicroScanner, na której można uzyskać token*

Data Source	Data Collected	Format	License
Debian Security Bug Tracker	Debian 6, 7, 8, unstable namespaces	dpkg	Debian
Ubuntu CVE Tracker	Ubuntu 12.04, 12.10, 13.04, 14.04, 14.10, 15.04, 15.10, 16.04 namespaces	dpkg	GPLv2
Red Hat Security Data	CentOS 5, 6, 7 namespaces	rpm	CVRF
Oracle Linux Security Data	Oracle Linux 5, 6, 7 namespaces	rpm	CVRF
Alpine SecDB	Alpine 3.3, Alpine 3.4, Alpine 3.5 namespaces	apk	MIT
NIST NVD	Generic Vulnerability Metadata	N/A	Public Domain

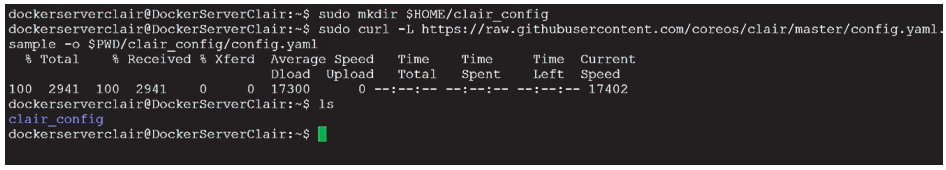
Rysunek 6.17. *Lista dystrybucji systemu Linux obsługiwanych przez narzędzie Clair*



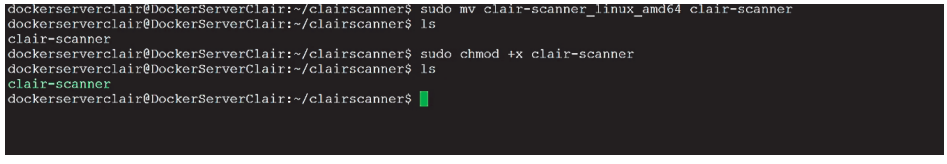
Rysunek 6.18. *Repozytorium GitHub zawierające skaner Clair*

```
version: '2'
services:
  postgres:
    container_name: clair_postgres
    image: postgres:latest
    environment:
      POSTGRES_PASSWORD: password

  clair:
    container_name: clair_clair
    image: hxquangnhat/clair:latest
    depends_on:
      - postgres
    ports:
      - "6060-6061:6060-6061"
    links:
      - postgres
    volumes:
      - /tmp:/tmp
      - ./clair_config:/config
      - /var/run/docker.sock:/var/run/docker.sock
    command: [-config, /config/config.yaml]
```

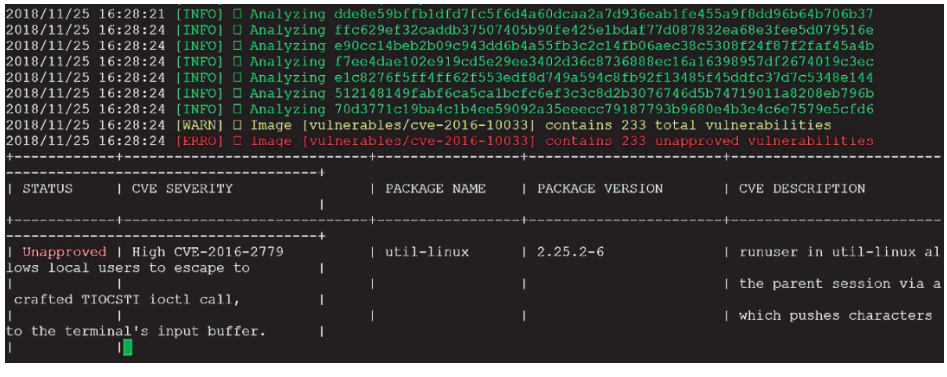
Rysunek 6.19. *Plik docker-compose.yml skanera Clair*



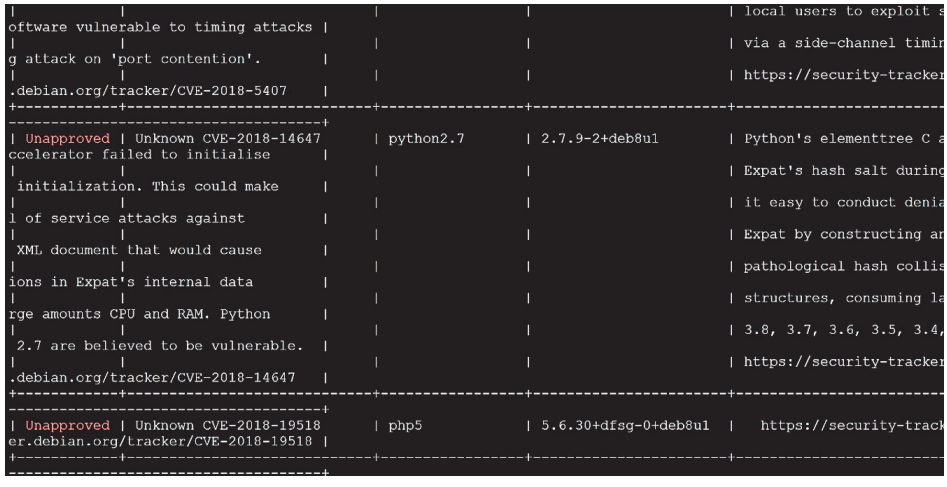
Rysunek 6.23. *Drugi krok instalacji*



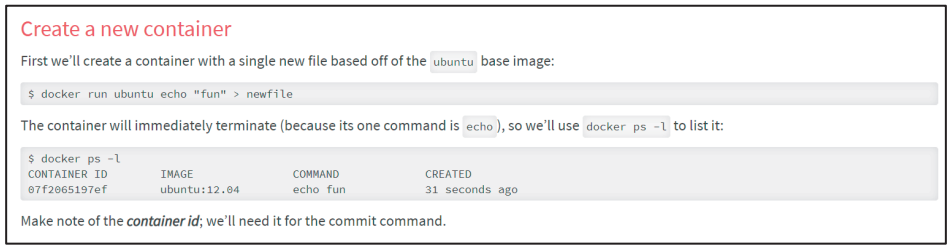
Rysunek 6.26. *Nadanie skanerowi Clair uprawnień do działania*



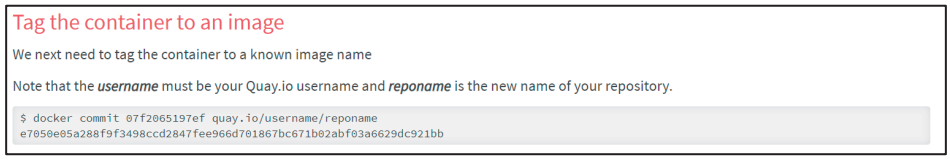
Rysunek 6.28. *Szósty krok instalacji*



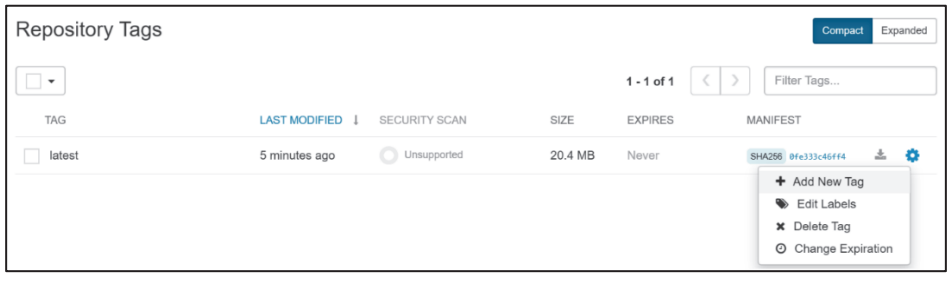
Rysunek 6.29. *Znalezione zagrożenia obrazu vulnerables/cve-2016-10033*



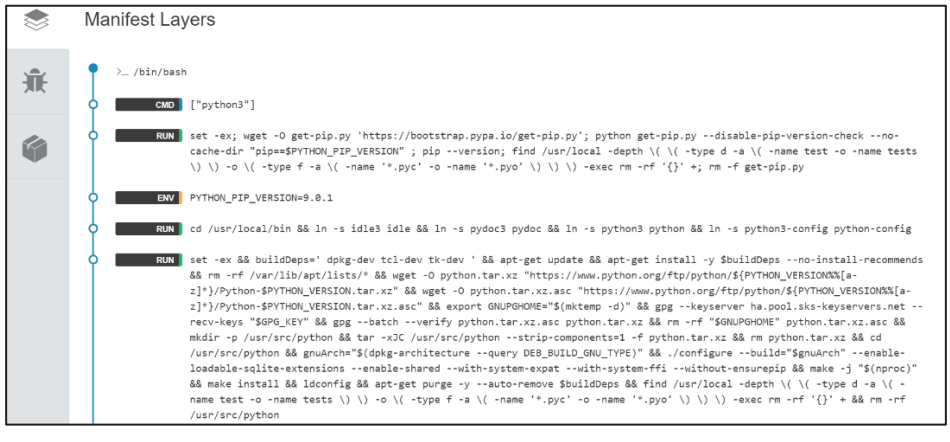
Rysunek 6.32. *Identyfikator zwrócony przez nowo utworzony kontener*



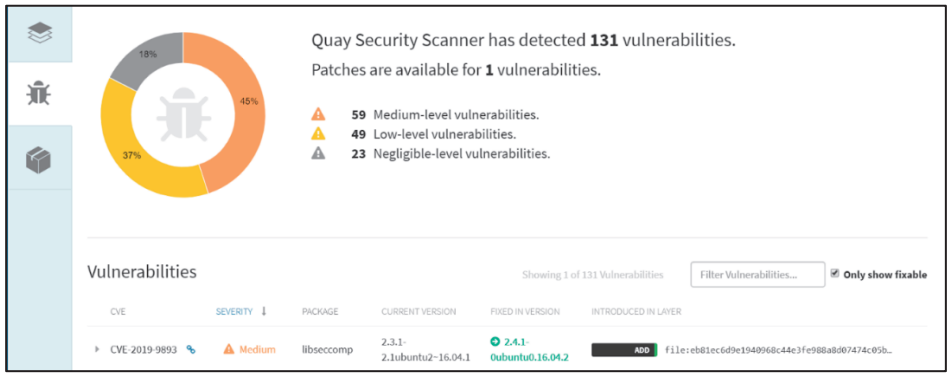
Rysunek 6.33. *Oznaczenie kontenera nazwą obrazu*



Rysunek 6.36. *Operacje na oznaczeniach obrazu*

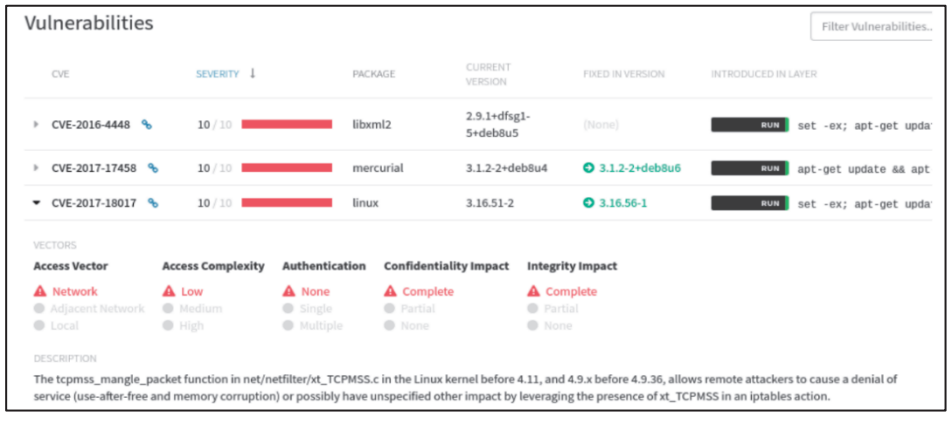


Rysunek 6.37. *Warstwy manifestu*



Rysunek 6.38. *Zagrożenia znalezione przez skaner Quay.io*

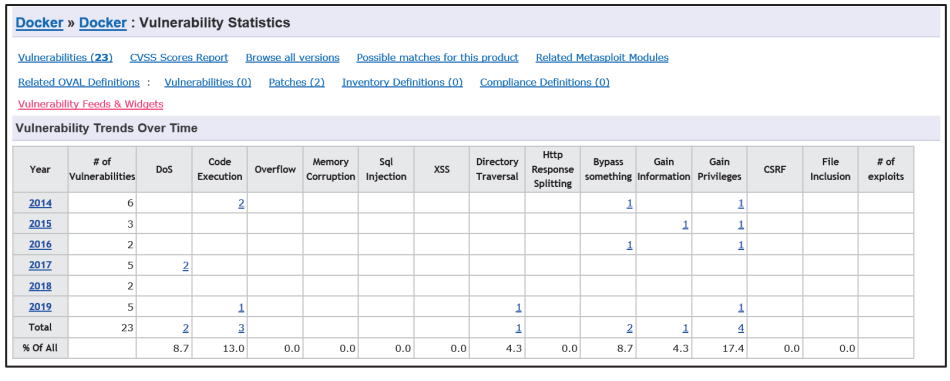
Rysunek 6.39. *Zagrożenia wraz z odpowiadającymi im numerami CVE*



Rysunek 6.40. *Szczegóły wykrytego zagrożenia*

```
$ git clone https://github.com/anchore/anchore-engine
Cloning into 'anchore-engine'...
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (82/82), done.
remote: Total 12804 (delta 51), reused 80 (delta 35), pack-reused 12683
Receiving objects: 100% (12804/12804), 20.92 MiB | 11.25 MiB/s, done.
Resolving deltas: 100% (8515/8515), done.
Checking out files: 100% (985/985), done.
[node1] (local) root@192.168.0.53 ~
$ cd anchore-engine
[node1] (local) root@192.168.0.53 ~/anchore-engine
$ LS
bash: LS: command not found
[node1] (local) root@192.168.0.53 ~/anchore-engine
$ ls
CHANGELOG.md            anchore_manager          requirements-test.txt
CONTRIBUTING.rst        conf                     requirements.txt
Dockerfile              docker-compose-dev.yaml  scripts
LICENSE                 docker-compose.yaml      setup.py
MANIFEST.in             docker-entrypoint.sh     test
```

Rysunek 6.42. *Pobranie kodu źródłowego silnika Anchore*



Docker » Docker : Vulnerability Statistics

Vulnerabilities (23) CVSS Scores Report Browse all versions Possible matches for this product Related Metasploit Modules

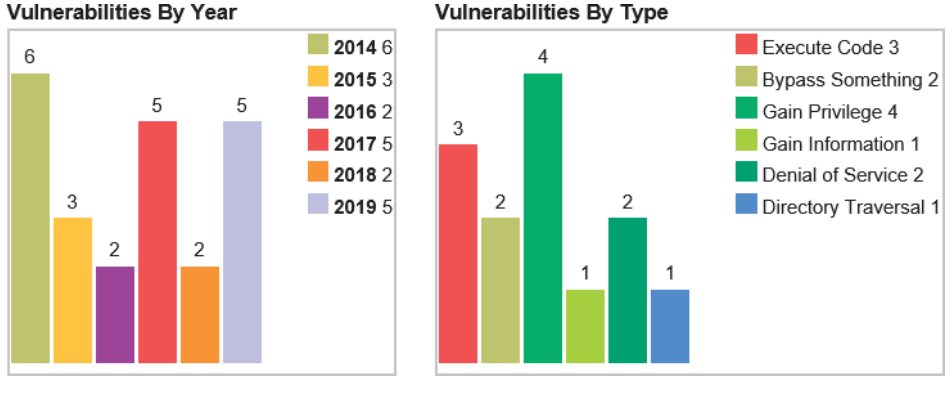
Related OVAL Definitions : Vulnerabilities (0) Patches (2) Inventory Definitions (0) Compliance Definitions (0)

Vulnerability Feeds & Widgets

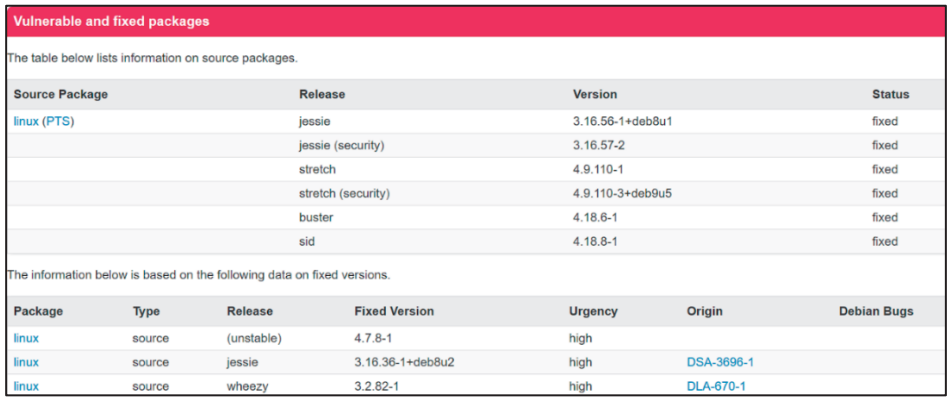
Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2014	6		2							1		1			
2015	3										1	1			
2016	2									1		1			
2017	5	2													
2018	2														
2019	5		1					1				1			
Total	23	2	3					1		2	1	4			
% Of All		8.7	13.0	0.0	0.0	0.0	0.0	4.3	0.0	8.7	4.3	17.4	0.0	0.0	

Rysunek 7.1. *Kategorie zagrożeń kontenerów Dockera*



Rysunek 7.2. *Najczęściej przeprowadzane ataki na kontenery Dockera*



Vulnerable and fixed packages

The table below lists information on source packages.

Source Package	Release	Version	Status
linux (PTS)	jessie	3.16.56-1+deb8u1	fixed
	jessie (security)	3.16.57-2	fixed
	stretch	4.9.110-1	fixed
	stretch (security)	4.9.110-3+deb9u5	fixed
	buster	4.18.6-1	fixed
	sid	4.18.8-1	fixed

The information below is based on the following data on fixed versions.

Package	Type	Release	Fixed Version	Urgency	Origin	Debian Bugs
linux	source	(unstable)	4.7.8-1	high		
linux	source	jessie	3.16.36-1+deb8u2	high	DSA-3696-1	
linux	source	wheezy	3.2.82-1	high	DLA-670-1	

Rysunek 7.7. *Wersje systemu Linux, których dotyczy zagrożenie Dirty COW*

Link	Usage	Description	Family
dirtyc0w.c	`./dirtyc0w file content`	Read-only write	/proc/self/mem
cowroot.c	`./cowroot`	SUID-based root	/proc/self/mem
dirtycow-mem.c	`./dirtycow-mem`	libc-based root	/proc/self/mem
pokemon.c	`./d file content`	Read-only write	PTRACE_POKEDATA
dirtycow.cr	`dirtycow --target --string --offset`	Read-only write	/proc/self/mem
dirtyc0w.c	`./dirtycow file content`	Read-only write (Android)	/proc/self/mem

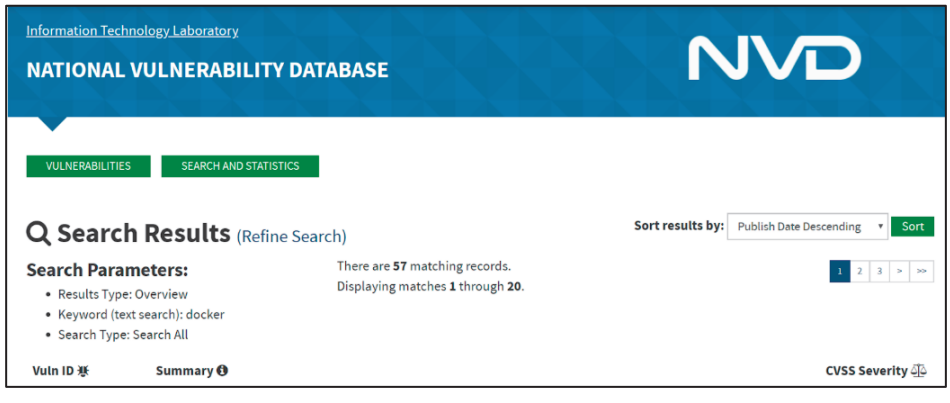
Rysunek 7.8. *Pliki wykorzystujące podatność Dirty COW*

Dockerfile	Minor changes (docker -> x86)	5 months ago
README.md	README.md	5 months ago
dirtyc0w.c	README.md	5 months ago
run.sh	README.md	5 months ago
safe_run.sh	README.md	5 months ago

Rysunek 7.9. *Repozytorium zawierające kontener do testowania zagrożenia Dirty COW*

```python
def is_layer_safe(layer_file):
    results = []
    try:
        tar = tarfile.open(layer_file, mode='r:gz')
    except tarfile.ReadError:
        tar = tarfile.open(layer_file, mode='r')

    while True:
        next_block = tar.next()
        if not next_block:
            break

        filename = next_block.name
        link_destination = next_block.linkname

        if not os.path.relpath(filename).find('..\\') or not os.path.relpath(filename).find('../'):
            results.append((filename, 0, layer_file))

        if link_destination:
            if link_destination[0] not in ['/', '\\']:
                full_path = os.path.dirname(filename) + "/" + link_destination
                if not os.path.relpath(full_path).find('..\\') or not os.path.relpath(full_path).find('../'):
                    results.append((full_path, 1, layer_file))
```

Rysunek 7.13. *Fragment skryptu w języku Python wykrywającego zagrożenie CVE-2018-8115*



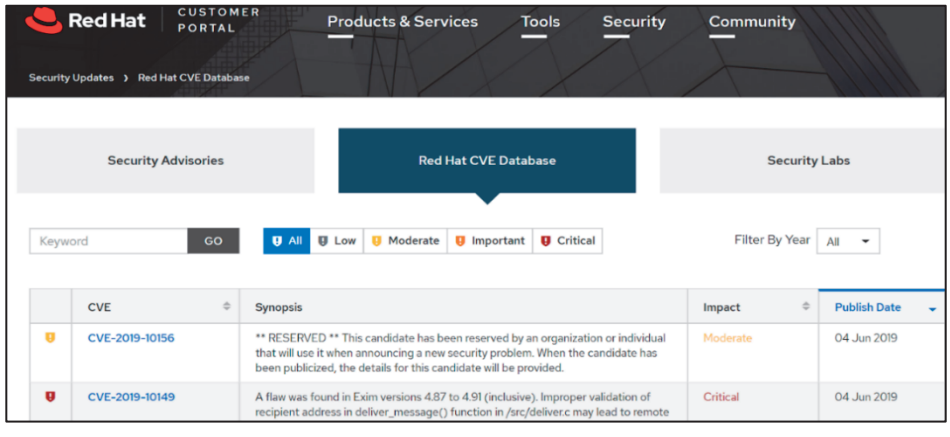
Rysunek 7.15. *Strona bazy NVD*

Vuln ID	Summary	CVSS Severity
CVE-2018-15664	In Docker through 18.06.1-ce-rc2, the API endpoints behind the 'docker cp' command are vulnerable to a symlink-exchange attack with Directory Traversal, giving attackers arbitrary read-write access to the host filesystem with root privileges, because daemon/archive.go does not do archive operations on a frozen filesystem (or from within a chroot). **Published:** May 23, 2019; 10:29:07 AM -04:00	V3: 8.7 HIGH V2: 9.3 HIGH
CVE-2019-5021	Versions of the Official Alpine Linux Docker images (since v3.3) contain a NULL password for the `root` user. This vulnerability appears to be the result of a regression introduced in December of 2015. Due to the nature of this issue, systems deployed using affected versions of the Alpine Linux container which utilize Linux PAM, or some other mechanism which uses the system shadow file as an authentication database, may accept a NULL password for the `root` user. **Published:** May 08, 2019; 01:29:01 PM -04:00	V3: 9.8 CRITICAL V2: 10.0 HIGH
CVE-2019-1003065	Jenkins CloudShare Docker-Machine Plugin stores credentials unencrypted in its global configuration file on the Jenkins master where they can be viewed by users with access to the master file system. **Published:** April 04, 2019; 12:29:00 PM -04:00	V3: 8.8 HIGH V2: 4.0 MEDIUM

Rysunek 7.16. *Zagrożenia w bazie NVD*

CVE	Package	Ubuntu 12.04 ESM (Precise Pangolin)	Ubuntu 14.04 ESM (Trusty Tahr)	Ubuntu 16.04 LTS (Xenial Xerus)	Ubuntu 18.04 LTS (Bionic Beaver)	Ubuntu 18.10 (Cosmic Cuttlefish)	Ubuntu 19.04 (Disco Dingo)	Ubuntu 19.10 (Eoan)	Links
CVE-2002-2439	gcc-4.6	needs-triage*	DNE	DNE	DNE	DNE	DNE	DNE	Mitre LP Debian
CVE-2005-4890	shadow	needed*	not-affected*	not-affected*	not-affected*	not-affected*	not-affected*	not-affected*	Mitre LP Debian
CVE-2008-7320	seahorse	DNE	DNE	needs-triage*	needs-triage*	needs-triage*	needs-triage*	needs-triage*	Mitre LP Debian
CVE-2009-1384	libpam-krb5	needed*	needed*	needed*	needed*	needed*	needed*	needed*	Mitre LP Debian
CVE-2009-5080	groff	needed*	needed*	needed*	needed*	needed*	needed*	needed*	Mitre LP Debian
CVE-2009-5155	eglibc	needed*	needed*	DNE	DNE	DNE	DNE	DNE	Mitre LP Debian
CVE-2009-5155	glibc	DNE	DNE	needed*	needed*	not-affected*	not-affected*	not-affected*	Mitre LP Debian

Rysunek 7.17. *Lista zagrożeń dystrybucji Ubuntu*



Rysunek 7.18. *Lista zagrożeń CVE firmy Red Hat*

CVE-ID

CVE-2019-5021 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Versions of the Official Alpine Linux Docker images (since v3.3) contain a NULL password for the `root` user. This vulnerability appears to be the result of a regression introduced in December of 2015. Due to the nature of this issue, systems deployed using affected versions of the Alpine Linux container which utilize Linux PAM, or some other mechanism which uses the system shadow file as an authentication database, may accept a NULL password for the `root` user.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BID:108288
- URL:http://www.securityfocus.com/bid/108288
- CONFIRM:https://security.netapp.com/advisory/ntap-20190510-0001/
- MISC:https://alpinelinux.org/posts/Docker-image-vulnerability-CVE-2019-5021.html
- MISC:https://talosintelligence.com/vulnerability_reports/TALOS-2019-0782
- SUSE:openSUSE-SU-2019:1495
- URL:http://lists.opensuse.org/opensuse-security-announce/2019-06/msg00004.html

Rysunek 7.19. *Zagrożenie systemu Linux Alpine*

Diffstat

-rwxr-xr-x scripts/genrootfs.sh 3

1 files changed, 3 insertions, 0 deletions

```
diff --git a/scripts/genrootfs.sh b/scripts/genrootfs.sh
index ac760e6e0d..5118027632 100755
--- a/scripts/genrootfs.sh
+++ b/scripts/genrootfs.sh
@@ -39,6 +39,9 @@ ${APK:-apk} fetch --keys-dir "$keys_dir" --no-cache \
         --repositories-file "$repositories_file" \
         --stdout --quiet alpine-base | tar -zx -C "$tmp" etc/

+# make sure root login is disabled
+sed -i -e 's/^root::/root:!:/' "$tmp"/etc/shadow
+
 branch=edge
 VERSION_ID=$(awk -F= '$1=="VERSION_ID" {print $2}'  "$tmp"/etc/os-release)
 case $VERSION_ID in
```

Rysunek 7.20. *Zagrożony skrypt obrazu Alpine*

Docker : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-1020014	415			2019-07-29	2019-08-19	2.1	None	Local	Low	Not required	Partial	None	None
docker-credential-helpers before 0.6.3 has a double free in the List functions.														
2	CVE-2019-16884	863		Bypass	2019-09-25	2019-10-07	5.0	None	Remote	Low	Not required	None	Partial	None
runc through 1.0.0-rc8, as used in Docker through 19.03.2-ce and other products, allows AppArmor restriction bypass because libcontainer/rootfs_linux.go incorrectly checks mount targets, and thus a malicious Docker image can mount over a /proc directory.														
3	CVE-2019-15752	264		+Priv	2019-08-28	2019-09-04	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
Docker Desktop Community Edition before 2.1.0.1 allows local users to gain privileges by placing a Trojan horse docker-credential-wincred.exe file in %PROGRAMDATA%\DockerDesktop \version-bin\ as a low-privilege user, and then waiting for an admin or service user to authenticate with Docker, restart Docker, or run 'docker login' to force the command.														
4	CVE-2019-14271	94			2019-07-29	2019-08-28	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
In Docker 19.03.x before 19.03.1 linked against the GNU C Library (aka glibc), code injection can occur when the nsswitch facility dynamically loads a library inside a chroot that contains the contents of the container.														
5	CVE-2019-13509	532			2019-07-18	2019-08-27	5.0	None	Remote	Low	Not required	Partial	None	None
In Docker CE and EE before 18.09.8 (as well as Docker EE before 17.06.2-ee-23 and 18.x before 18.03.1-ee-10), Docker Engine in debug mode may sometimes add secrets to the debug log. This applies to a scenario where docker stack deploy is run to redeploy a stack that includes (non external) secrets. It potentially applies to other API users of the stack API if they resend the secret.														
6	CVE-2019-5736	216		Exec Code	2019-02-11	2019-06-03	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
runc through 1.0-rc6, as used in Docker before 18.09.2 and other products, allows attackers to overwrite the host runc binary (and consequently obtain host root access) by leveraging the ability to execute a command as root within one of these types of containers: (1) a new container with an attacker-controlled image, or (2) an existing container, to which the attacker previously had write access, that can be attached with docker exec. This occurs because of file-descriptor mishandling, related to /proc/self/exe.														
7	CVE-2018-15664	362		Dir. Trav.	2019-05-23	2019-06-25	6.2	None	Local	High	Not required	Complete	Complete	Complete
In Docker through 18.06.1-ce-rc2, the API endpoints behind the 'docker cp' command are vulnerable to a symlink-exchange attack with Directory Traversal, giving attackers arbitrary read-write access to the host filesystem with root privileges, because daemon/archive.go does not do archive operations on a frozen filesystem (or from within a chroot).														
8	CVE-2018-15514	502			2018-08-31	2018-11-09	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
HandleRequestAsync in Docker for Windows before 18.06.0-ce-rc3-win68 (edge) and before 18.06.0-ce-win72 (stable) deserialized requests over the \\.\pipe\dockerBackend named pipe without verifying the validity of the deserialized .NET objects. This would allow a malicious user in the "docker-users" group (who may not otherwise have administrator access) to escalate to administrator privileges.														
9	CVE-2017-14992	20		DoS	2017-11-01	2017-11-22	4.3	None	Remote	Medium	Not required	None	None	Partial
Lack of content verification in Docker-CE (Also known as Moby) versions 1.12.6-0, 1.10.3, 17.03.0, 17.03.1, 17.03.2, 17.06.0, 17.06.1, 17.06.2, 17.09.0, and earlier allows a remote attacker to cause a Denial of Service via a crafted image layer payload, aka gzip bombing.														
10	CVE-2017-11468	770		DoS	2017-07-20	2019-10-02	5.0	None	Remote	Low	Not required	None	None	Partial
Docker Registry before 2.6.2 in Docker Distribution does not properly restrict the amount of content accepted from a user, which allows remote attackers to cause a denial of service (memory consumption) via the manifest endpoint.														
11	CVE-2017-7297				2017-03-28	2019-10-02	6.5	None	Remote	Low	Single system	Partial	Partial	Partial

Rysunek 7.23. *Baza zagrożeń CVE platformy Docker*

CVE	Vendors	Products	Updated	CVSS
CVE-2019-14271	1 Docker	1 Docker	2019-08-28	7.5
In Docker 19.03.x before 19.03.1 linked against the GNU C Library (aka glibc), code injection can occur when the nsswitch facility dynamically loads a library inside a chroot that contains the contents of the container.				
CVE-2019-13509	1 Docker	1 Docker	2019-08-27	5.0
In Docker CE and EE before 18.09.8 (as well as Docker EE before 17.06.2-ee-23 and 18.x before 18.03.1-ee-10), Docker Engine in debug mode may sometimes add secrets to the debug log. This applies to a scenario where docker stack deploy is run to...				
CVE-2019-1020014	1 Docker	1 Credential Helpers	2019-08-19	2.1
docker-credential-helpers before 0.6.3 has a double free in the List functions.				
CVE-2018-15664	1 Docker	1 Docker	2019-06-25	6.2
In Docker through 18.06.1-ce-rc2, the API endpoints behind the 'docker cp' command are vulnerable to a symlink-exchange attack with Directory Traversal, giving attackers arbitrary read-write access to the host filesystem with root privileges,...				
CVE-2019-5736	10 Docker, Google, Linuxcontainers *and 7 more*	12 Docker, Kubernetes Engine, Lxc *and 9 more*	2019-06-03	9.3

Rysunek 7.24. *Baza saucs zawierająca zagrożenia CVE*

```
~$ kubectl logs kube-bench-node
[INFO] 2 Worker Node Security Configuration
[INFO] 2.1 Kubelet
[FAIL] 2.1.1 Ensure that the --allow-privileged argument is set to false (Scored)
[PASS] 2.1.2 Ensure that the --anonymous-auth argument is set to false (Scored)
[PASS] 2.1.3 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Scored)
[PASS] 2.1.4 Ensure that the --client-ca-file argument is set as appropriate (Scored)
[PASS] 2.1.5 Ensure that the --read-only-port argument is set to 0 (Scored)
[FAIL] 2.1.6 Ensure that the --streaming-connection-idle-timeout argument is not set to 0 (Scored)
[FAIL] 2.1.7 Ensure that the --protect-kernel-defaults argument is set to true (Scored)
[FAIL] 2.1.8 Ensure that the --make-iptables-util-chains argument is set to true (Scored)
[FAIL] 2.1.9 Ensure that the --keep-terminated-pod-volumes argument is set to false (Scored)
[FAIL] 2.1.10 Ensure that the --hostname-override argument is not set (Scored)
[FAIL] 2.1.11 Ensure that the --event-qps argument is set to 0 (Scored)
[PASS] 2.1.12 Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate (Scored)
[PASS] 2.1.13 Ensure that the --cadvisor-port argument is set to 0 (Scored)
[FAIL] 2.1.14 Ensure that the RotateKubeletClientCertificate argument is set to true
[FAIL] 2.1.15 Ensure that the RotateKubeletServerCertificate argument is set to true
[INFO] 2.2 Configuration Files
[FAIL] 2.2.1 Ensure that the kubelet.conf file permissions are set to 644 or more restrictive (Scored)
[FAIL] 2.2.2 Ensure that the kubelet.conf file ownership is set to root:root (Scored)
[FAIL] 2.2.3 Ensure that the kubelet service file permissions are set to 644 or more restrictive (Scored)
[FAIL] 2.2.4 2.2.4 Ensure that the kubelet service file ownership is set to root:root (Scored)
[FAIL] 2.2.5 Ensure that the proxy kubeconfig file permissions are set to 644 or more restrictive (Scored)
[FAIL] 2.2.6 Ensure that the proxy kubeconfig file ownership is set to root:root (Scored)
[WARN] 2.2.7 Ensure that the certificate authorities file permissions are set to 644 or more restrictive (Scored)
[WARN] 2.2.8 Ensure that the client certificate authorities file ownership is set to root:root
```

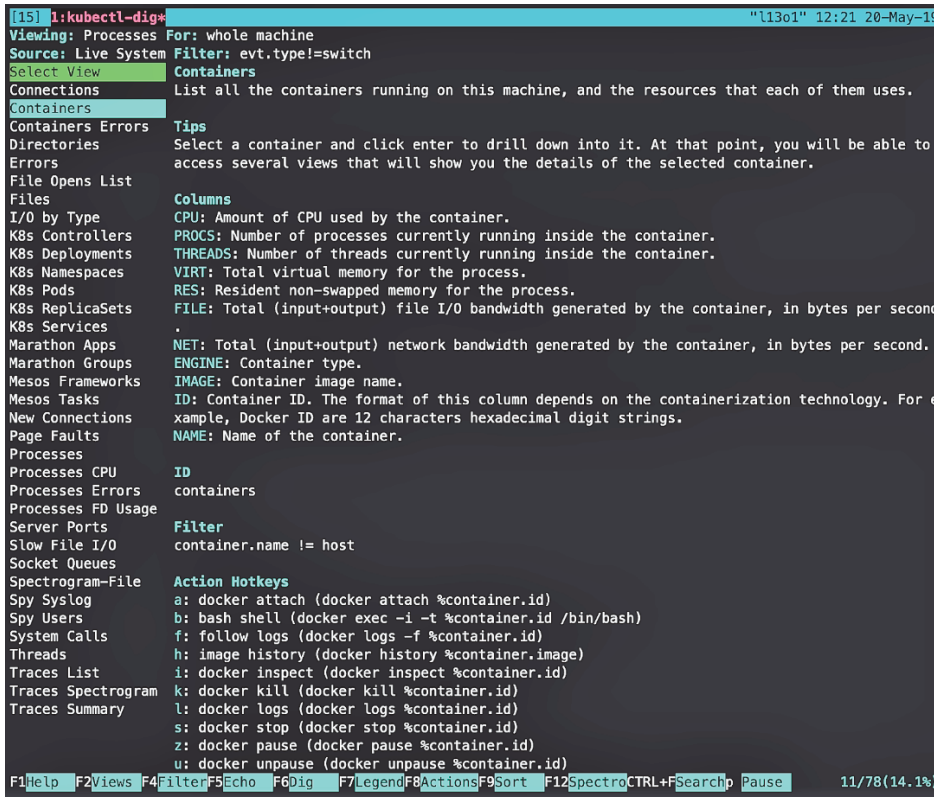
Rysunek 8.2. *Uruchomienie narzędzia Kube Bench w węźle roboczym*

```
[root@master ~]# docker run --pid=host -t aquasec/kube-bench:latest master --version 1.8
[INFO] 1 Master Node Security Configuration
[INFO] 1.1 API Server
[FAIL] 1.1.1 Ensure that the --anonymous-auth argument is set to false (Scored)
[PASS] 1.1.2 Ensure that the --basic-auth-file argument is not set (Scored)
[PASS] 1.1.3 Ensure that the --insecure-allow-any-token argument is not set (Scored)
[PASS] 1.1.4 Ensure that the --kubelet-https argument is set to true (Scored)
[PASS] 1.1.5 Ensure that the --insecure-bind-address argument is not set (Scored)
[PASS] 1.1.6 Ensure that the --insecure-port argument is set to 0 (Scored)
[PASS] 1.1.7 Ensure that the --secure-port argument is not set to 0 (Scored)
[FAIL] 1.1.8 Ensure that the --profiling argument is set to false (Scored)
[FAIL] 1.1.9 Ensure that the --repair-malformed-updates argument is set to false (Scored)
[FAIL] 1.1.10 Ensure that the admission control policy is not set to AlwaysAdmit (Scored)
[FAIL] 1.1.11 Ensure that the admission control policy is set to AlwaysPullImages (Scored)
[FAIL] 1.1.12 Ensure that the admission control policy is set to DenyEscalatingExec (Scored)
[FAIL] 1.1.13 Ensure that the admission control policy is set to SecurityContextDeny (Scored)
[FAIL] 1.1.14 Ensure that the admission control policy is set to NamespaceLifecycle (Scored)
[FAIL] 1.1.15 Ensure that the --audit-log-path argument is set as appropriate (Scored)
[FAIL] 1.1.16 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Scored)
```

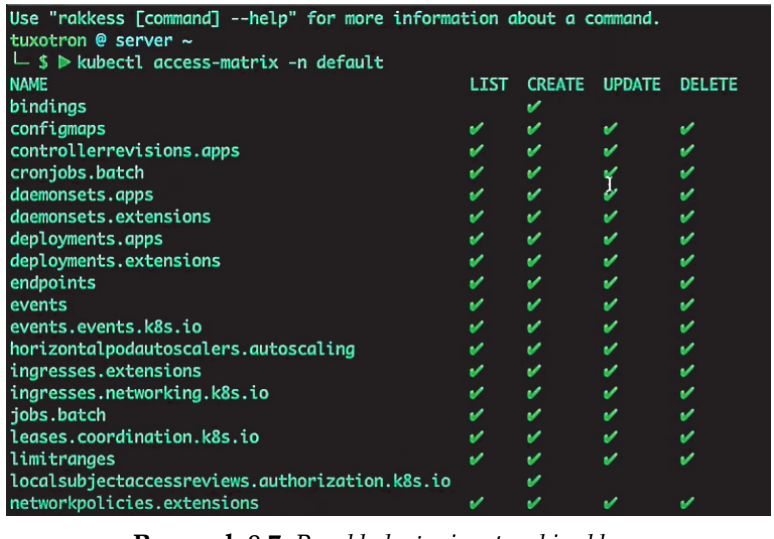
Rysunek 8.3. *Uruchomienie narzędzia w węźle głównym*

```
→  ~ kubectl get pods
NAME                                READY     STATUS    RESTARTS   AGE
hello-minikube-7c77b68cff-qhq5b   1/1       Running   0          29m
→  ~ kubectl plugin sniff hello-minikube-7c77b68cff-qhq5b
[+] Sniffing on pod: hello-minikube-7c77b68cff-qhq5b container:  namespace:
[+] Verifying pod status
NAME                                READY     STATUS    RESTARTS   AGE
hello-minikube-7c77b68cff-qhq5b   1/1       Running   0          29m
[+] checking if tcpdump already exist
-rwxrwxr-x 1 1000 1000 2700408 Jun 22 14:08 /static-tcpdump
[+] static tcpdump is already installed on container!
[+] Starting remote sniffing!
```

Rysunek 8.5. *Weryfikacja stanu podów za pomocą wtyczki ksniff*



Rysunek 8.6. *Przykład użycia wtyczki kubectl-dig*



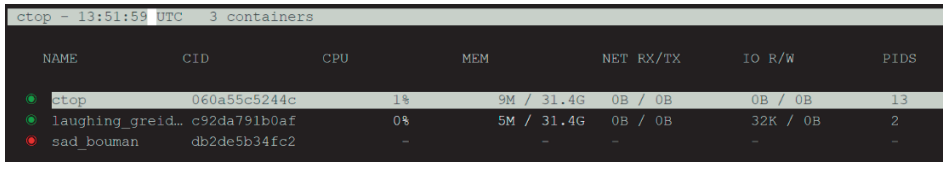
Rysunek 8.7. *Przykład użycia wtyczki rakkess*

```
$ docker network create my-network
79896668b483d38ca83642f3197afdce9188116d079b70203c065c053dfe8980
[node1] (local) root@192.168.0.28 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
80302a34d531        bridge              bridge              local
21276d934921        host                host                local
79896668b483        my-network          bridge              local
03bf600bbfec        none                null                local
```

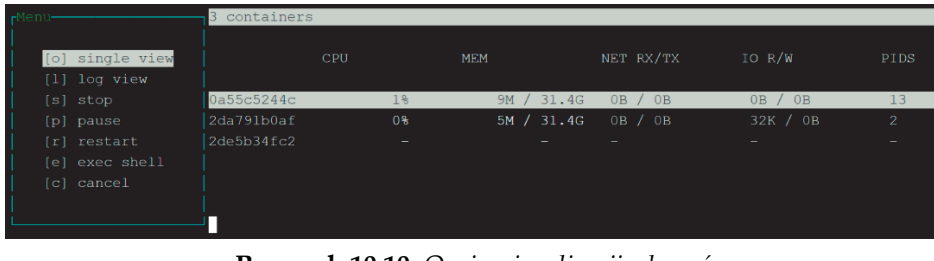
Rysunek 9.16. *Tworzenie sieci*

```
root@a05699ed73ab:/# set | grep -i elasticSearch
ELASTICSEARCH_ENV_CA_CERTIFICATES_JAVA_VERSION=20140324
ELASTICSEARCH_ENV_ELASTICSEARCH_VERSION=2.2.2
ELASTICSEARCH_ENV_GOSU_VERSION=1.7
ELASTICSEARCH_ENV_JAVA_DEBIAN_VERSION=8u111-b14-2~bpo8+1
ELASTICSEARCH_ENV_JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
ELASTICSEARCH_ENV_JAVA_VERSION=8u111
ELASTICSEARCH_ENV_LANG=C.UTF-8
ELASTICSEARCH_NAME=/ubuntu/elasticSearch
ELASTICSEARCH_PORT=tcp://172.17.0.2:9200
ELASTICSEARCH_PORT_9200_TCP=tcp://172.17.0.2:9200
ELASTICSEARCH_PORT_9200_TCP_ADDR=172.17.0.2
ELASTICSEARCH_PORT_9200_TCP_PORT=9200
ELASTICSEARCH_PORT_9200_TCP_PROTO=tcp
ELASTICSEARCH_PORT_9300_TCP=tcp://172.17.0.2:9300
ELASTICSEARCH_PORT_9300_TCP_ADDR=172.17.0.2
ELASTICSEARCH_PORT_9300_TCP_PORT=9300
ELASTICSEARCH_PORT_9300_TCP_PROTO=tcp
```

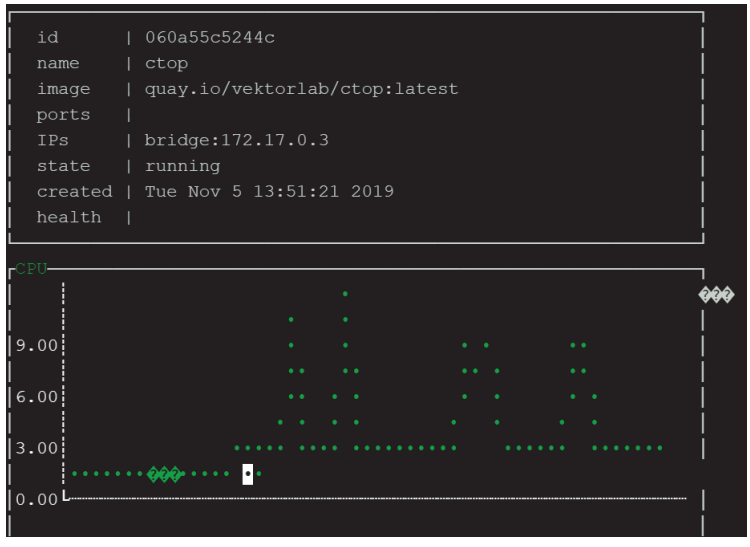
Rysunek 9.19. *Zmienne środowiskowe kontenera Ubuntu zawierające informacje o kontenerze Elasticsearch*



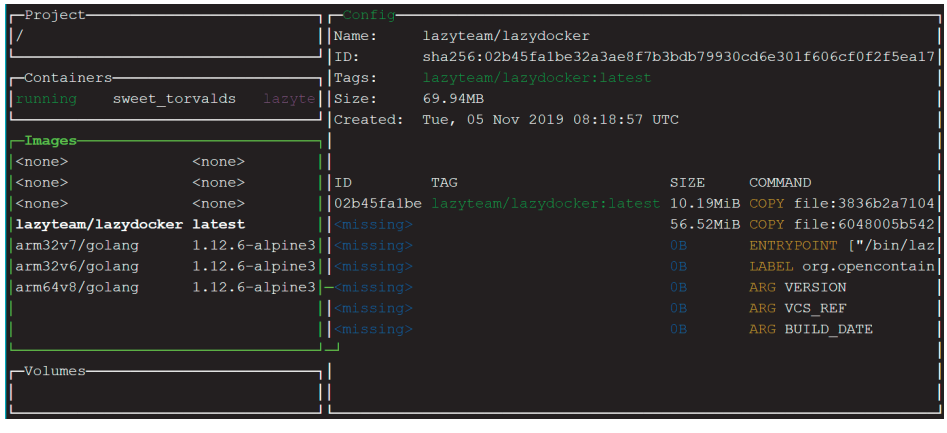
Rysunek 10.9. *Przykład użycia narzędzia ctop*



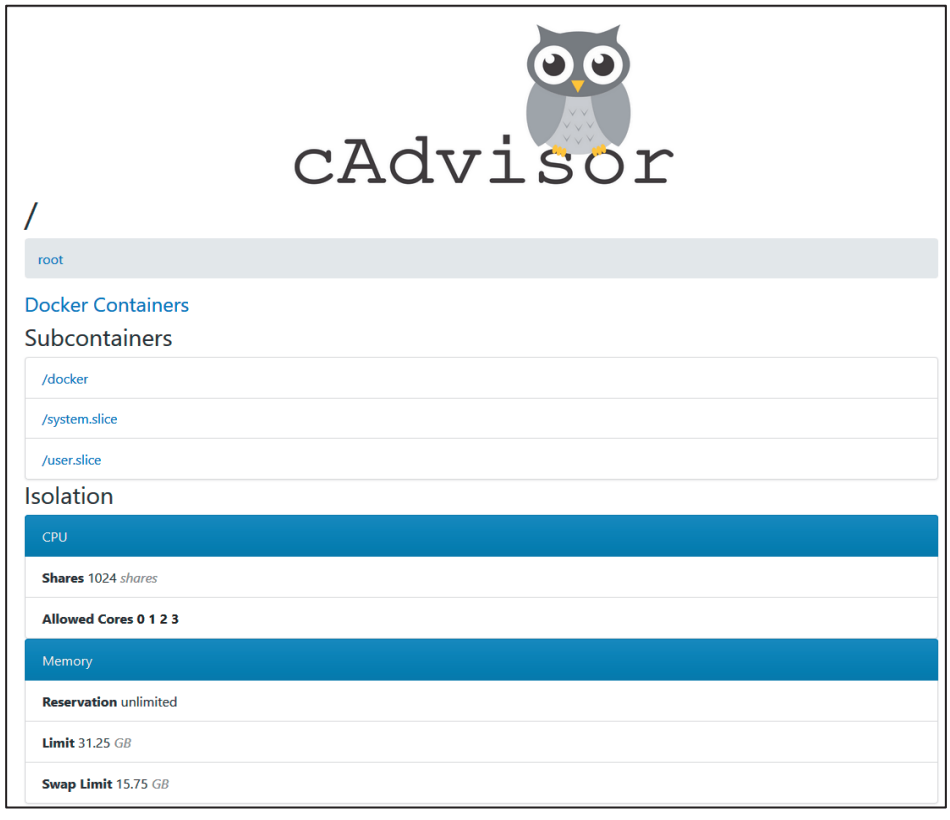
Rysunek 10.10. *Opcje wizualizacji zdarzeń*



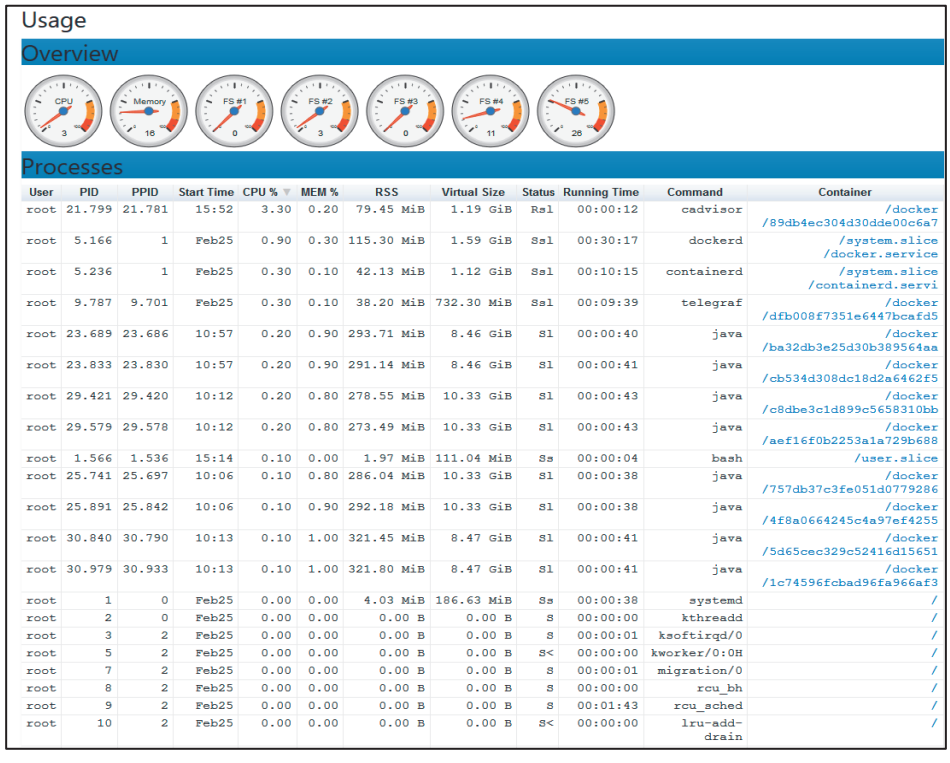
Rysunek 10.11. *Szczegółowe informacje o wykorzystaniu procesora w kontenerze*



Rysunek 10.12. *Szczegółowe informacje o kontenerze uzyskane za pomocą narzędzia LazyDocker*

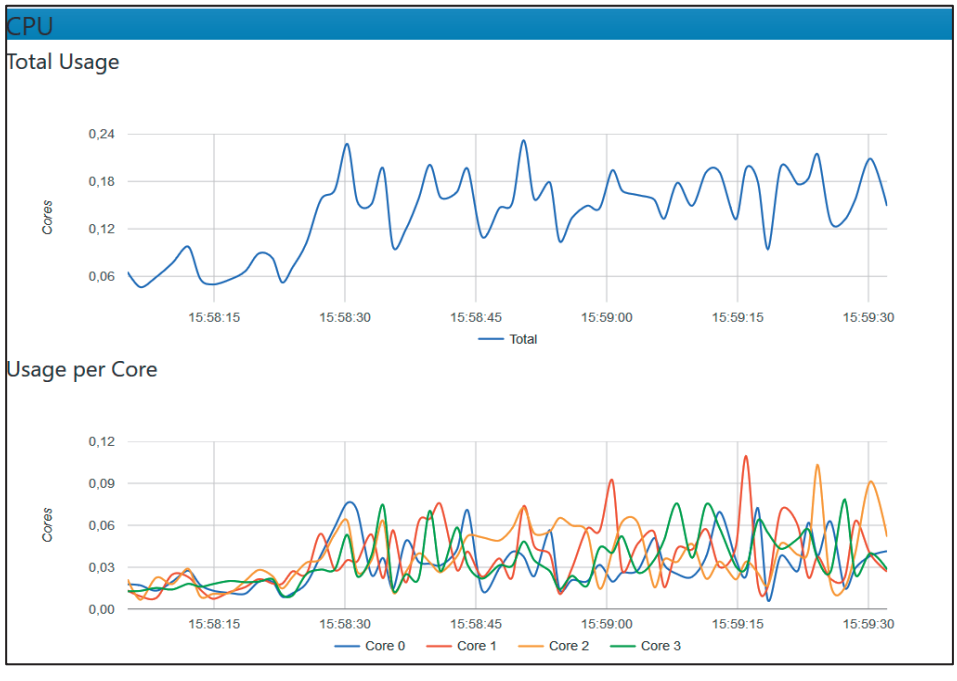


Rysunek 10.14. *Informacje o uruchomionych kontenerach*

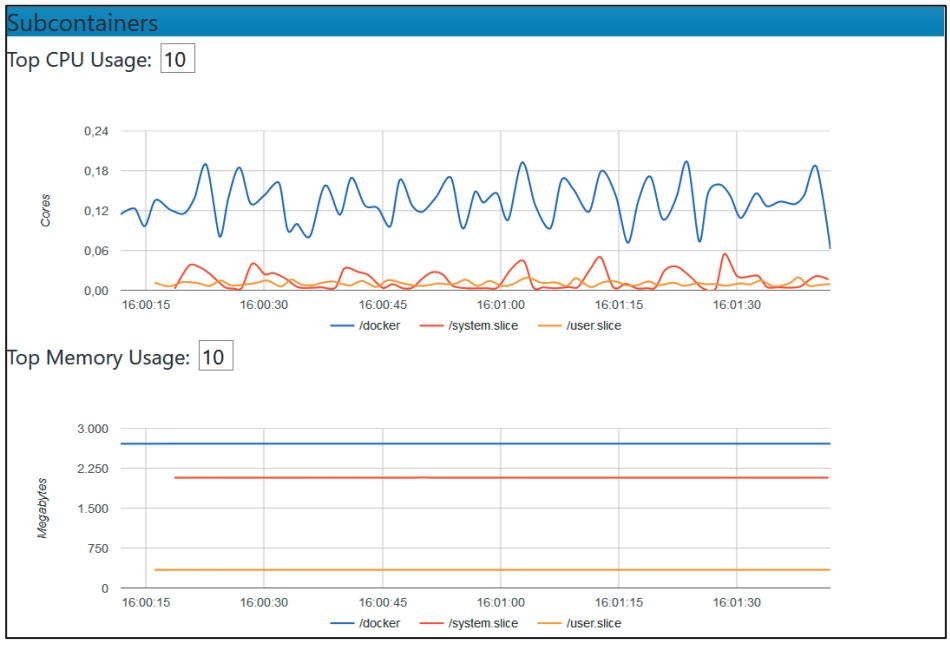


User	PID	PPID	Start Time	CPU %	MEM %	RSS	Virtual Size	Status	Running Time	Command	Container
root	21.799	21.781	15:52	3.30	0.20	79.45 MiB	1.19 GiB	Rsl	00:00:12	cadvisor	/docker /89db4ec304d30dde00c6a7
root	5.166	1	Feb25	0.90	0.30	115.30 MiB	1.59 GiB	Ssl	00:30:17	dockerd	/system.slice /docker.service
root	5.236	1	Feb25	0.30	0.10	42.13 MiB	1.12 GiB	Ssl	00:10:15	containerd	/system.slice /containerd.servi
root	9.787	9.701	Feb25	0.30	0.10	38.20 MiB	732.30 MiB	Ssl	00:09:39	telegraf	/docker /dfb008f7351e6447bcafd5
root	23.689	23.686	10:57	0.20	0.90	293.71 MiB	8.46 GiB	Sl	00:00:40	java	/docker /ba32db3e25d30b389564aa
root	23.833	23.830	10:57	0.20	0.90	291.14 MiB	8.46 GiB	Sl	00:00:41	java	/docker /cb534d308dc18d2a6462f5
root	29.421	29.420	10:12	0.20	0.80	278.55 MiB	10.33 GiB	Sl	00:00:43	java	/docker /c8dbe3c1d899c5658310bb
root	29.579	29.578	10:12	0.20	0.80	273.49 MiB	10.33 GiB	Sl	00:00:43	java	/docker /aef16f0b2253a1a729b688
root	1.566	1.536	15:14	0.10	0.00	1.97 MiB	111.04 MiB	Ss	00:00:04	bash	/user.slice
root	25.741	25.697	10:06	0.10	0.80	286.04 MiB	10.33 GiB	Sl	00:00:38	java	/docker /757db37c3fe051d0779286
root	25.891	25.842	10:06	0.10	0.90	292.18 MiB	10.33 GiB	Sl	00:00:38	java	/docker /4f8a0664245c4a97ef4255
root	30.840	30.790	10:13	0.10	1.00	321.45 MiB	8.47 GiB	Sl	00:00:41	java	/docker /5d65cec329c52416d15651
root	30.979	30.933	10:13	0.10	1.00	321.80 MiB	8.47 GiB	Sl	00:00:41	java	/docker /1c74596fcbad96fa966af3
root	1	0	Feb25	0.00	0.00	4.03 MiB	186.63 MiB	Ss	00:00:38	systemd	/
root	2	0	Feb25	0.00	0.00	0.00 B	0.00 B	S	00:00:00	kthreadd	/
root	3	2	Feb25	0.00	0.00	0.00 B	0.00 B	S	00:00:01	ksoftirqd/0	/
root	5	2	Feb25	0.00	0.00	0.00 B	0.00 B	S<	00:00:00	kworker/0:0H	/
root	7	2	Feb25	0.00	0.00	0.00 B	0.00 B	S	00:00:01	migration/0	/
root	8	2	Feb25	0.00	0.00	0.00 B	0.00 B	S	00:00:00	rcu_bh	/
root	9	2	Feb25	0.00	0.00	0.00 B	0.00 B	S	00:01:43	rcu_sched	/
root	10	2	Feb25	0.00	0.00	0.00 B	0.00 B	S<	00:00:00	lru-add-drain	/

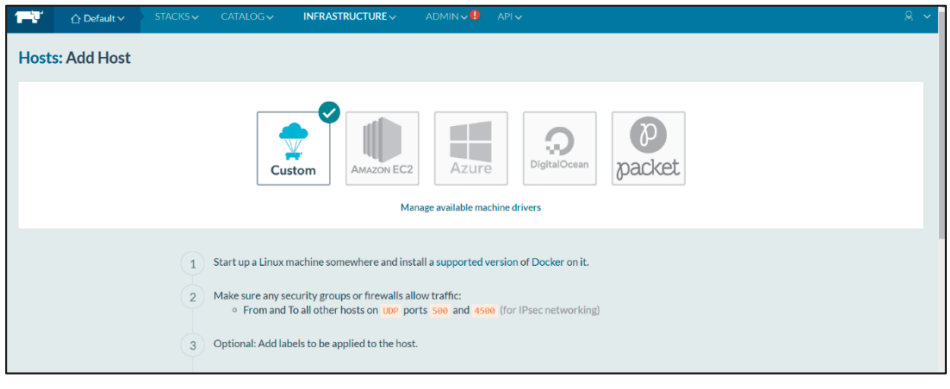
Rysunek 10.15. *Informacje o procesach*



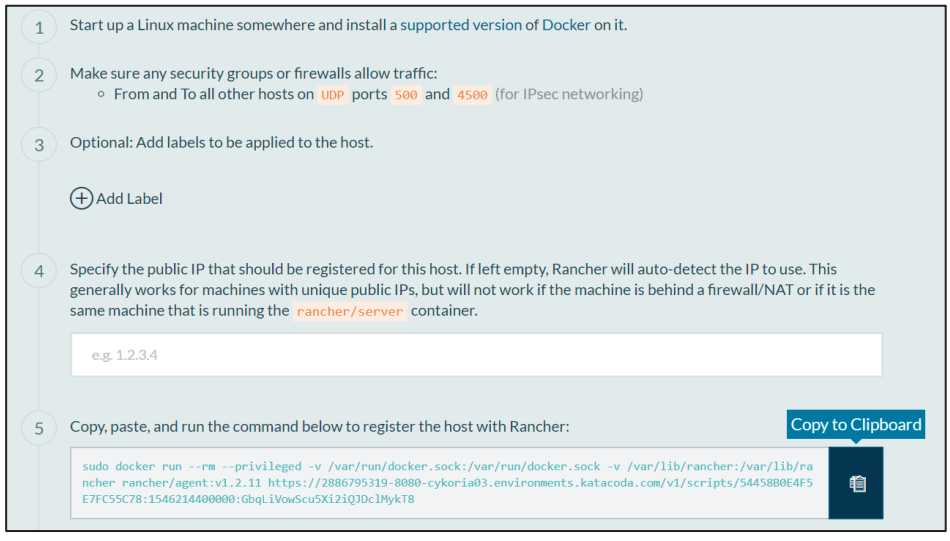
Rysunek 10.16. *Wykorzystanie rdzeni procesora*



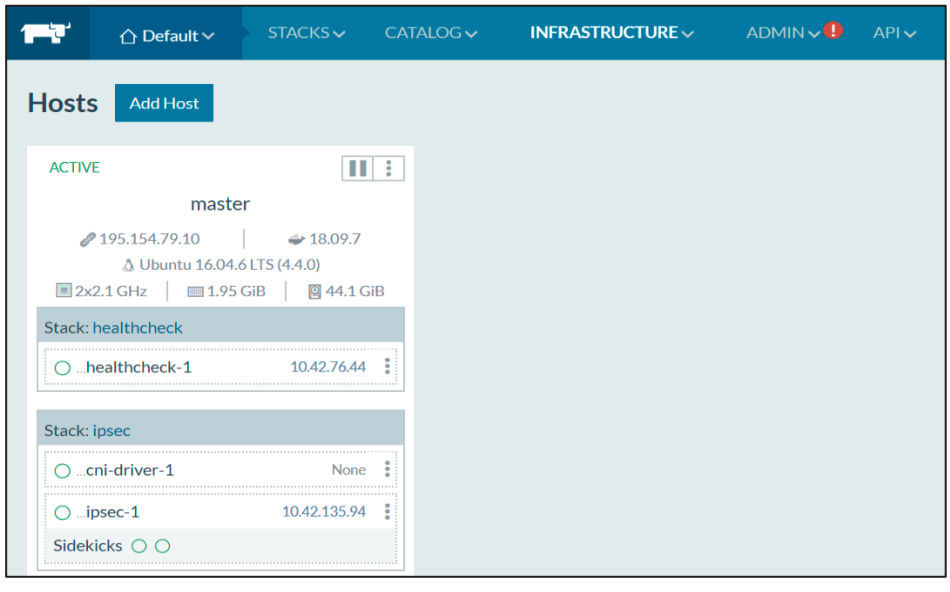
Rysunek 10.17. *Wykorzystanie procesora i pamięci przez poszczególne kontenery*



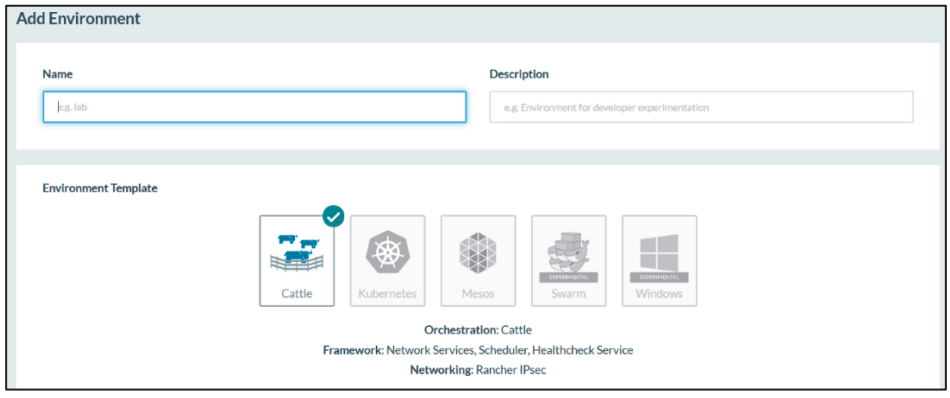
Rysunek 11.1. *Dodawanie hosta za pomocą narzędzia Rancher*



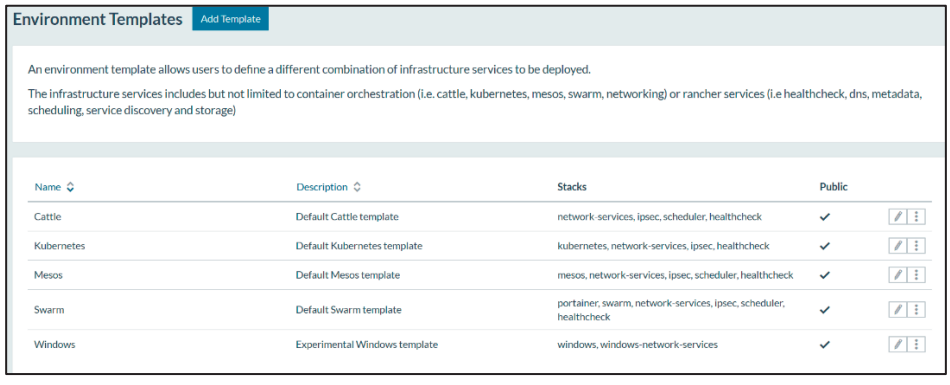
Rysunek 11.2. *Rejestrowanie hosta za pomocą narzędzia Rancher*



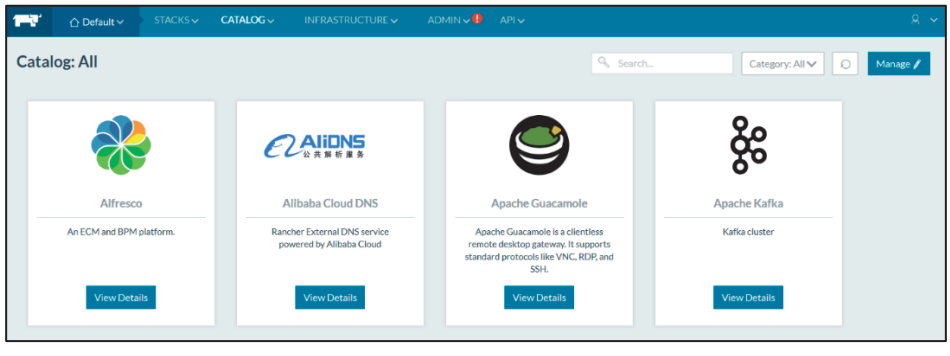
Rysunek 11.3. *Informacje o hoście w interfejsie narzędzia Rancher*



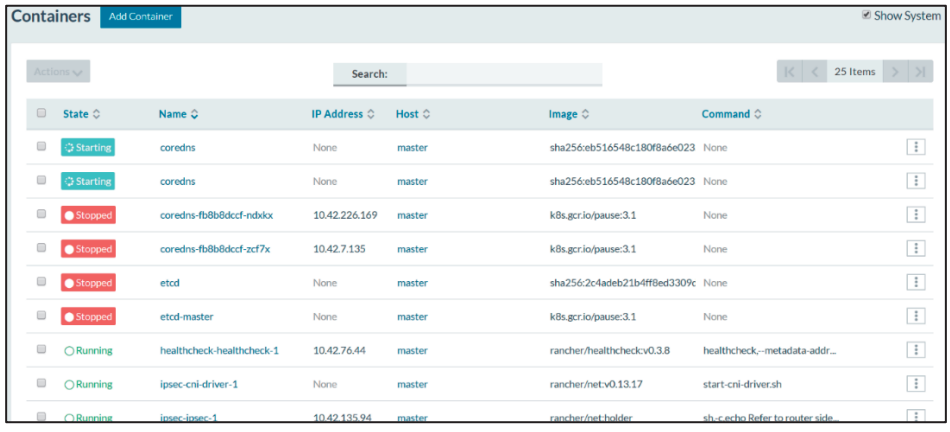
Rysunek 11.4. *Szablon środowiska w narzędziu Rancher*



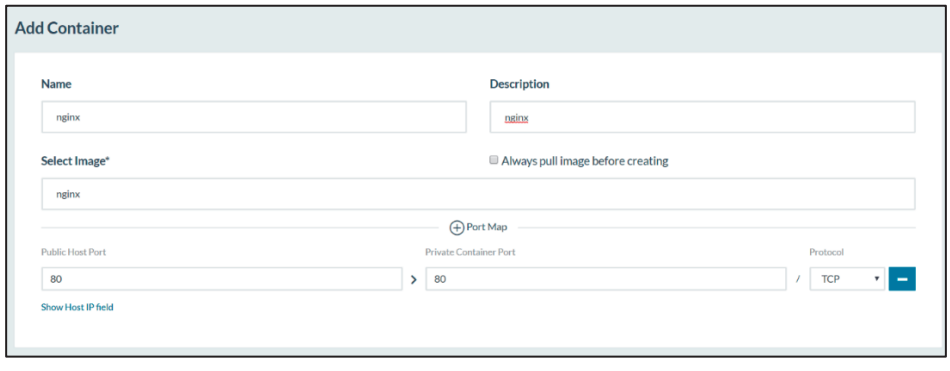
Rysunek 11.5. *Szablony środowiska w narzędziu Rancher*



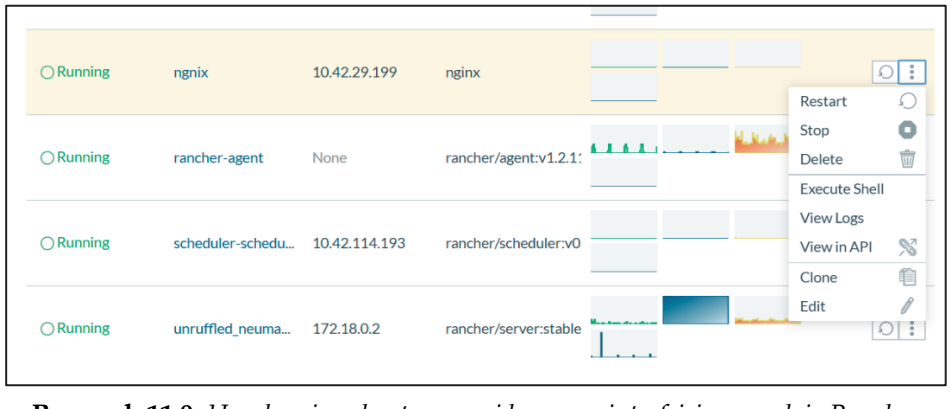
Rysunek 11.6. *Katalog aplikacji w interfejsie narzędzia Rancher*



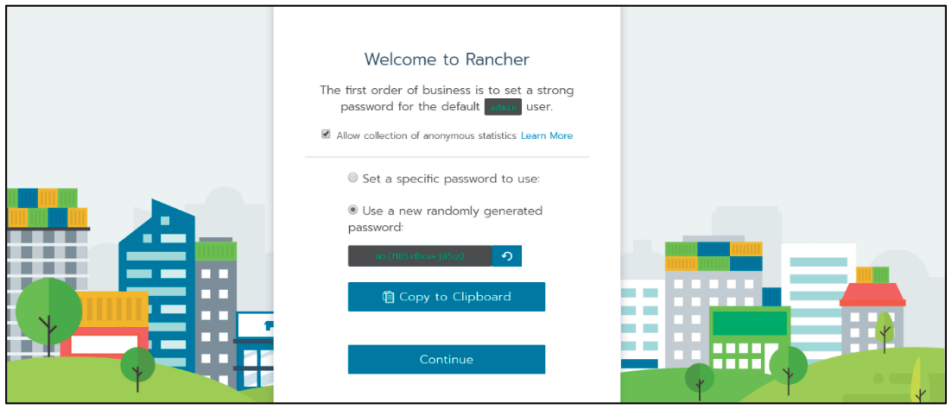
Rysunek 11.7. *Panel kontenerów w narzędziu Rancher*



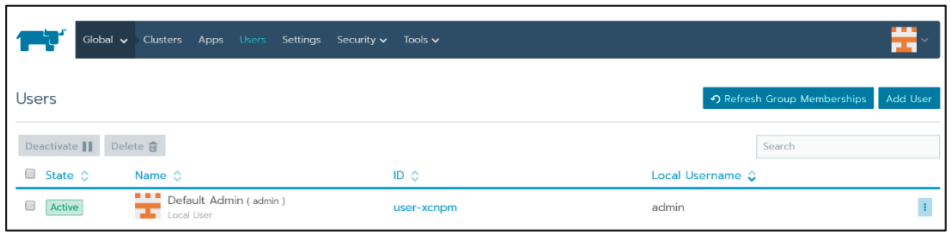
Rysunek 11.8. *Dodawanie kontenera za pomocą narzędzia Rancher*



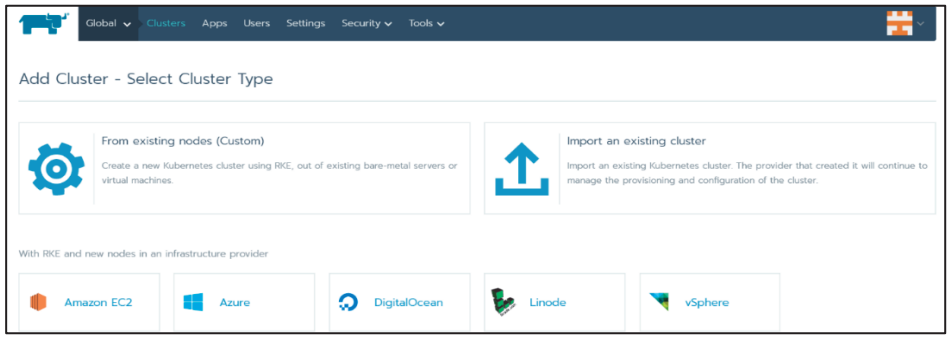
Rysunek 11.9. *Uruchomione kontenery widoczne w interfejsie narzędzia Rancher*



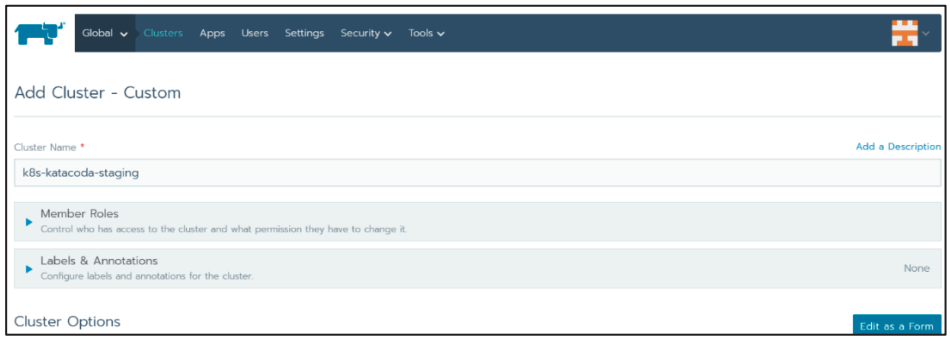
Rysunek 11.10. *Strona powitalna narzędzia Rancher*



Rysunek 11.11. *Sekcja Users w narzędziu Rancher*



Rysunek 11.12. *Tworzenie klastra*

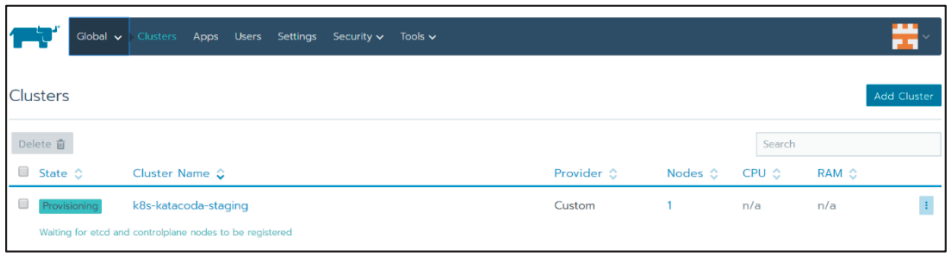


Rysunek 11.13. *Wybór typu klastra*

Rysunek 11.14. *Plik konfiguracyjny klastra*

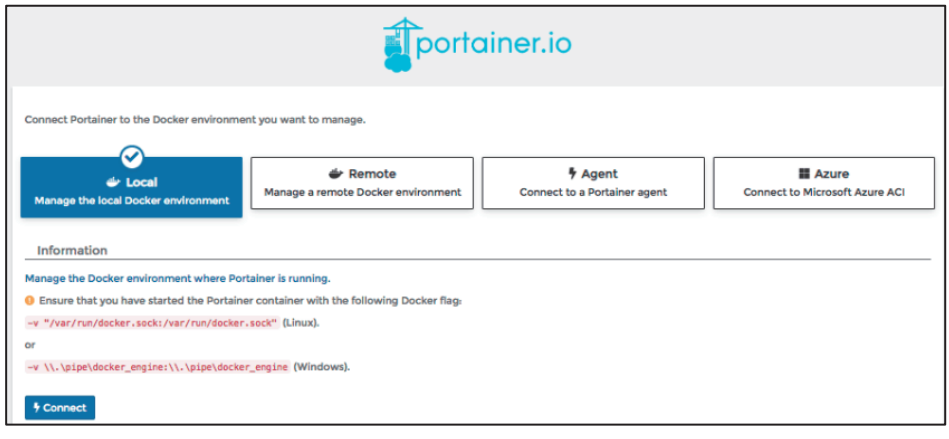
Rysunek 11.15. *Konfiguracja węzła klastra*

Rysunek 11.16. *Rejestracja głównego węzła klastra*

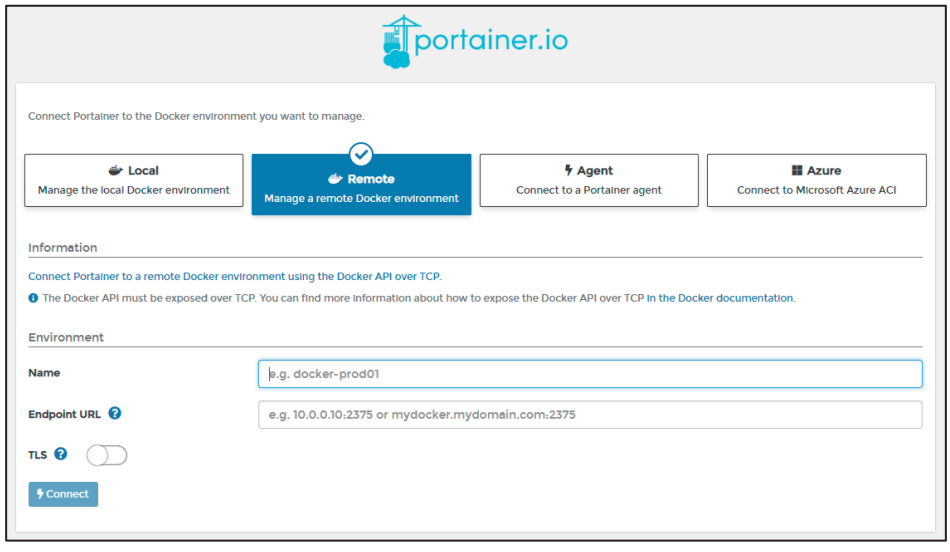


Rysunek 11.17. *Stan klastra w interfejsie narzędzia Rancher*

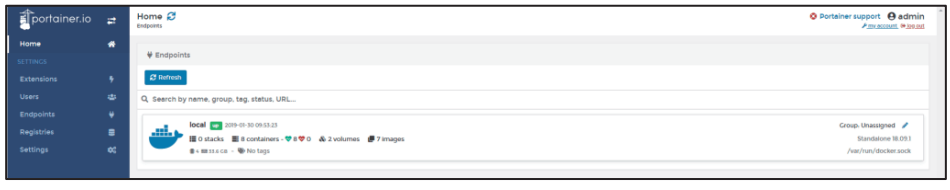
Rysunek 11.19. *Definiowanie konta administratora*



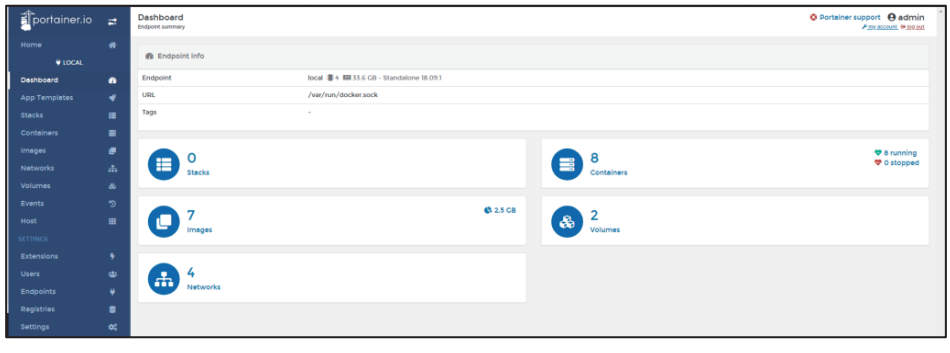
Rysunek 11.20. *Lokalna instalacja narzędzia Portainer*



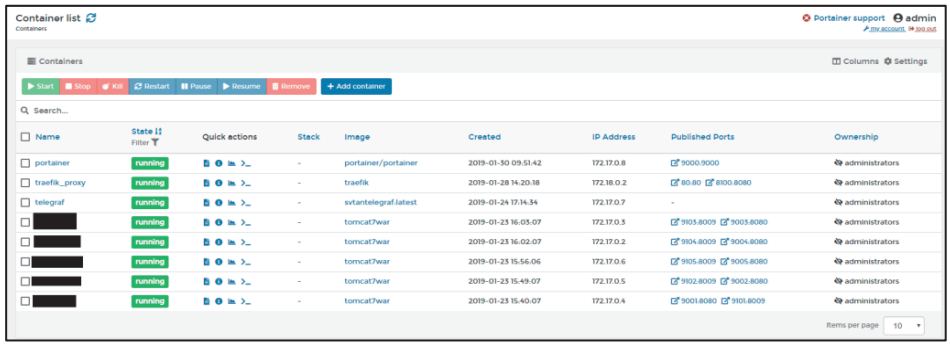
Rysunek 11.21. *Zarządzanie zewnętrznym środowiskiem za pomocą narzędzia Portainer*



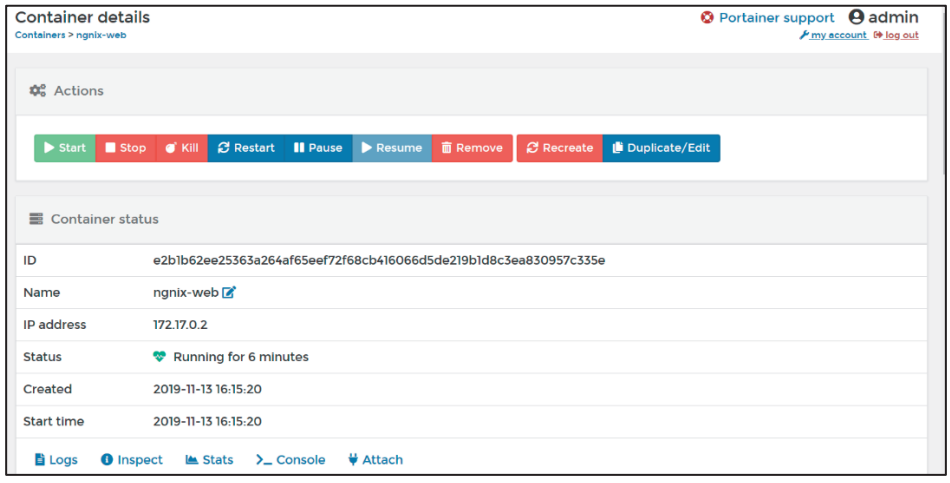
Rysunek 11.22. *Interfejs graficzny narzędzia Portainer*



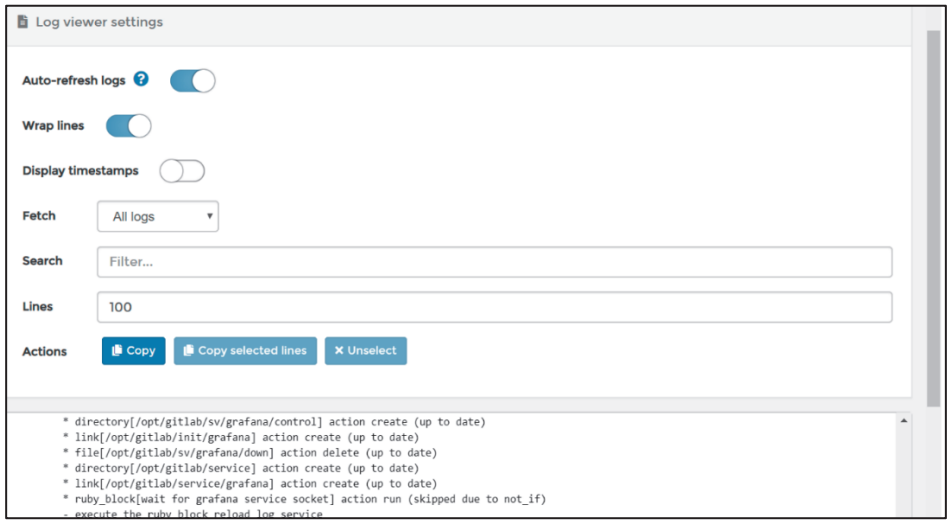
Rysunek 11.23. *Główna strona narzędzia Portainer*



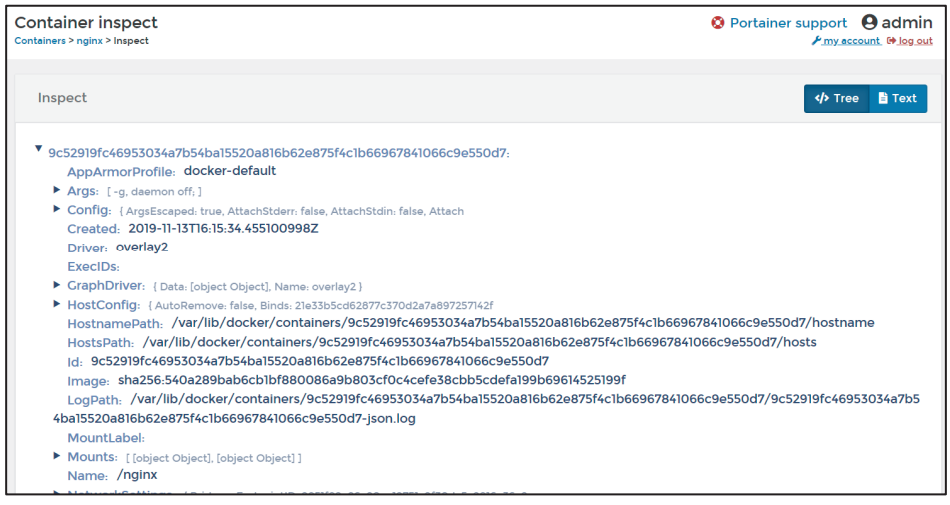
Rysunek 11.24. *Lista kontenerów w interfejsie narzędzia Portainer*



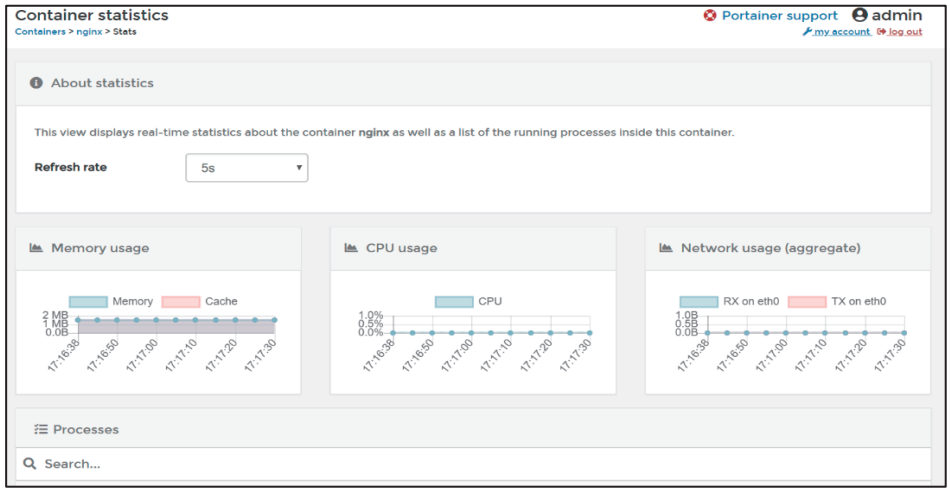
Rysunek 11.25. *Szczegóły kontenera w interfejsie narzędzia Portainer*



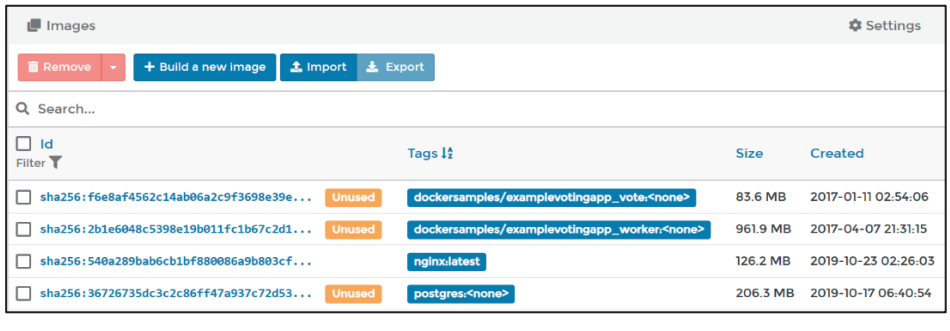
Rysunek 11.26. *Szczegółowe informacje o dziennikach kontenera*



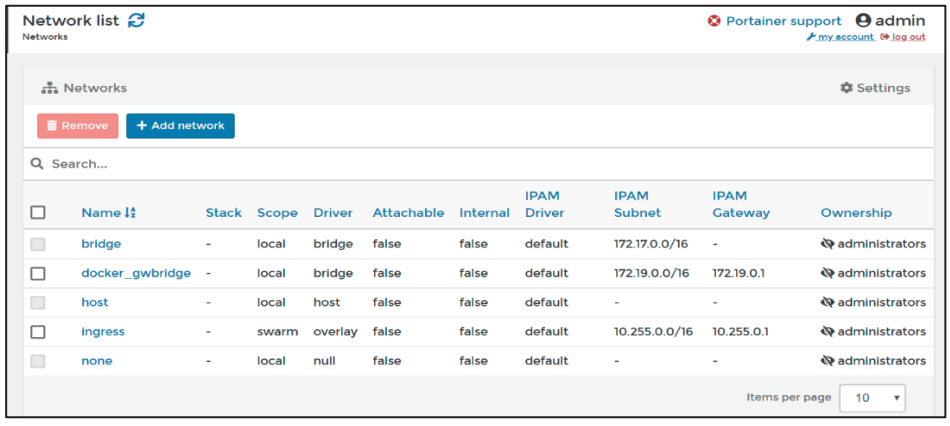
Rysunek 11.27. *Szczegółowe informacje o kontenerze*



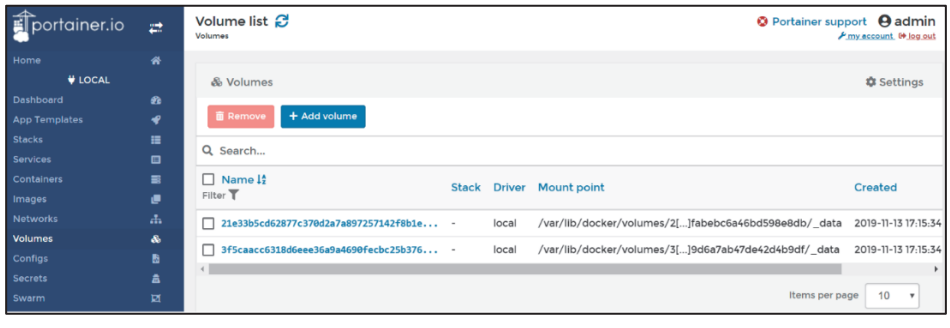
Rysunek 11.28. *Wskaźniki opisujące pracę kontenera*



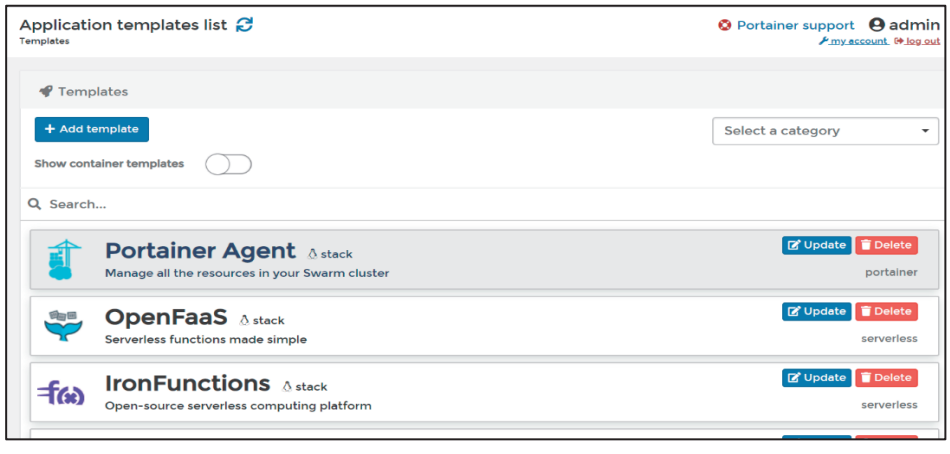
Rysunek 11.29. *Strona Images w narzędziu Portainer*



Rysunek 11.30. *Lista sieci w interfejsie narzędzia Portainer*

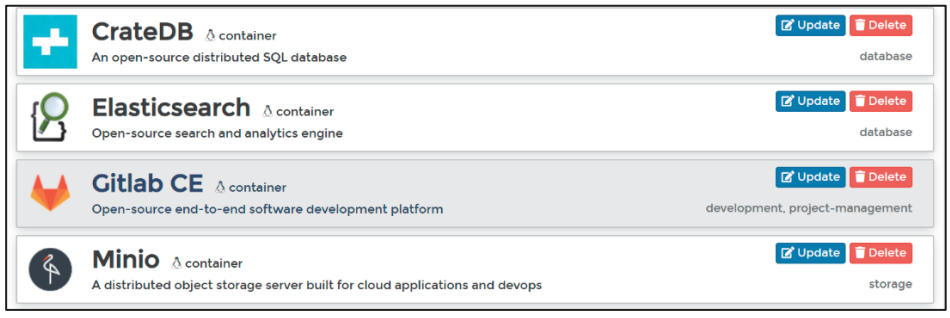


Rysunek 11.31. *Lista woluminów w interfejsie narzędzia Portainer*

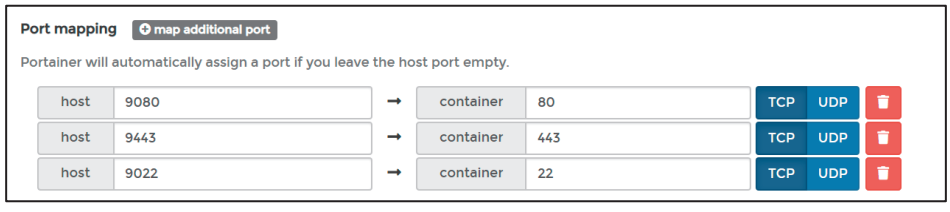


Rysunek 11.32. *Szablony aplikacji w interfejsie narzędzia Portainer*

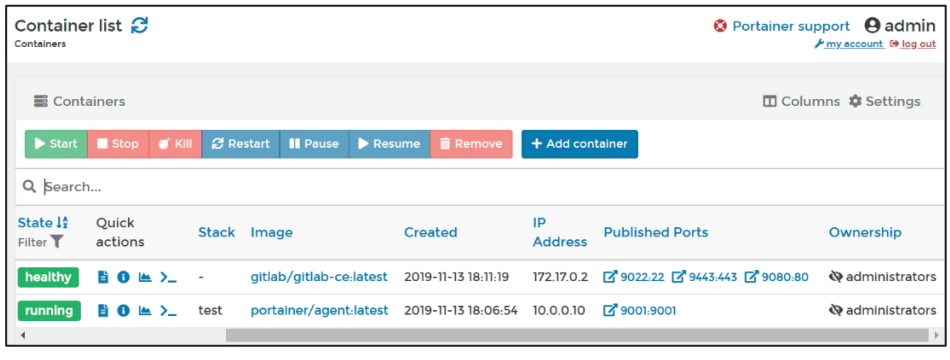
Rysunek 11.33. *Szablony kontenerów w interfejsie narzędzia Portainer*



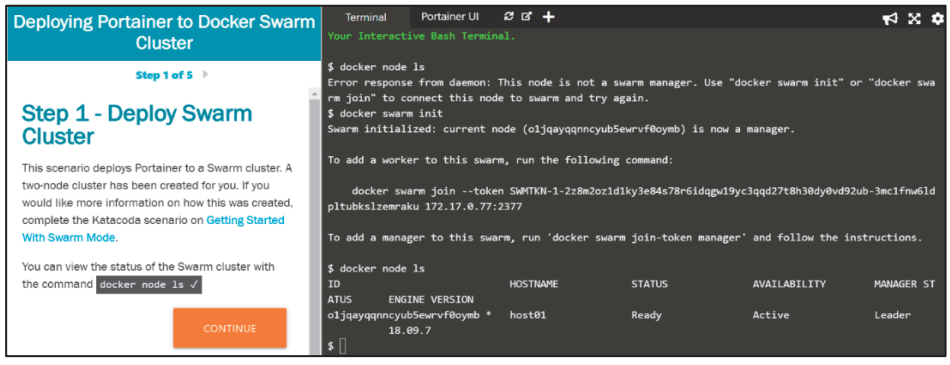
Rysunek 11.34. *Szablony narzędzia GitLab w interfejsie narzędzia Portainer*



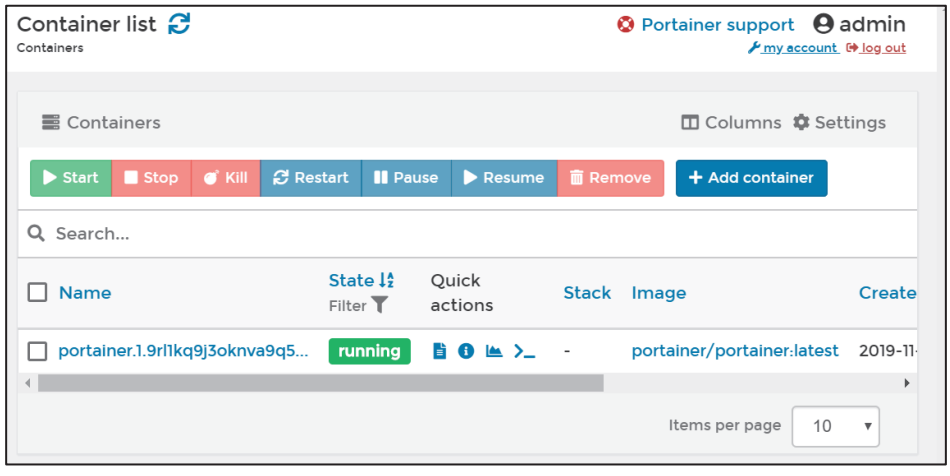
Rysunek 11.35. *Wiązanie portów w interfejsie narzędzia Portainer*



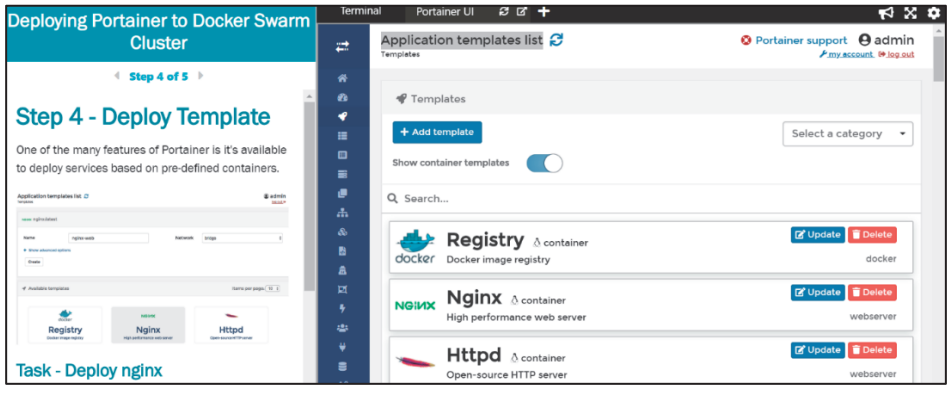
Rysunek 11.36. *Lista kontenerów w interfejsie narzędzia Portainer*



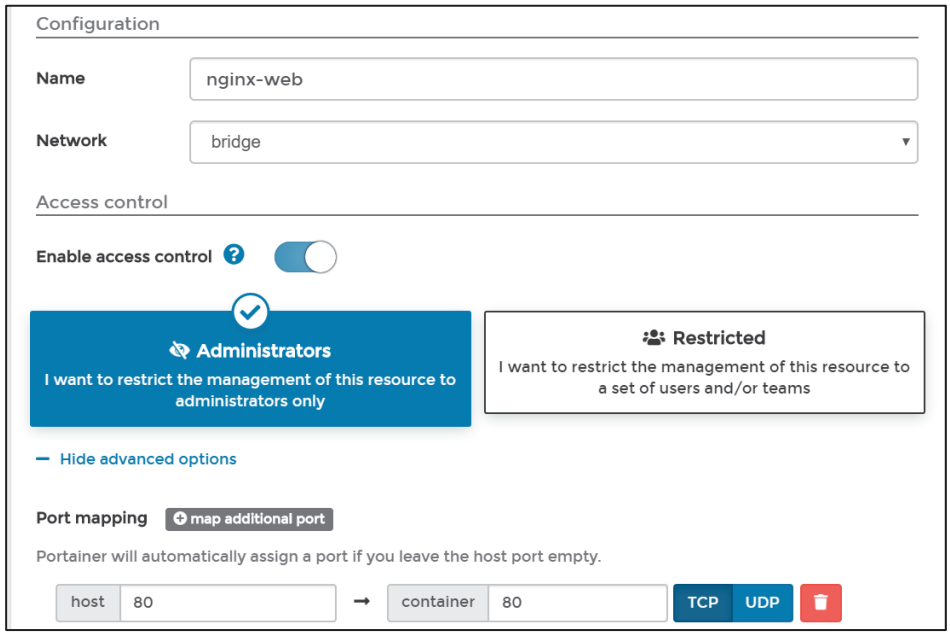
Rysunek 11.37. *Wdrożenie narzędzia Portainer w klastrze Docker Swarm*



Rysunek 11.38. *Uruchomiony kontener Portainer*



Rysunek 11.39. *Wdrożenie szablonu serwera Nginx*



Rysunek 11.40. *Konfiguracja kontenera z serwerem Nginx*

Container details

Image	nginx:latest@sha256:540a289bab6cb1bf880086a9b803cf0c4cefe38cbb5cdefa199b69614525199f
Port configuration	0.0.0.0:80 → 80/tcp
CMD	`nginx -g daemon off;`
ENTRYPOINT	`null`

ENV

PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NGINX_VERSION	1.17.5
NJS_VERSION	0.3.6
PKG_RELEASE	1~buster

Labels

maintainer	NGINX Docker Maintainers <docker-maint@nginx.com>

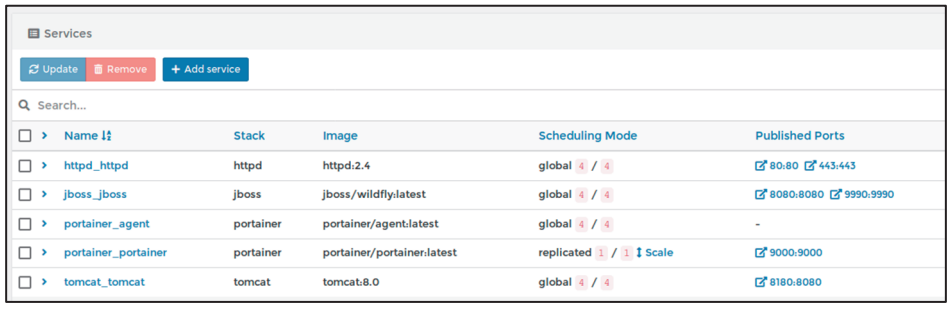
Rysunek 11.41. *Szczegóły kontenera z serwerem Nginx*

```
[root@docker-master1 ~]# docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS
olw40ii82gslxgfjtmy5h2ncr *   docker-master1      Ready               Active              Leader
lvvjz6c4bfe4ez16kzkyuc31o     docker-master2      Ready               Active              Reachable
q8em4kafthkm1phkuuab48eau     docker-worker1      Ready               Active
hb4ylcxtchc6kevimak7wxr9e     docker-worker2      Ready               Active
```

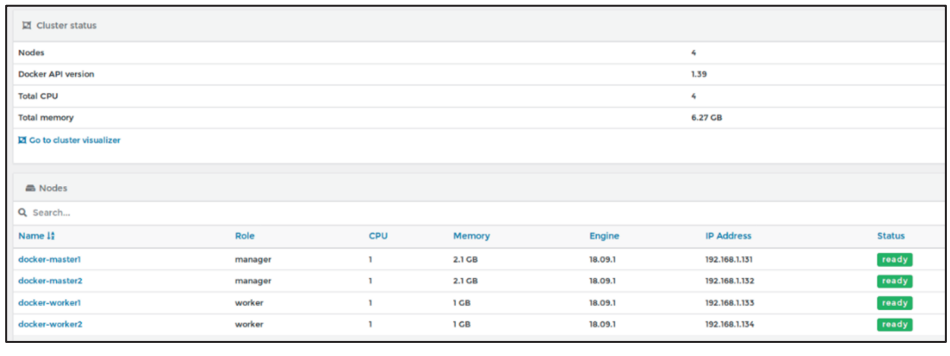
Rysunek 11.42. *Lista węzłów klastra Docker Swarm*

```
root@docker-master1 ~]# docker service ls
ID                  NAME                MODE                REPLICAS            IMAGE
tpmlzwsun4xt        httpd_httpd         global              4/4                 httpd:2.4
9040dr4bjqtf        jboss_jboss         global              4/4                 jboss/wildfly:latest
rhjjxlrqimj7        portainer_agent     global              4/4                 portainer/agent:latest
rs5wrgflhmax        portainer_portainer replicated          1/1                 portainer/portainer:latest
oxj7bf6e36kj        tomcat_tomcat       global              4/4                 tomcat:8.0
```

Rysunek 11.43. *Lista usług w klastrze Docker Swarm*



Rysunek 11.44. *Aktywne usługi w klastrze Docker Swarm*



Rysunek 11.45. *Węzły tworzące klaster Docker Swarm*