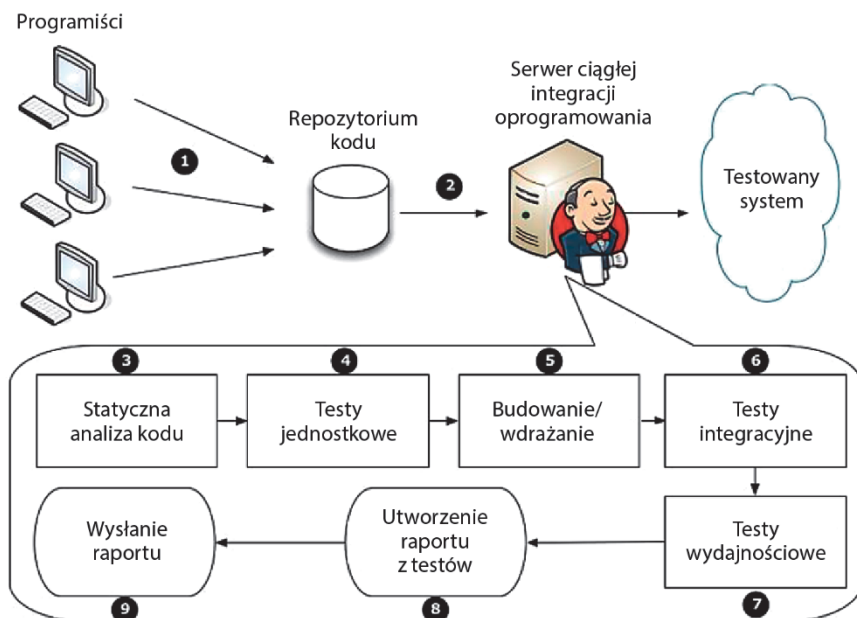
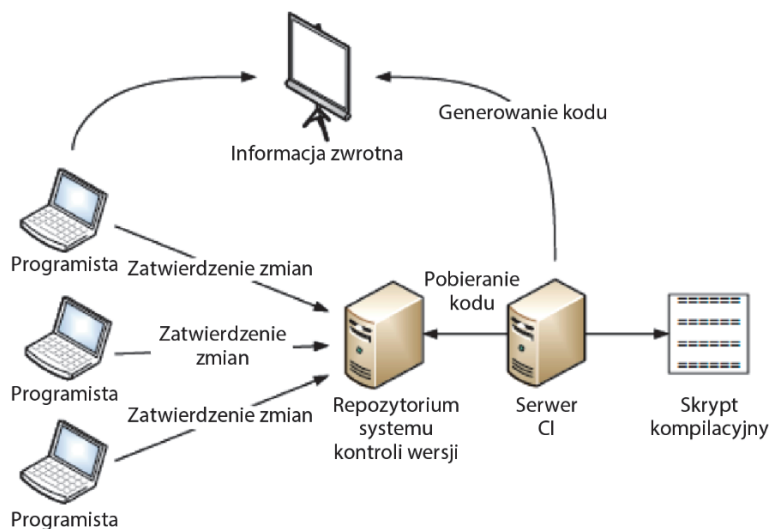


Bezpieczeństwo kontenerów w DevOps.

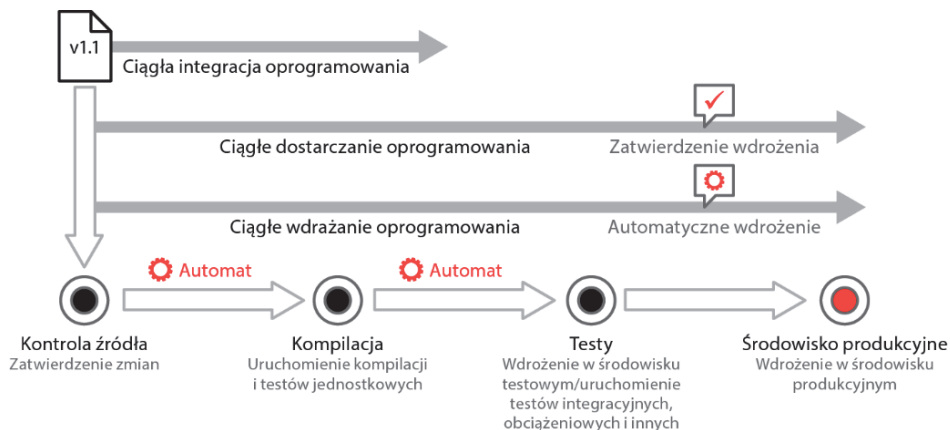
Zabezpieczanie i monitorowanie kontenerów Docker



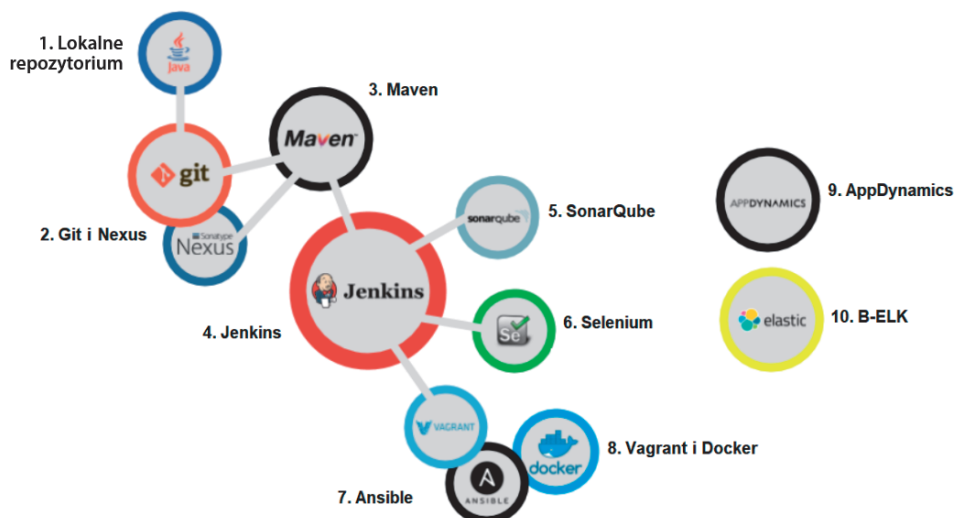
Rysunek 1.3. Proces ciągłego dostarczania oprogramowania





Rysunek 1.4. Proces ciągłej integracji oprogramowania




Rysunek 1.5. Kanał dostarczania oprogramowania



Rysunek 1.6. Narzędzia DevOps

PERIODIC TABLE OF DEVOPS TOOLS (v3)																EMBED		DOWNLOAD																													
1 Gl GitLab															2 Sp Splunk																																
3 Gh GitHub	4 Dt Docker	5 Xlr XebiaLabs 12. Release	6 Aws AWS	7 Az Azure	8 Gc Google Cloud	9 Op OpenShift	10 SI Sumo Logic									11 Sv Subversion																															
12 Dk Docker	13 Ur UrbanCode Release	14 Af Apache Fury	15 Ld Lambda	16 Ic IBM Cloud	17 Fd F5	18 Fw Fluentd									19 Cw CSPW																																
20 Dp Docker	21 Jn Jenkins	22 Cs CodeShip	23 Fn Fn	24 Ju Junit	25 Ka Karma	26 Su SauceLab	27 Ch Chef	28 Tf Terraform	29 Xld XebiaLabs 12. Deploy	30 Ud UrbanCode Deploy	31 Ku Kubernetes	32 Cc CA CD Director	33 Pr Pylons Release	34 Al Al	35 Os OpenStack	36 Ps Prometheus																															
37 At Artifactory	38 Rg Restic	39 Ba Bamboo	40 Vs VSTS	41 Se Selenium	42 Jm JMeter	43 Ja Jasmine	44 Sl Sauce Labs	45 An Ansible	46 Ru Rudder	47 Oc Octopus Deploy	48 Go GoCD	49 Ms Mesos	50 Gke GKE	51 Om OpenMata	52 Cp AWS CodePipeline	53 Cy Cloud Foundry	54 It ITRS																														
55 Nx Nexus	56 Fw Flyway	57 Tr Travis CI	58 Tc TeamCity	59 Ga Gatling	60 Tn TestNG	61 Tt Tentacle Toolbox	62 Pe Perforce	63 Pu Puppet	64 Pa Packard	65 Cd CD Deploy	66 Ec ElectricCloud	67 Ra Rancher	68 Aks AKS	69 Rk Rkt	70 Sp Spiralizer	71 Ir Iron.io	72 Mg MooGoo																														
73 Bb BitBucket	74 Pf Pulumi	75 Cr Circle CI	76 Cb AWS CodeBuild	77 Cu Cucumber	78 Mc Mocha	79 Lo Locust.io	80 Mf Maven JST	81 Sl Salt	82 Ce Ceph CFEngine	83 Eb ElasticSearch	84 Ca CA Automic	85 De Docker Enterprise	86 Ae AWS ECS	87 Cf Codefresh	88 Hm Helm	89 Aw Apache OpenStack	90 Ls Logstash																														
<div><div> XebiaLabs Enterprise DevOps</div><div> Follow @xebialabs</div><div>Publication Guidelines</div><div>Download</div></div>																		91 Xli XebiaLabs 12. Impact	92 Ki Kibana	93 Nr New Relic	94 Dt Datadog	95 Dd Dedicated	96 Ad AppDynamics	97 El ElasticSearch	98 Ni Nagios	99 Zb Zabbix	100 Zn Zenoss	101 Cx Chef Chefman's GAST	102 Sg Signal Sentry	103 Bd Black Duck	104 Sr SonarQube	105 Hv Hudson Hudson Corp Toolbox	106 Cn Cn CobaltNet VersionOne	107 Jr Jira	108 Ti Travis	109 Sl Slack	110 St Stride	111 Cn Cn CobaltNet VersionOne	112 Ry Remedy	113 Og Og Agile Central	114 Pd PagerDuty	115 Ss Ss PagerDuty	116 Tw Tw Twine	117 Ck Ck CyberArk CyberArk	118 Vc Veeva	119 Ff Fortify	120 Sd SDA Fortify SCA

Rysunek 1.7. Układ okresowy narzędzi DevOps




ANSIBLE





ANSIBLE

Website | Wikipedia

Ansible is an open-source software platform for configuring and managing computers. It combines multi-node software deployment, ad hoc task execution, and configuration management. It manages nodes over SSH or PowerShell and requires Python (2.4 or later) to

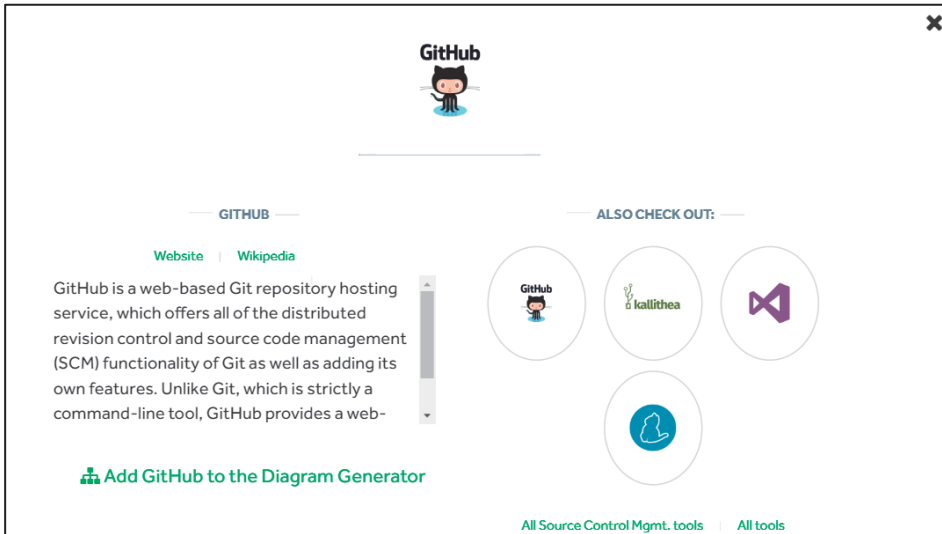
 Add Ansible to the Diagram Generator

ALSO CHECK OUT:

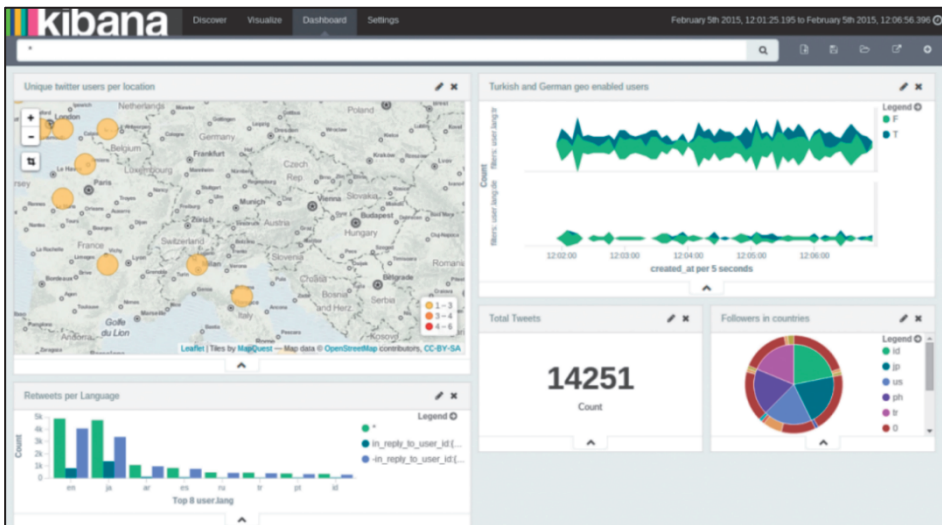





All Configuration tools | All tools

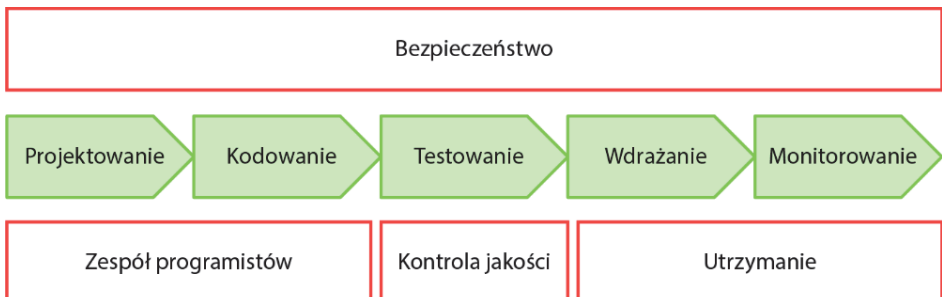
Rysunek 1.8. Opis narzędzia Ansible



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: sha256:9b1702dcfe32c873a770a32cfd306dd7fclc4fd134adfb783db68defc8894b3c
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

03:00:45

CLOSE SESSION

Instances  

+ ADD NEW INSTANCE


192.168.0.23

node1

bkmok9lt_bkmpftttcgkg009kvsmg

IP: 192.168.0.23

Memory: 0.77% (30.97MiB / 3.906GiB) CPU: 1.38%

SSH: ssh ip172-18-0-27-bkmok9ltcgkg009kvqdg@direct.labs.play 

DELETE

EDITOR

```
[node1] (local) root@192.168.0.23 ~
$ #####
# WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The PWD team.
#####
docker info
Client:
  Debug Mode: false
Server:
```

Rysunek 2.5. Uruchamianie kontenera Dockera w chmurze

Play with Docker classroom

About

First Alpine Linux Containers

Sep 19, 2017 • @jimcodified

In this lab you will run a popular, free, lightweight container and explore the basics of how containers work, how the Docker Engine executes and isolates containers from each other. If you already have experience running containers and basic Docker commands you can probably skip this intro exercise.

Concepts in this exercise:

- Docker engine
- Containers & Images
- Image registries and Docker Store (AKA Docker Hub)
- Container isolation

Tips:

Code snippets are shown in one of three ways throughout this environment:

- Code that looks like **this** is sample code snippets that is usually part of an explanation.
- Code that appears in box like the one below can be clicked on and it will automatically be

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing browser window.

```
#####
# WARNING!!!! #
# This is a sandbox environment. Using personal credentials #
# is HIGHLY discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
# #
# The FWD team. #
#####
[node1] (local) root@192.168.0.43 ~
$ docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
 Images: 0
 Server Version: 19.03.0-beta2
```

Your use of Play With Docker is subject to the Docker [automated](#)

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 responsibilities. #
# #
# The PWD team. #
#####
[node2] (local) root@192.168.0.38 ~
$ docker swarm join --token SWMTKN-1-lu4oi5o8db4831lpn37cyof8attas3h0xt2bf2y0fdf2my0id5-6xje7khf
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
4b5umtqguhjxj88xv3yk91520
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
4b5umtqguhjx    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				
tidzeczyt0z8	helloworld.2	alpine:latest	node2	Running
Running 7 seconds ago				
fqmov6k5j2wa	helloworld.3	alpine:latest	node1	Running
Running 7 seconds ago				
3qx814c75jlv	helloworld.4	alpine:latest	node1	Running
Running 7 seconds ago				

Rysunek 2.10. Zwiększenie liczby replik usługi w klastrze

Section 2: Configure Swarm Mode

Real-world applications are typically deployed across multiple hosts as discussed earlier. This improves application performance and availability, as well as allowing individual application components to scale independently. Docker has powerful native tools to help you do this.

An example of running things manually and on a single host would be to create a new container on `node1` by running `docker run -dt ubuntu sleep infinity`.

```
docker run -dt ubuntu sleep infinity
```

Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
d54ef98db41d: Pull complete
f8b845f45a87: Pull complete
e8db7b7c39f: Pull complete
965c48e9879: Pull complete
6d9ef339eaaa: Pull complete
Digest: sha256:df7888d9792c5941d90c466122f1acfb9e2dd1f56404f8d1e56298048885e45535
Status: Downloaded newer image for ubuntu:latest
846af84799444486943c08a39c8a68373cd19d1feaa932719268a5f5afddb7f1

This command will create a new container based on the `ubuntu:latest` image and will run the `sleep`

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

```
[node1] (local) root@192.168.0.21 ~
$ docker run -dt ubuntu sleep infinity
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
5b7339215d1d: Pull complete

# completely the user's responsibilities.
#
# The PWD team.
#####
[node2] (local) root@192.168.0.22 ~
$

# completely the user's responsibilities.
#
# The PWD team.
#####
[node3] (local) root@192.168.0.23 ~
```

Rysunek 2.11. Konfiguracja trybu Swarm

Swarm Mode Introduction for IT Pros

Sep 12, 2017 • @jimcodified

So far we have explored using single instances of containers running on a single host, much like a developer might do when working on a single service application or like an IT administrator might do on a test rig. Production applications are usually much more complex and this single server model will not work to coordinate 10s or 100s of containers and the network connections amongst them, not to mention the need to ensure availability and the ability to scale.

For real applications IT users and app teams need more sophisticated tools. Docker supplies two such tools: **Docker Compose** and **Docker Swarm Mode**. The two tools have some similarities but some important differences:

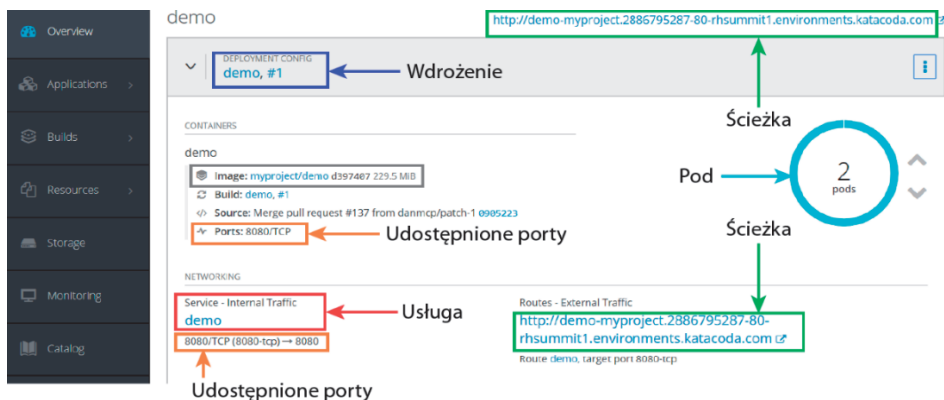
- Compose** is used to control multiple containers on a single system. Much like the `Dockerfile` we looked at to build an image, there is a text file that describes the application: which images to use, how many instances, the network connections, etc. But `Compose` only runs on a single system so while it is useful, we are going to skip `Compose` and go straight to `Docker Swarm Mode`.
- Swarm Mode** tells Docker that you will be running many Docker engines and you want to coordinate operations across all of them. `Swarm mode` combines the

If the commandline doesn't appear in the terminal, make sure popups are enabled or try resizing the browser window.

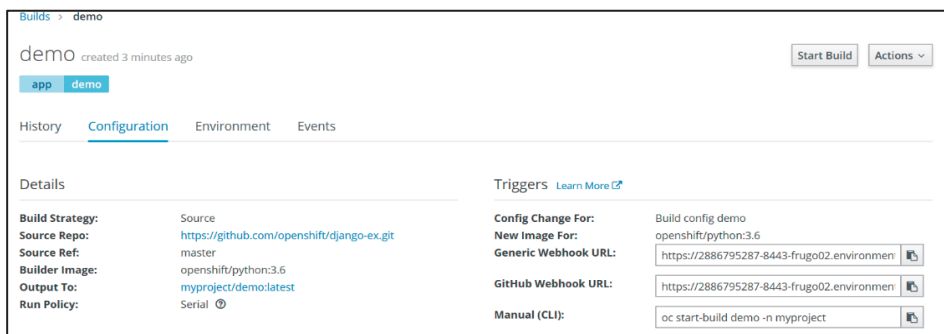
```
tive      Leader      19.03.0-beta2
[node1] (local) root@192.168.0.13 ~
$ docker node ls
ID                HOSTNAME        STATUS        AV
ce22be5272nk581bdattf103zz *  node1          Ready        Ac
tive      Leader      19.03.0-beta2
[node1] (local) root@192.168.0.13 ~
$

# WARNING!!!!
# This is a sandbox environment. Using personal credentials is
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The PWD team.
#####
[node2] (local) root@192.168.0.12 ~
$
```

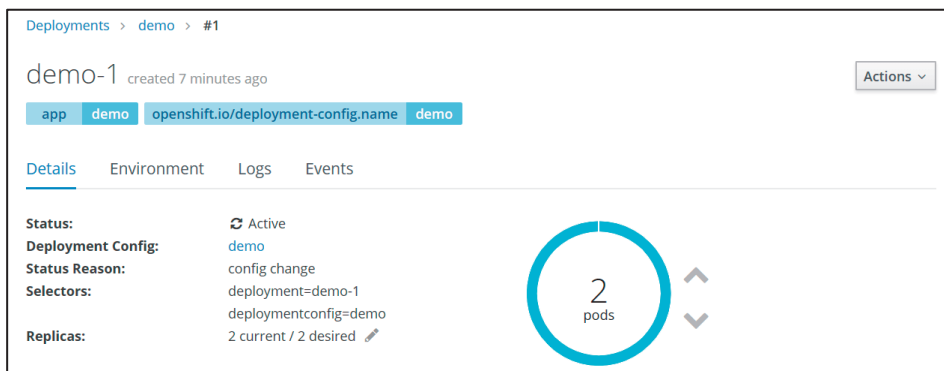
Rysunek 2.12. Wprowadzenie do trybu Swarm dla profesjonalistów IT



Rysunek 2.13. Projekt OpenShift



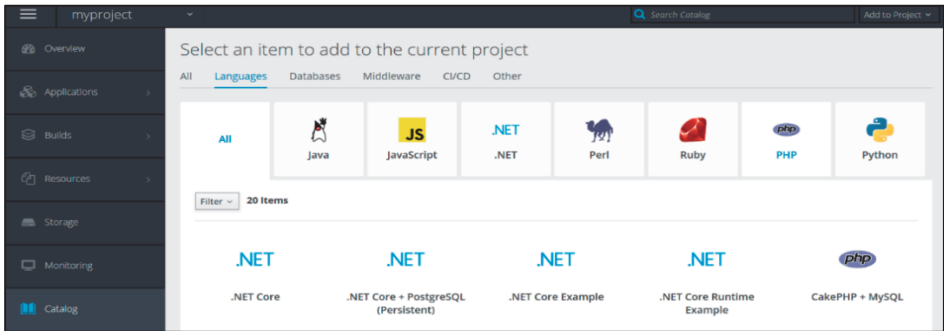
Rysunek 2.14. Konfiguracja wdrożenia



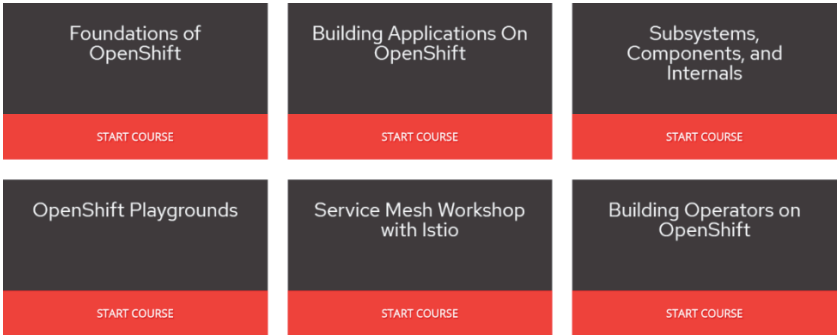
Rysunek 2.15. Szczegóły konfiguracji wdrożenia

Containers				
demo				
Image: myproject/demo e1d501d 229.5 MiB				
Build: demo, #1				
Source: Merge pull request #137 from danmcp/patch-1 0905223 authored by Ben Parees				
Ports: 8080/TCP				
Volumes				
Add Storage Add Config Files				
Pods				
Name	Status	Containers Ready	Container Restarts	Age
demo-1-qsqx6	Running	1/1	0	6 minutes
demo-1-7l45h	Running	1/1	0	8 minutes

Rysunek 2.16. Lista uruchomionych podów



Rysunek 2.17. Wbudowane szablony platformy OpenShift



Rysunek 2.18. Scenariusze szkoleniowe platformy OpenShift

Metadata from image python

Last inspected 28 minutes ago.

Versions

latest

buster

3

3.7

3.7.4

3.7.4-buster

3.7-buster

3-buster

Tags	
Created	July 30, 2019 at 02:33 AM
ID	008b021b6899
Download Size	330.4 MB
Labels	No labels
Layers	18

297.7 MB

buildpack-deps

testing

buster

What's this?

48.0 MB	ADD file:2cddee716e84c40540a69c48051bd2dcf6cd3bd02a3...	
	CMD ["bash"]	
7.4 MB	RUN apt-get update && apt-get install -y --no-instal...	
9.5 MB	RUN set -ex; if ! command -v gpg > /dev/null; then ...	
49.4 MB	RUN apt-get update && apt-get install -y --no-instal...	

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

Supported tags and respective Dockerfile links

Simple Tags

- 3.8.0b2-buster, 3.8-rc-buster, rc-buster
- 3.8.0b2-slim-buster, 3.8-rc-slim-buster, rc-slim-buster, 3.8.0b2-slim, 3.8-rc-slim, rc-slim
- 3.8.0b2-alpine3.10, 3.8-rc-alpine3.10, rc-alpine3.10, 3.8.0b2-alpine, 3.8-rc-alpine, rc-alpine
- 3.8.0b2-windowsservercore-ltsc2016, 3.8-rc-windowsservercore-ltsc2016, rc-windowsservercore-ltsc2016
- 3.8.0b2-windowsservercore-1803, 3.8-rc-windowsservercore-1803, rc-windowsservercore-1803
- 3.8.0b2-windowsservercore-1809, 3.8-rc-windowsservercore-1809, rc-windowsservercore-1809
- 3.7.4-buster, 3.7-buster, 3-buster, buster
- 3.7.4-slim-buster, 3.7-slim-buster, 3-slim-buster, slim-buster, 3.7.4-slim, 3.7-slim, 3-slim, slim

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

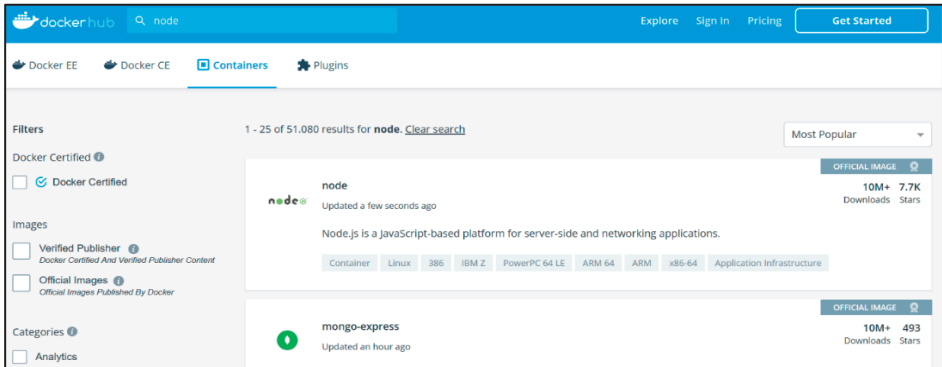
```
[node1] (local) root@192.168.0.33 ~/ubuntutest
$ docker run -dti --name mycontainer e22b016f25f7
aca0408912343d042b3352146dd87e891179941498237e220a5d254175288625
[node1] (local) root@192.168.0.33 ~/ubuntutest
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
aca040891234	e22b016f25f7	"bash"	12 seconds ago	Up 10 sec

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
79b33d74c408	mysql	"docker-entrypoint.s..."	20 seconds ago	Up 19 seconds

```

PORTS          NAMES
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@sha256:35ff9f44818f8850fld318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
sha256:35ff9f44818f8850fld318aa69c2e7ba61d85e3b93283078c10e56e7d864c183: Pulling from library/python
e5el55d5ald1: 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: sha256:35ff9f44818f8850fld318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
Status: Downloaded newer image for python@sha256:35ff9f44818f8850fld318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
docker.io/library/python@sha256:35ff9f44818f8850fld318aa69c2e7ba61d85e3b93283078c10e56e7d864c183
[node1] (local) root@192.168.0.28 ~
$ docker pull python@sha256:35ff9f44818f8850fld318aa69c2e7ba61d85e3b93283078c10e56e7d864c1831
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:db9f2e75b91780c804c283081123fbelb2b4080fe124eeb040f8ad1bfd70fe38
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 E3FF2839C048825C084DEBE9B26995E310250568
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
d47e0b90b502870403e3f634632ffd4012c792b02aballf4264cc5eed56c7089
[nodet] (local) root@192.168.0.28 ~

$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
d47e0b90b502	registry:2	"/entrypoint.sh /etc..."	2 minutes ago	Up 2 minutes	0.0.0.0.5000->5000/tcp

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

» Ubuntu » Packages » bionic (18.04LTS) » admin » apparmor-profiles

[Source: apparmor] [xenial][xenial-updates][bionic][bionic-updates][cosmic][disco][disco-updates][eoan]

Package: apparmor-profiles (2.12-4ubuntu5.1) [security]

experimental profiles for AppArmor security policies

Other Packages Related to apparmor-profiles

depends

recommends

suggests

enhances

apparmor

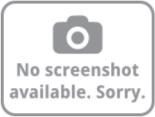
(>= 2.8.96~2535-0ubuntu1~)

user-space parser utility for AppArmor

Download apparmor-profiles

Architecture	Package Size	Installed Size	Files
all	31.1 kB	360.0 kB	[list of files]

Links for apparmor-profiles



Ubuntu Resources:

[Bug Reports](#)

Rysunek 5.6. Pakiet modułu AppArmor dla systemu Ubuntu

Docker Bench for Security

```
# -----
# Docker Bench for Security v1.3.3
# Docker, Inc. (c) 2015-
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker Community Edition Benchmark v1.1.0.
# -----


Initializing Fri Jul 14 09:18:42 UTC 2017

[INFO] 1 - Host Configuration
[WARN] 1.1 - Ensure a separate partition for containers has been created
[NOTE] 1.2 - Ensure the container host has been Hardened
[PASS] 1.3 - Ensure Docker is up to date
[INFO] * Using 17.06.0 which is current
[INFO] * Check with your operating system vendor for support and security maintenance for Docker
[INFO] 1.4 - Ensure only trusted users are allowed to control Docker daemon
[INFO] * docker:x992:vagrant
[WARN] 1.5 - Ensure auditing is configured for the Docker daemon
[WARN] 1.6 - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.7 - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.8 - Ensure auditing is configured for Docker files and directories - docker.service
[INFO] 1.9 - Ensure auditing is configured for Docker files and directories - docker.socket
[INFO] * File not found
[INFO] 1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO] * File not found
[INFO] 1.11 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
```


Docker Pull Command

```
docker pull docker/docker-bench-security
```

Owner

 docker

Source Repository


 [github/docker/docker-bench-security](#)

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

Branch: master

docker-bench-security / tests /

Create new file Upload files Find file History

 konstruktoid

use only year and month for version check #309

Latest commit 326e31f on 13 Apr

1_host_configuration.sh	use mountpoint and DockerRootDir #332	10 months ago
2_docker_daemon_configuration.sh	use only year and month for version check #309	4 months ago
3_docker_daemon_configuration_files.sh	linting	10 months ago
4_container_images.sh	revert grep thought fail	5 months ago
5_container_runtime.sh	linting	10 months ago
6_docker_security_operations.sh	Improve docker-bench-security json output	10 months ago
7_docker_swarm_configuration.sh	fix test 7.4 using 5.25 as a model	9 months ago
99_community_checks.sh	Improve docker-bench-security json output	10 months ago

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}}' "$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
done

```

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 '{{.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
done

```

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

```

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

    if [ "$user" = "User=0" ] || [ "$user" = "User=root" ] || [ "$user" = "User=" ] || [ "$user" = "User=[]" ] || [ "$user" = "User=<no value>" ]; then
        # 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))
    fi

```

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
done

```

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
DockerScan 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

There are 41 vulnerable components (New scan in progress, showing results from 14 days ago) [Provide Feedback](#)

Component	Vulnerability	Severity
libseccomp 2.6.7-2	CVE-2019-5893	Critical
LGPL: Lgpl License		
PATHS:		
libx86_64-linux-gnu/libaudit.so.1.0.0		
glibc 2.24-11+deb9u4	CVE-2018-1000001	Critical
LGPL: Lgpl License	CVE-2018-20796	Major
PATHS:	CVE-2019-9192	Major
libx86_64-linux-gnu/libc-2.24.so	CVE-2019-6488	Major
usr/share/doc/multiarch-support/changelog.Debian.gz	CVE-2019-7309	Minor
sbin/ldconfig		
berkeleydb 5.3.28-12+deb9u1	CVE-2016-0689	Major
sleepycat: Copyleft License	CVE-2016-0682	Major
PATHS:	CVE-2016-0694	Major
usr/libx86_64-linux-gnu/libdb-5.3.so	CVE-2016-3418	Major
	CVE-2016-0692	Major

Rysunek 6.1. Analiza warstw obrazu Dockera

7 ENV REDIS_DOWNLOAD_S...df2a0352ab575c159df2d
Compressed size: 0.0

No components in this layer

8 /bin/sh -c set -e
COMPONENT

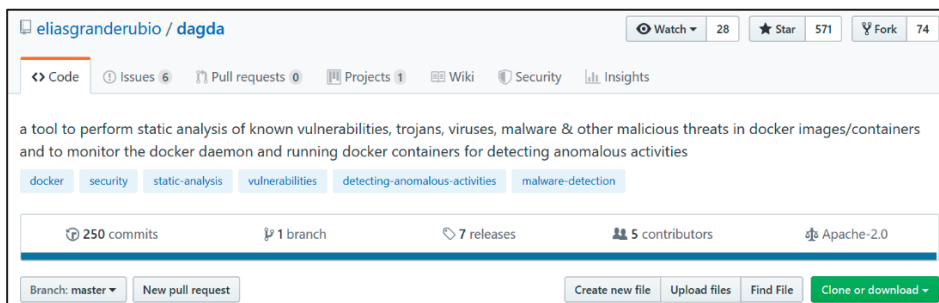
lua 5.1.5
MIT:Permissive License

[CVE-2014-5461](#) Major

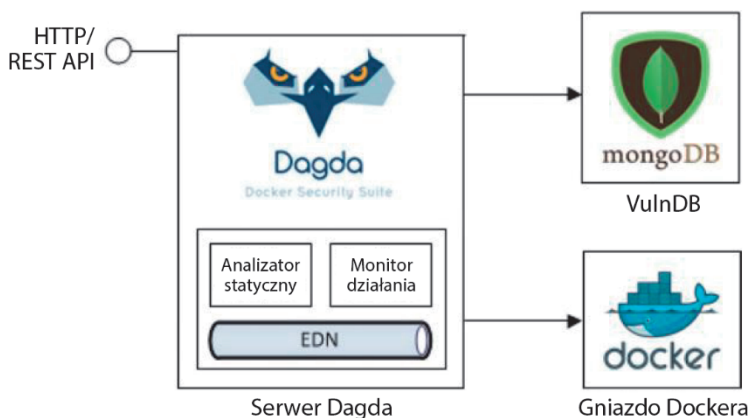
Buffer overflow in the vararg functions in ldo.c in Lua 5.1 through 5.2.x before 5.2.3 allows context-dependent attackers to cause a denial of service (crash) via a small number of arguments to a function with a large number of fixed arguments.

VIEW ALL


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



aqua MicroScanner

Aqua Security's MicroScanner lets you check your container images for vulnerabilities. If your image has any known high-severity issue, MicroScanner can fail the image build, making it easy to include as a step in your CI/CD pipeline.

Registering for a token

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

clair_config

Dockerfile

README.MD

docker-compose.yml

README.MD

clair

clair

clair

clair


2 years ago

2 years ago

2 years ago

2 years ago

Clair CoreOS with local-image-analyzer ready

clair

A Dockerfile and docker-compose for running Vulnerability Static Analysis for Containers using Clair.

The Dockerfile also contains the local-image-analyzer for analyzing local docker images.

Rysunek 6.18. Repozytorium GitHub zawierające skaner Clair

```
1 version: '2'
2 services:
3   postgres:
4     container_name: clair_postgres
5     image: postgres:latest
6     environment:
7       POSTGRES_PASSWORD: password
8
9   clair:
10    container_name: clair_clair
11    image: hxquangnhat/clair:latest
12    depends on:
13      - postgres
14    ports:
15      - "6060-6061:6060-6061"
16    links:
17      - postgres
18    volumes:
19      - /tmp:/tmp
20      - ./clair_config:/config
21      - /var/run/docker.sock:/var/run/docker.sock
22    command: [-config, /config/config.yaml]
```

Rysunek 6.19. Plik docker-compose.yml skanera Clair

```

dockerserverclair@DockerServerClair:~$ sudo mkdir $HOME/clair_config
dockerserverclair@DockerServerClair:~$ sudo curl -L https://raw.githubusercontent.com/coreos/clair/master/config.yaml
sample -o $PWD/clair_config/config.yaml
% Total    % Received % Xferd Average Speed      Time     Time     Time   Current
           Dload  Upload    Total   Spent    Left   Speed
100 2941  100 2941    0    0 17300    0  --:--:-- --:--:-- --:--:-- 17402
dockerserverclair@DockerServerClair:~$ ls
clair_config
dockerserverclair@DockerServerClair:~$ █

```

Rysunek 6.23. Drugi krok instalacji

```

dockerserverclair@DockerServerClair:~/clairscanners$ sudo mv clair-scanner_linux_amd64 clair-scanner
dockerserverclair@DockerServerClair:~/clairscanners$ ls
clair-scanner
dockerserverclair@DockerServerClair:~/clairscanners$ sudo chmod +x clair-scanner
dockerserverclair@DockerServerClair:~/clairscanners$ ls
clair-scanner
dockerserverclair@DockerServerClair:~/clairscanners$ █

```

Rysunek 6.26. Nadanie skanerowi Clair uprawnień do działania

```

2018/11/25 16:28:21 [INFO] □ Analyzing dde0e59bf1b1dfd7fc5f6d4a60dcaa2a7d936eab1fe455a9f0d396b64b706b37
2018/11/25 16:28:24 [INFO] □ Analyzing ffc629ef32caddb37507405b90fe425e1bdaf77d087832ea68e3fee5d079516e
2018/11/25 16:28:24 [INFO] □ Analyzing e90cc14beb2b09c943ddGb4a55fb3c2c14fb06aac38c5308f24f07f2faf45a4b
2018/11/25 16:28:24 [INFO] □ Analyzing f7ee4dae102e919cd5e29ee3402d36c8736888ec16a16399957df2674019c3ec
2018/11/25 16:28:24 [INFO] □ Analyzing elc8276f5ff4ff62f553edf8d749a594c8fb92f13485f45ddc37d7c5348e144
2018/11/25 16:28:24 [INFO] □ Analyzing 512148149fabf6ca5calbcfc6ef3c3c8d2b3076746d5b74719011a8208eb796b
2018/11/25 16:28:24 [INFO] □ Analyzing 70d3771c19ba4c1b4ee59092a35eeccc79187793b9680e4b3e4c6e7579e5cfd6
2018/11/25 16:28:24 [WARN] □ Image [vulnerables/cve-2016-10033] contains 233 total vulnerabilities
2018/11/25 16:28:24 [ERROR] □ Image [vulnerables/cve-2016-10033] contains 233 unapproved vulnerabilities

```

STATUS	CVE SEVERITY	PACKAGE NAME	PACKAGE VERSION	CVE DESCRIPTION
Unapproved	High	util-linux	2.25.2-6	runuser in util-linux al
				lows local users to escape to
				crafted TIOCSTI ioctl call,
				the parent session via a
				which pushes characters
				to the terminal's input buffer.

Rysunek 6.28. Szósty krok instalacji

```

software vulnerable to timing attacks | | | | local users to exploit s
| | | | | | | | via a side-channel timin
g attack on 'port contention'. | | | | | | | |
| | | | | | | | https://security-tracker
.debian.org/tracker/CVE-2018-5407 | | | | | | | |

```

STATUS	CVE SEVERITY	PACKAGE NAME	PACKAGE VERSION	CVE DESCRIPTION
Unapproved	Unknown	python2.7	2.7.9-24deb8u1	Python's elementtree C a
				ccelerator failed to initialise
				initialization. This could make
				l of service attacks against
				XML document that would cause
				ions in Expat's internal data
				rge amounts CPU and RAM. Python
				2.7 are believed to be vulnerable.
				https://security-tracker
				debian.org/tracker/CVE-2018-14647

```


```

STATUS	CVE SEVERITY	PACKAGE NAME	PACKAGE VERSION	CVE DESCRIPTION
Unapproved	Unknown	php5	5.6.30+dfsg-0+deb8u1	https://security-track
				er.debian.org/tracker/CVE-2018-19518

Rysunek 6.29. Znalezienie zagrożenia obrazu vulnerables/cve-2016-10033

Create a new container

First we'll create a container with a single new file based off of the `ubuntu` base image:

```
$ docker run ubuntu echo "fun" > newfile
```

The container will immediately terminate (because its one command is `echo`), so we'll use `docker ps -l` to list it:

\$ docker ps -l	IMAGE	COMMAND	CREATED
CONTAINER ID	ubuntu:12.04	echo fun	31 seconds ago
07f2065197ef			

Make note of the *container id*; we'll need it for the commit command.

Rysunek 6.32. Identyfikator zwrócony przez nowo utworzony kontener

Tag the container to an image

We next need to tag the container to a known image name

Note that the *username* must be your Quay.io username and *reponame* is the new name of your repository.

```
$ docker commit 07f2065197ef quay.io/username/reponame
e7050e05a288f9f3498ccd2847fee966d701867bc671b02abf03a6629dc921bb
```

Rysunek 6.33. Oznaczenie kontenera nazwą obrazu

Repository Tags

CompactExpanded

1 - 1 of 1

Filter Tags...

TAG	LAST MODIFIED	SECURITY SCAN	SIZE	EXPIRES	MANIFEST
<input type="checkbox"/> latest	5 minutes ago	Unsupported	20.4 MB	Never	SHA256 0fe333c46ff4

+ Add New Tag

Edit Labels

Delete Tag

Change Expiration

Rysunek 6.36. Operacje na oznaczeniach obrazu

Manifest Layers

>_ /bin/bash

CMD

["python3"]

RUN

set -ex; wget -O get-pip.py 'https://bootstrap.pypa.io/get-pip.py'; python get-pip.py --disable-pip-version-check --no-cache-dir "pip==\$PYTHON_PIP_VERSION"; pip --version; find /usr/local -depth \(\(-type d -a \(-name test -o -name tests \) \) -o \(-type f -a \(-name '*.pyc' -o -name '*.pyo' \) \) \) -exec rm -rf '{}' +; rm -f get-pip.py

ENV

PYTHON_PIP_VERSION=9.0.1

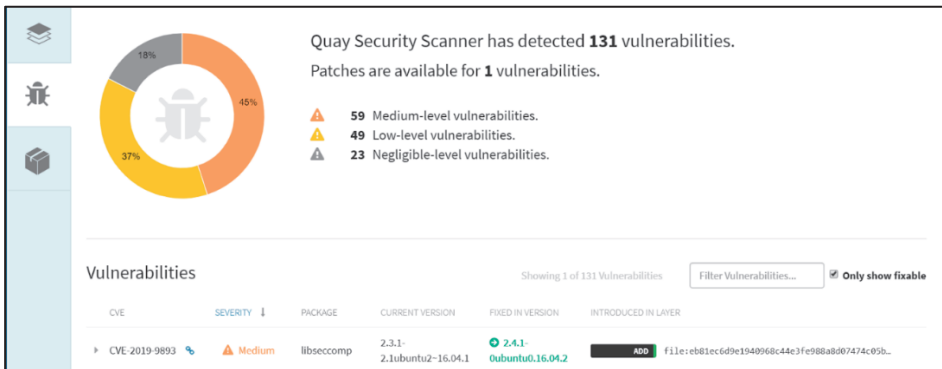
RUN

cd /usr/local/bin && ln -s idle3 idle && ln -s pydoc3 pydoc && ln -s python3 python && ln -s python3-config python-config

RUN

set -ex && buildDeps=' dpkg-dev tcl-dev tk-dev ' && apt-get update && apt-get install -y \$buildDeps --no-install-recommends && rm -rf /var/lib/apt/lists/* && 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 --keyserver ha.pool.sks-keyservers.net --recv-keys "\$GPG_KEY" && gpg --batch --verify python.tar.xz.asc python.tar.xz && 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 && cd /usr/src/python && gnuArch="\$(dpkg-architecture --query DEB_BUILD_GNU_TYPE)" && ./configure --build="\$gnuArch" --enable-loadable-sqlite-extensions --enable-shared --with-system-expat --with-system-ffi --without-ensurepip && make -j "\$(nproc)" && make install && ldconfig && apt-get purge -y --auto-remove \$buildDeps && find /usr/local -depth \(\(-type d -a \(-name test -o -name tests \) \) -o \(-type f -a \(-name '*.pyc' -o -name '*.pyo' \) \) \) -exec rm -rf '{}' + && rm -rf /usr/src/python

Rysunek 6.37. Warstwy manifestu



Rysunek 6.38. Zagrożenia znalezione przez skaner Quay.io

Image Vulnerabilities					
Showing 31 of 569 Vulnerabilities					
Filter Vulnerabilities... <input checked="" type="checkbox"/> Only show fixable					
CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN IMAGE
CVE-2017-17805	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2017-17558	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2018-2562	High	mysql-5.5	5.5.58-0+deb8u1	5.5.59-0+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2017-16538	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2017-8824	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2017-17806	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2017-16939	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>
CVE-2017-15968	High	linux	3.16.51-2	3.16.51-3+deb8u1	<div>set -ex; apt-get update; apt...</div>

Rysunek 6.39. Zagrożenia wraz z odpowiadającymi im numerami CVE

Vulnerabilities

Filter Vulnerabilities..

CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN LAYER
▶ CVE-2016-4448	10 / 10	libxml2	2.9.1+dfsg1-5+deb8u5	(None)	<div>set -ex; apt-get upda</div>
▶ CVE-2017-17458	10 / 10	mercurial	3.1.2-2+deb8u4	3.1.2-2+deb8u6	<div>apt-get update && apt</div>
▼ CVE-2017-18017	10 / 10	linux	3.16.51-2	3.16.56-1	<div>set -ex; apt-get upda</div>

VECTORS

Access Vector	Access Complexity	Authentication	Confidentiality Impact	Integrity Impact
Network	Low	None	Complete	Complete
Adjacent Network	Medium	Single	Partial	Partial
Local	High	Multiple	None	None

DESCRIPTION

The tcpmss_mangle_packet function in net/netfilter/xt_TCPMSS.c in the Linux kernel before 4.11, and 4.9.x before 4.9.36, allows remote attackers to cause a denial of service (use-after-free and memory corruption) or possibly have unspecified other impact by leveraging the presence of xt_TCPMSS in an iptables action.

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

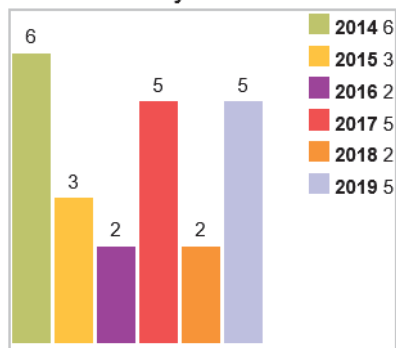
```

Rysunek 6.42. Pobranie kodu źródłowego silnika Anchore

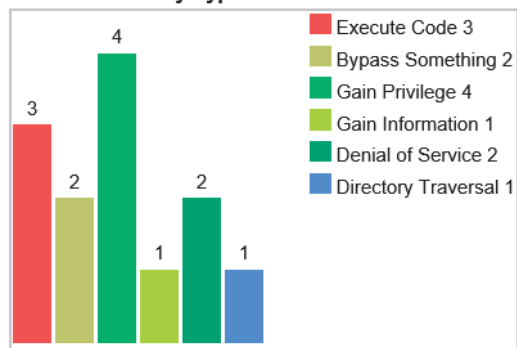
Docker » Docker : Vulnerability Statistics																
Vulnerabilities (23) CVSS Scores Report Browse all versions Possible matches for this product Related Metasploit Modules																
Related OVAL Definitions : Vulnerabilities (0) Patches (2) Inventory Definitions (0) Compliance Definitions (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

Vulnerabilities By Year



Vulnerabilities By Type



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	<code>./dirtyc0w file content</code>	Read-only write	/proc/self/mem
cowroot.c	<code>./cowroot</code>	SUID-based root	/proc/self/mem
dirtycow-mem.c	<code>./dirtycow-mem</code>	libc-based root	/proc/self/mem
pokemon.c	<code>./d file content</code>	Read-only write	PTRACE_POKEDATA
dirtycow.cr	<code>dirtycow --target --string --offset</code>	Read-only write	/proc/self/mem
dirtyc0w.c	<code>./dirtycow file content</code>	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

```

165 def is_layer_safe(layer_file):
166     results = []
167     try:
168         tar = tarfile.open(layer_file, mode='r:gz')
169     except tarfile.ReadError:
170         tar = tarfile.open(layer_file, mode='r')
171
172     while True:
173         next_block = tar.next()
174         if not next_block:
175             break
176
177         filename = next_block.name
178         link_destination = next_block.linkname
179
180         if not os.path.relpath(filename).find('..\') or not os.path.relpath(filename).find('../'):
181             results.append((filename, 0, layer_file))
182
183         if link_destination:
184             if link_destination[0] not in ['/', '\\']:
185                 full_path = os.path.dirname(filename) + "/" + link_destination
186                 if not os.path.relpath(full_path).find('..\') or not os.path.relpath(full_path).find('../'):
187                     results.append((full_path, 1, layer_file))

```

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

Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NVD

VULNERABILITIES
SEARCH AND STATISTICS

Search Results (Refine Search)

Sort results by: Publish Date Descending Sort

Search Parameters: There are 57 matching records.
Displaying matches 1 through 20.

- Results Type: Overview
- Keyword (text search): docker
- Search Type: Search All

Vuln ID Summary CVSS Severity

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

CUSTOMER PORTAL

[Products & Services](#)
[Tools](#)
[Security](#)
[Community](#)

Security Updates > Red Hat CVE Database

Security Advisories

Red Hat CVE Database

Security Labs

Filter By Year

	CVE	Synopsis	Impact	Publish Date
<input checked="" type="button" value="Low"/>	CVE-2019-10156	** RESERVED ** This candidate has been reserved by an organization or individual that will use it when announcing a new security problem. When the candidate has been publicized, the details for this candidate will be provided.	Moderate	04 Jun 2019
<input checked="" type="button" value="Critical"/>	CVE-2019-10149	A flaw was found in Exim versions 4.87 to 4.91 (inclusive). Improper validation of recipient address in deliver_message() function in /src/deliver.c may lead to remote	Critical	04 Jun 2019

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:106288
- URL:<http://www.securityfocus.com/bid/106288>
- 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/2019-05/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-2019-16984	863		Bypass	2019-09-25	2019-10-07	5.0	None	Remote	Low	Not required	None	Partial	None
run 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
run 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
HandlerRequestAsync 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

```
~$ kubectll 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-aps 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)
[FAIL] 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                                I 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

```
[15] 1:kubectldig "l13o1" 12:21 20-May-19
Viewing: Processes For: whole machine
Source: Live System Filter: evt.type!=switch
Select View Containers
Connections List all the containers running on this machine, and the resources that each of them uses.
Containers
Containers Errors Tips
Directories Select a container and click enter to drill down into it. At that point, you will be able to
Errors access several views that will show you the details of the selected container.
File Opens List
Files Columns
I/O by Type CPU: Amount of CPU used by the container.
K8s Controllers PROCs: Number of processes currently running inside the container.
K8s Deployments THREADS: Number of threads currently running inside the container.
K8s Namespaces VIRT: Total virtual memory for the process.
K8s Pods RES: Resident non-swapped memory for the process.
K8s ReplicaSets FILE: Total (input+output) file I/O bandwidth generated by the container, in bytes per second
K8s Services .
Marathon Apps NET: Total (input+output) network bandwidth generated by the container, in bytes per second.
Marathon Groups ENGINE: Container type.
Mesos Frameworks IMAGE: Container image name.
Mesos Tasks ID: Container ID. The format of this column depends on the containerization technology. For e
New Connections xample, Docker ID are 12 characters hexadecimal digit strings.
Page Faults NAME: Name of the container.
Processes
Processes CPU ID
Processes Errors containers
Processes FD Usage
Server Ports Filter
Slow File I/O container.name != host
Socket Queues
Spectrogram-File
Spy Syslog Action Hotkeys
Spy Users a: docker attach (docker attach %container.id)
System Calls b: bash shell (docker exec -i -t %container.id /bin/bash)
Threads f: follow logs (docker logs -f %container.id)
Traces List h: image history (docker history %container.image)
Traces Spectrogram i: docker inspect (docker inspect %container.id)
Traces Summary k: docker kill (docker kill %container.id)
l: docker logs (docker logs %container.id)
s: docker stop (docker stop %container.id)
z: docker pause (docker pause %container.id)
u: docker unpause (docker unpause %container.id)
F1Help F2Views F4Filter F5Echo F6Dig F7Legend F8Actions F9Sort F12Spectro CTRL+FSearchp Pause 11/78 (14.1%)
```

Rysunek 8.6. Przykład użycia wtyczki kubectldig

```
Use "rakless [command] --help" for more information about a command.
tuxotron @ server ~
└─ $ ▶ kubectl access-matrix -n default
NAME LIST CREATE UPDATE DELETE
bindings ✓ ✓
configmaps ✓ ✓ ✓ ✓
controllerrevisions.apps ✓ ✓ ✓ ✓
cronjobs.batch ✓ ✓ ✓ ✓
daemonsets.apps ✓ ✓ ✓ ✓
daemonsets.extensions ✓ ✓ ✓ ✓
deployments.apps ✓ ✓ ✓ ✓
deployments.extensions ✓ ✓ ✓ ✓
endpoints ✓ ✓ ✓ ✓
events ✓ ✓ ✓ ✓
events.events.k8s.io ✓ ✓ ✓ ✓
horizontalpodautoscalers.autoscaling ✓ ✓ ✓ ✓
ingresses.extensions ✓ ✓ ✓ ✓
ingresses.networking.k8s.io ✓ ✓ ✓ ✓
jobs.batch ✓ ✓ ✓ ✓
leases.coordination.k8s.io ✓ ✓ ✓ ✓
limitranges ✓ ✓ ✓ ✓
localsubjectaccessreviews.authorization.k8s.io ✓ ✓ ✓ ✓
networkpolicies.extensions ✓ ✓ ✓ ✓
```

Rysunek 8.7. Przykład użycia wtyczki rakless

```
$ 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
```

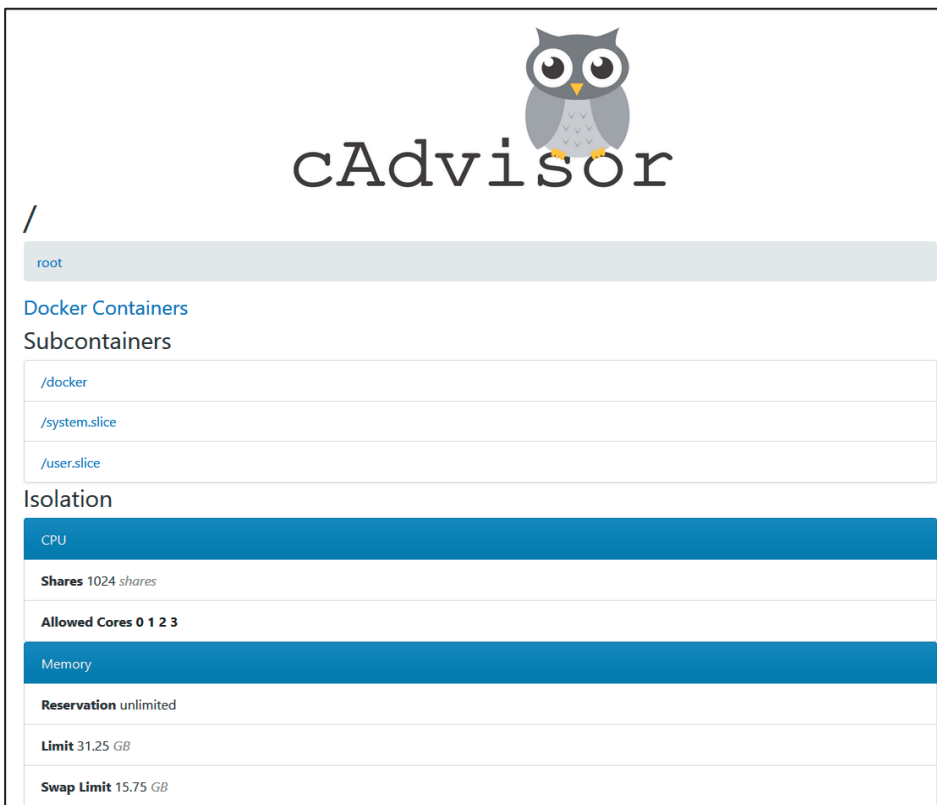
Rysunek 9.19. Zmienne środowiskowe kontenera Ubuntu zawierające informacje o kontenerze Elasticsearch

ctop - 13:51:59 UTC 3 containers							
NAME	CID	CPU	MEM	NET RX/TX	IO R/W	PIDS	
ctop	060a55c5244c	1%	9M / 31.4G	0B / 0B	0B / 0B	13	
laughing_greid...	c92da791b0af	0%	5M / 31.4G	0B / 0B	32K / 0B	2	
sad_bouman	db2de5b34fc2	-	-	-	-	-	

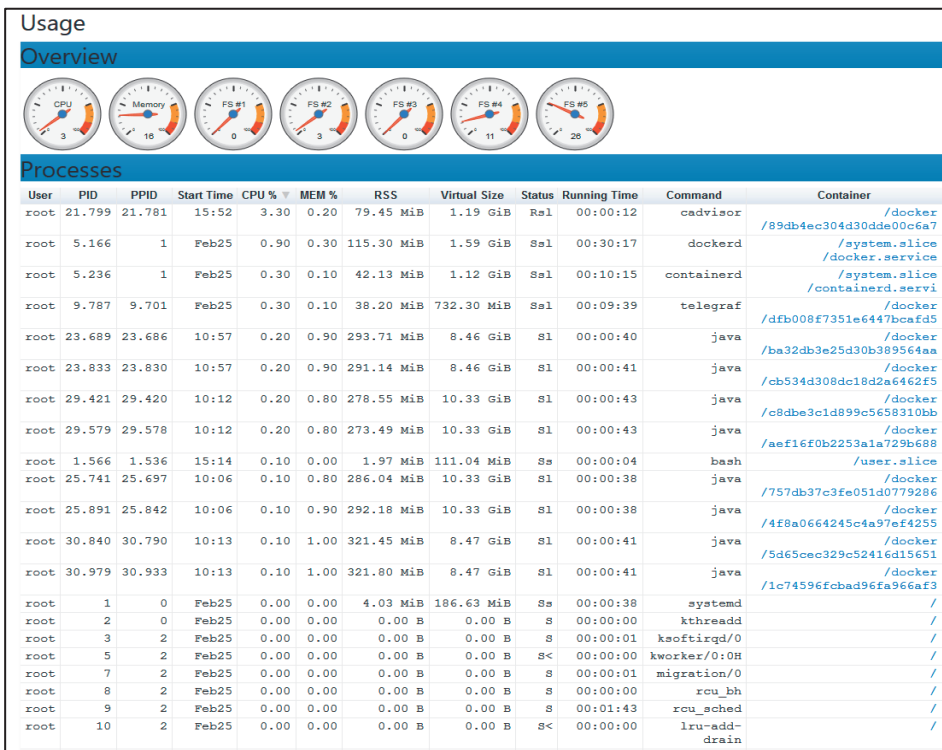
Rysunek 10.9. Przykład użycia narzędzia ctop

Menu 3 containers							
		CPU	MEM	NET RX/TX	IO R/W	PIDS	
[o] single view							
[l] log view							
[s] stop	0a55c5244c	1%	9M / 31.4G	0B / 0B	0B / 0B	13	
[p] pause	2da791b0af	0%	5M / 31.4G	0B / 0B	32K / 0B	2	
[r] restart	2de5b34fc2	-	-	-	-	-	
[e] exec shell							
[c] cancel							

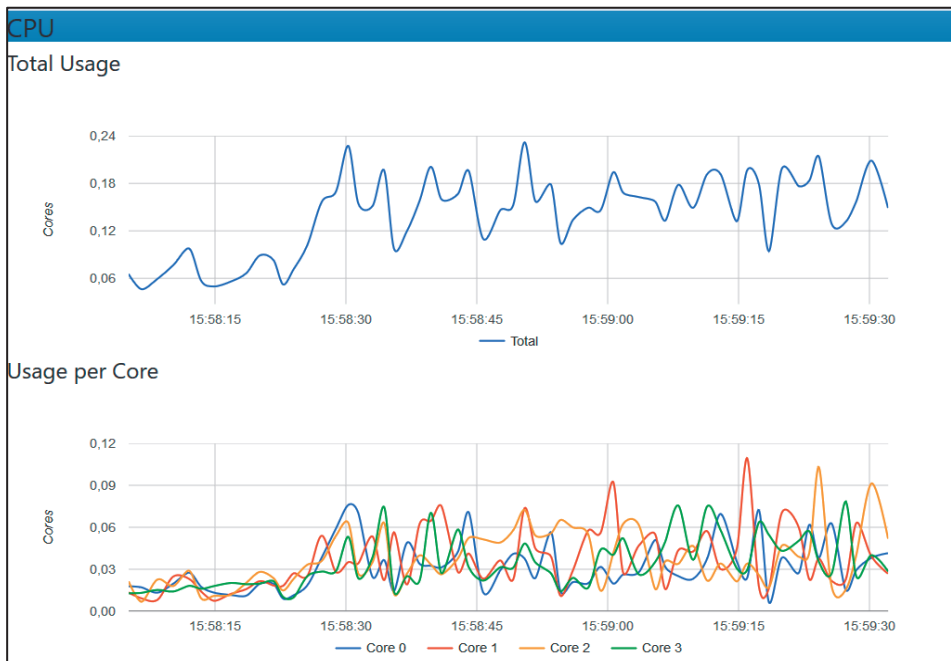
Rysunek 10.10. Opcje wizualizacji zdarzeń



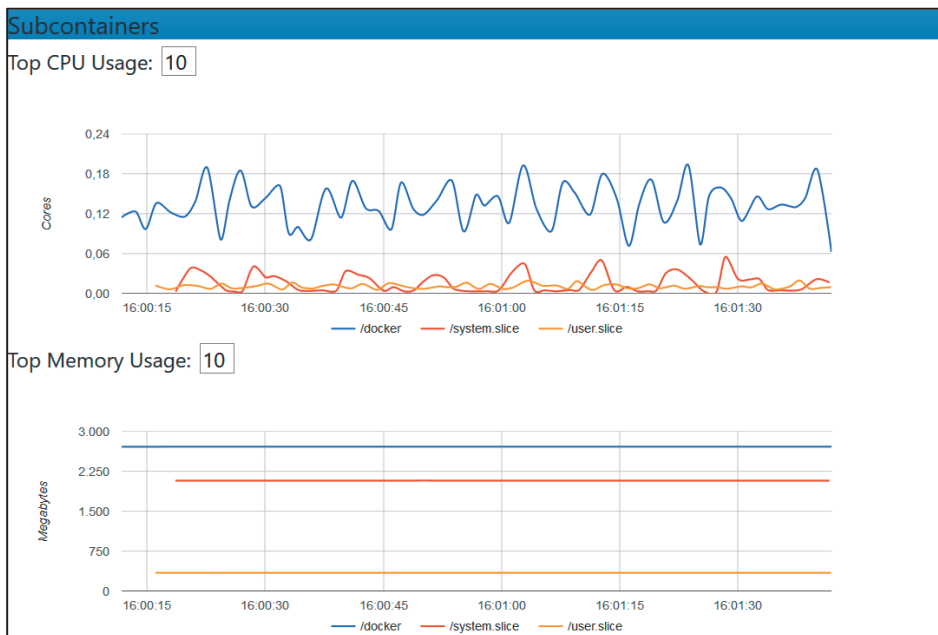
Rysunek 10.14. *Informacje o uruchomionych kontenerach*



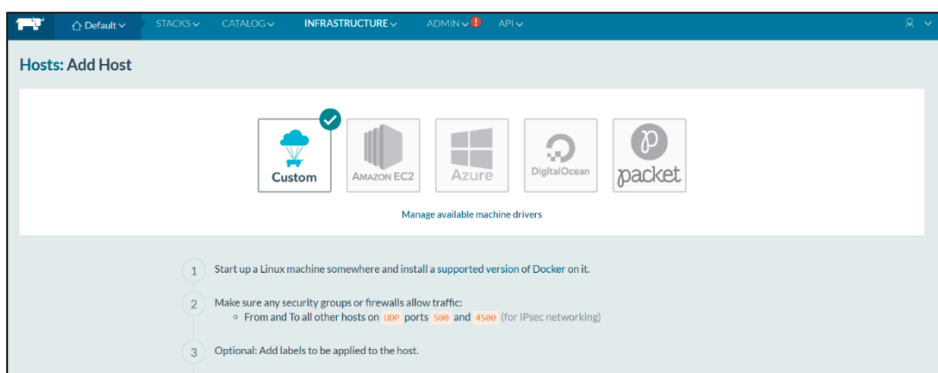
Rysunek 10.15. Informacje o procesach



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

1

Start up a Linux machine somewhere and install a supported version of Docker on it.

2

Make sure any security groups or firewalls allow traffic:

- From and To all other hosts on UDP ports 500 and 4500 (for IPsec networking)

3

Optional: Add labels to be applied to the host.

⊕

Add Label

4

Specify the public IP that should be registered for this host. If left empty, Rancher will auto-detect the IP to use. This generally works for machines with unique public IPs, but will not work if the machine is behind a firewall/NAT or if it is the same machine that is running the rancher/server container.

e.g. 1.2.3.4

5

Copy, paste, and run the command below to register the host with Rancher:

Copy to Clipboard

```
sudo docker run --rm --privileged -v /var/run/docker.sock:/var/run/docker.sock -v /var/lib/rancher:/var/lib/rancher rancher/agent:v1.2.11 https://2886795319-8080-cykoria03.environments.katacoda.com/v1/scripts/5445880E4F5E7FC5C78:1546214400000:GbqLiVowScu5Xi2iQDc1MykT8
```

Rysunek 11.2. Rejestrowanie hosta za pomocą narzędzia Rancher



Default

STACKS

CATALOG

INFRASTRUCTURE

ADMIN

API

Hosts

Add Host

ACTIVE

master

195.154.79.10 | 18.09.7

Ubuntu 16.04.6 LTS (4.4.0)

2x2.1 GHz | 1.95 GiB | 44.1 GiB

Stack: healthcheck

- healthcheck-1 10.42.76.44

Stack: ipsec

- cni-driver-1 None
- ipsec-1 10.42.135.94

Sidekicks

Rysunek 11.3. Informacje o hoście w interfejsie narzędzia Rancher

Add Environment

Name

j.g. lab

Description

e.g. Environment for developer experimentation

Environment Template

Cattle

Kubernetes

Mesos

Swarm

Windows

Orchestration: Cattle

Framework: Network Services, Scheduler, Healthcheck Service

Networking: Rancher IPsec

Rysunek 11.4. Szablon środowiska w narzędziu Rancher

Environment Templates

Add Template

An environment template allows users to define a different combination of infrastructure services to be deployed.

The infrastructure services includes but not limited to container orchestration (i.e. cattle, kubernetes, mesos, swarm, networking) or rancher services (i.e healthcheck, dns, metadata, scheduling, service discovery and storage)

Name	Description	Stacks	Public
Cattle	Default Cattle template	network-services, ipsec, scheduler, healthcheck	✓
Kubernetes	Default Kubernetes template	kubernetes, network-services, ipsec, healthcheck	✓
Mesos	Default Mesos template	mesos, network-services, ipsec, scheduler, healthcheck	✓
Swarm	Default Swarm template	portainer, swarm, network-services, ipsec, scheduler, healthcheck	✓
Windows	Experimental Windows template	windows, windows-network-services	✓

Rysunek 11.5. Szablony środowiska w narzędziu Rancher

Default

STACKS

CATALOG

INFRASTRUCTURE

ADMIN

API

Catalog: All

Search...

Category: All

Manage

Alfresco

An ECM and BPM platform.

View Details

Alibaba Cloud DNS

Rancher External DNS service powered by Alibaba Cloud

View Details

Apache Guacamole

Apache Guacamole is a clientless remote desktop gateway. It supports standard protocols like VNC, RDP, and SSH.

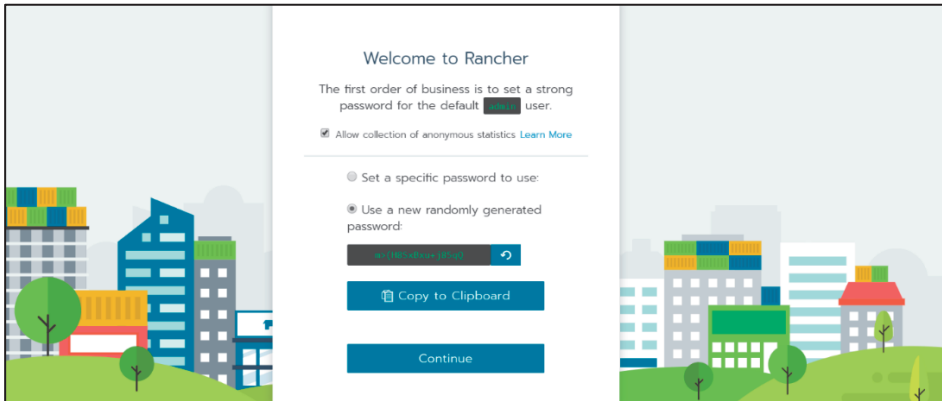
View Details

Apache Kafka

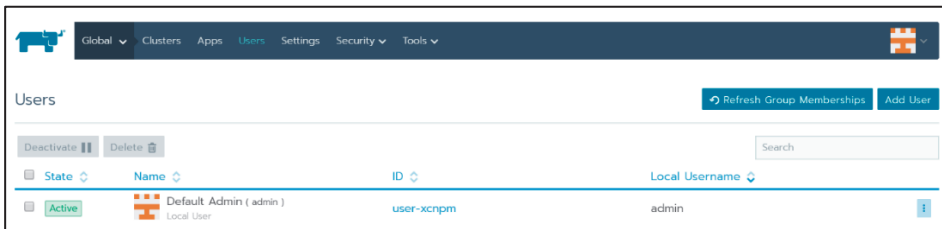
Kafka cluster

View Details

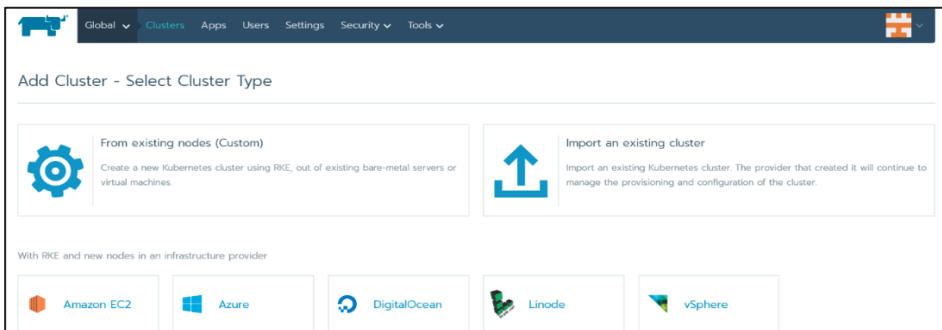
Rysunek 11.6. Katalog aplikacji w interfejsie narzędzia Rancher



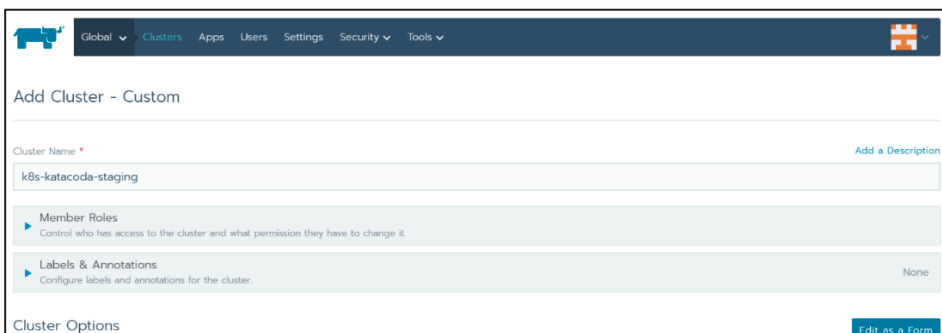
Rysunek 11.10. Strona powitalna narzędzia Rancher



Rysunek 11.11. Sekcja Users w narzędziu Rancher



Rysunek 11.12. Tworzenie klastra



Rysunek 11.13. Wybór typu klastra

Copy to ClipboardRead from a file

```
1 #
2 # Cluster Config
3 #
4 docker_root_dir: /var/lib/docker
5 enable_cluster_alerting: false
6 enable_cluster_monitoring: false
7 enable_network_policy: false
8 local_cluster_auth_endpoint:
9   enabled: true
10 name: k8s-katacoda-staging
11 #
12 # Rancher Config
13 #
14 rancher_kubernetes_engine_config:
15   addon_job_timeout: 30
16   authentication:
17     strategy: x509
18   ignore_docker_version: true
19 #
```

Rysunek 11.14. Plik konfiguracyjny klastra

1

Node Options

Choose what roles the node will have in the cluster

Node Role

☒ etcd

☒ Control Plane

☒ Worker

Show advanced options

2

Run this command on one or more existing machines already running a supported version of Docker.

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.3.2 --server https://2886795332-80-cykoria01.environments.katacoda.com --token hgj6qvlgm5grnlz4jj5qslvtkdjh7x67j2wrzp6vxj95kkrmvt --ca-checksum c02cc0b5a7203f0c0c4b29bcd14af55c1c2ed73949c9b974746ecc3397a6cd3 --etcd --controlplane --worker
```

Rysunek 11.15. Konfiguracja węzła klastra

This cluster is currently **Provisioning**, areas that interact directly with it will not be available until the API is ready.


Waiting for etcd and controlplane nodes to be registered

Edit Cluster

Nodes

Delete	Search					
State	Name	Roles	Version	CPU	RAM	Pods
Registering	master 172.17.0.68	Worker	n/a	n/a	n/a	n/a
Waiting to register with Kubernetes						

Rysunek 11.16. Rejestracja głównego węzła klastra



Global

Clusters


Apps

Users

Settings

Security

Tools





Clusters

Add Cluster

Delete

Search

State	Cluster Name	Provider	Nodes	CPU	RAM	
 Provisioning	k8s-katacoda-staging	Custom	1	n/a	n/a	

Waiting for etcd and controlplane nodes to be registered

Rysunek 11.17. Stan klastra w interfejsie narzędzia Rancher



The screenshot shows the Portainer.io logo at the top. Below it, a message says "Please create the initial administrator user." There are three input fields: "Username" with the value "admin", "Password" (empty), and "Confirm password" (empty). A red error message "The password must be at least 8 characters long" is displayed below the password fields. A "Create user" button is at the bottom.

Please create the initial administrator user.

Username

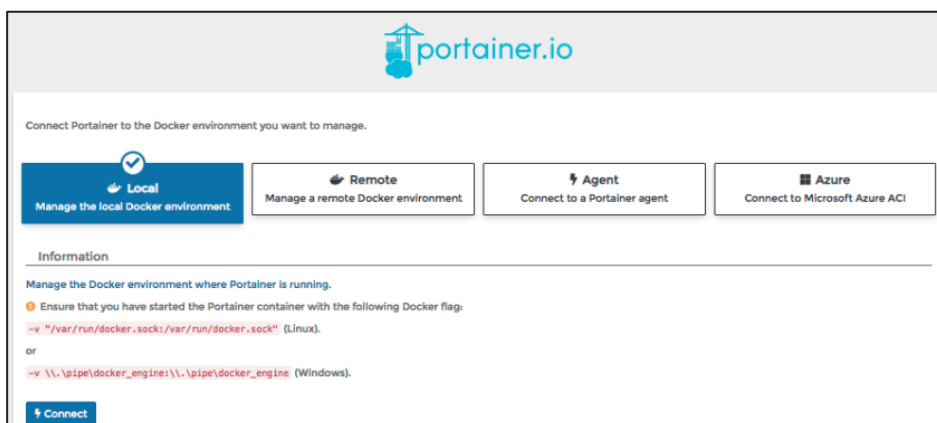
Password

Confirm password

✖ The password must be at least 8 characters long

[Create user](#)

Rysunek 11.19. Definiowanie konta administratora



The screenshot shows the Portainer.io logo at the top. Below it, a message says "Connect Portainer to the Docker environment you want to manage." There are four buttons: "Local" (selected), "Remote", "Agent", and "Azure". Below the buttons, there is an "Information" section with instructions on how to connect to the Docker environment. A "Connect" button is at the bottom.

Connect Portainer to the Docker environment you want to manage.

[Local](#) [Remote](#) [Agent](#) [Azure](#)

Manage the local Docker environment Manage a remote Docker environment Connect to a Portainer agent Connect to Microsoft Azure ACI

Information

Manage the Docker environment where Portainer is running.

Ensure that you have started the Portainer container with the following Docker flag:

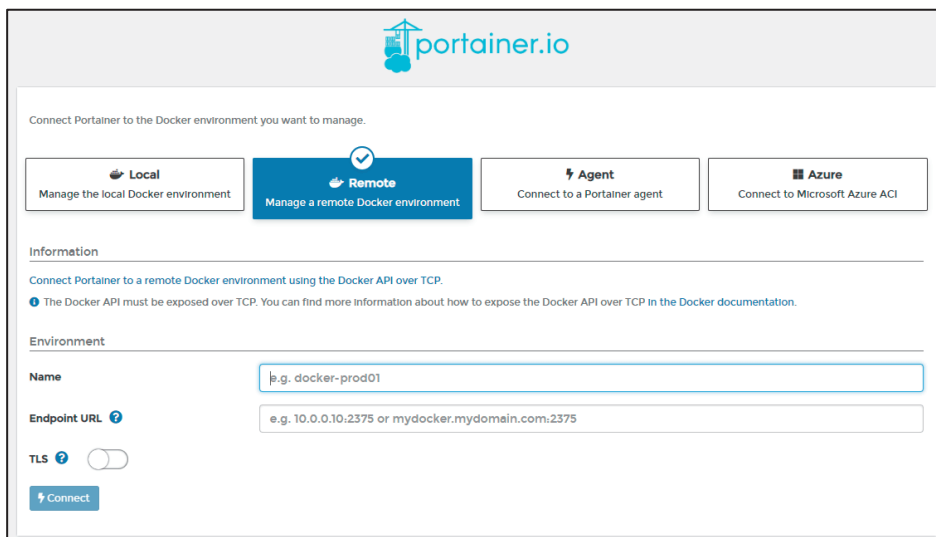
`-v "/var/run/docker.sock:/var/run/docker.sock"` (Linux).

or

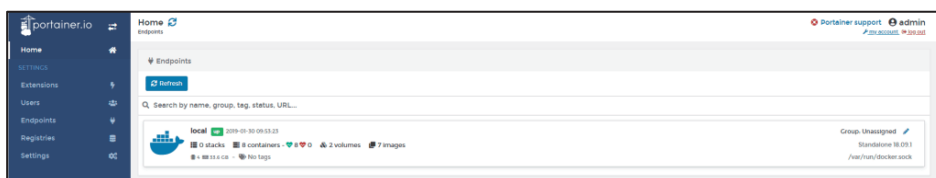
`-v \\.\pipe\docker_engine:\\.\pipe\docker_engine` (Windows).

[Connect](#)

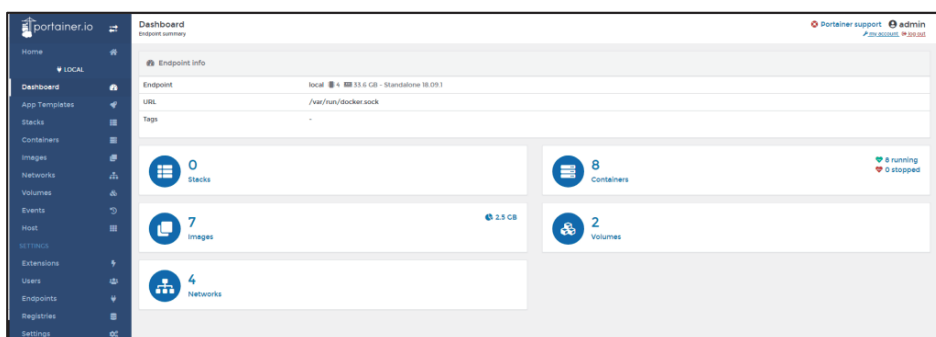
Rysunek 11.20. Lokalna instalacja narzędzia Portainer



Rysunek 11.21. Zarządzanie zewnętrznym środowiskiem za pomocą narzędzia Portainer



Rysunek 11.22. Interfejs graficzny narzędzia Portainer



Rysunek 11.23. Główna strona narzędzia Portainer

Container list

Portainer support admin

Containers

Columns Settings

Start Stop Kill Restart Pause Resume Remove Add container

Search...

Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
portainer	running		-	portainer/portainer	2019-01-30 09:51:42	172.17.0.8	9000:9000	administrators
traefik_proxy	running		-	traefik	2019-01-28 14:20:18	172.18.0.2	80:80 8080:8080	administrators
telegraf	running		-	stuartalegraf/latest	2019-01-24 17:16:34	172.17.0.7	-	administrators
	running		-	tomcat7war	2019-01-23 16:05:07	172.17.0.3	9003:9009 9005:8080	administrators
	running		-	tomcat7war	2019-01-23 16:02:07	172.17.0.2	9004:8009 9004:8080	administrators
	running		-	tomcat7war	2019-01-23 15:56:06	172.17.0.6	9005:8009 9005:8080	administrators
	running		-	tomcat7war	2019-01-23 15:49:07	172.17.0.5	9002:8009 9002:8080	administrators
	running		-	tomcat7war	2019-01-23 15:40:07	172.17.0.4	9001:8080 9001:8009	administrators

Items per page 10

Rysunek 11.24. Lista kontenerów w interfejsie narzędzia Portainer

Container details

Portainer support admin

Containers > nginx-web

my account log out

Actions

Start Stop Kill Restart Pause Resume Remove Recreate Duplicate/Edit

Container status

ID	e2b1b62ee25363a264af65ee7f2f68cb416066d5de219b1d8c3ea830957c335e
Name	nginx-web
IP address	172.17.0.2
Status	Running for 6 minutes
Created	2019-11-13 16:15:20
Start time	2019-11-13 16:15:20

Logs Inspect Stats Console Attach

Rysunek 11.25. Szczegóły kontenera w interfejsie narzędzia Portainer

Log viewer settings

Auto-refresh logs

Wrap lines

Display timestamps

Fetch All logs

Search Filter...

Lines 100

Copy

Copy selected lines

Unselect

```

* directory[/opt/gitlab/sv/grafana/control] action create (up to date)
* link[/opt/gitlab/init/grafana] action create (up to date)
* file[/opt/gitlab/sv/grafana/down] action delete (up to date)
* directory[/opt/gitlab/service] action create (up to date)
* link[/opt/gitlab/service/grafana] action create (up to date)
* ruby_block[wait for grafana service socket] action run (skipped due to not_if)
- execute the ruby block reload log service

```

Rysunek 11.26. Szczegółowe informacje o dziennikach kontenera

Container inspect

Containers > nginx > Inspect

Portainer supportadminmy accountlog out

Inspect

TreeText

▼ 9c52919fc46953034a7b54ba15520a816b62e875f4c1b66967841066c9e550d7:

AppArmorProfile: docker-default

▶ Args: [-g, daemon off,]

▶ Config: { ArgsEscaped: true, AttachStderr: false, AttachStdin: false, AttachCreated: 2019-11-13T16:15:34.455100998ZDriver: overlay2ExecIDs:

▶ GraphDriver: { Data: [object Object], Name: overlay2 }

HostConfig: { AutoRemove: false, Blinds: 21e33b5cd62877c370d2a7a897257142fHostnamePath: /var/lib/docker/containers/9c52919fc46953034a7b54ba15520a816b62e875f4c1b66967841066c9e550d7/hostnameHostsPath: /var/lib/docker/containers/9c52919fc46953034a7b54ba15520a816b62e875f4c1b66967841066c9e550d7/hostsId: 9c52919fc46953034a7b54ba15520a816b62e875f4c1b66967841066c9e550d7Image: sha256:540a289bab6cb1bf880086a9b803f0c4cfe38cbb5cdefa199b69614525199fLogPath: /var/lib/docker/containers/9c52919fc46953034a7b54ba15520a816b62e875f4c1b66967841066c9e550d7/9c52919fc46953034a7b54ba15520a816b62e875f4c1b66967841066c9e550d7-json.logMountLabel:

▶ Mounts: [[object Object], [object Object]]Name: /nginx

Rysunek 11.27. Szczegółowe informacje o kontenerze

Container statistics

Containers > nginx > Stats

Portainer supportadminmy accountlog out

About statistics

This view displays real-time statistics about the container nginx as well as a list of the running processes inside this container.

Refresh rate5s

Memory usage

CPU usage

Network usage (aggregate)

Processes

Search...

Rysunek 11.28. Wskaźniki opisujące pracę kontenera

Images

Settings

RemoveBuild a new imageImportExport

Search...

IdFilter

Tags

Size

Created

sha256:f6e8af4562c14ab06a2c9f3698e39e...Unuseddockersamples/examplevotingapp_vote:<none>83.6 MB2017-01-11 02:54:06

sha256:2b1e6048c5398e19b011fc1b67c2d1...Unuseddockersamples/examplevotingapp_worker:<none>961.9 MB2017-04-07 21:31:15

sha256:540a289bab6cb1bf880086a9b803cf...nginx:latest126.2 MB2019-10-23 02:26:03

sha256:36726735dc3c2c86ff47a937c72d53...Unusedpostgres:<none>206.3 MB2019-10-17 06:40:54

Rysunek 11.29. Strona Images w narzędziu Portainer

Network list

Portainer support admin

my account log out

Networks

Settings

Remove Add network

Search...

	Name	Stack	Scope	Driver	Attachable	Internal	IPAM Driver	IPAM Subnet	IPAM Gateway	Ownership
<input type="checkbox"/>	bridge	-	local	bridge	false	false	default	172.17.0.0/16	-	administrators
<input type="checkbox"/>	docker_gwbridge	-	local	bridge	false	false	default	172.19.0.0/16	172.19.0.1	administrators
<input type="checkbox"/>	host	-	local	host	false	false	default	-	-	administrators
<input type="checkbox"/>	Ingress	-	swarm	overlay	false	false	default	10.255.0.0/16	10.255.0.1	administrators
<input type="checkbox"/>	none	-	local	null	false	false	default	-	-	administrators

Items per page

10

Rysunek 11.30. Lista sieci w interfejsie narzędzia Portainer

portainer.io

Portainer support admin

my account log out

Volume list

Volumes

Settings

Remove Add volume

Search...

	Name	Stack	Driver	Mount point	Created
<input type="checkbox"/>	21e33b5cd62877c370d2a7a897257142f8b1e...	-	local	/var/lib/docker/volumes/2[...]/_data	2019-11-13 17:15:34
<input type="checkbox"/>	3f5caacc6318d6eee36a9a4690fecbc25b376...	-	local	/var/lib/docker/volumes/3[...]/_data	2019-11-13 17:15:34

Items per page

10

Rysunek 11.31. Lista woluminów w interfejsie narzędzia Portainer

Application templates list

Portainer support admin

my account log out

Templates

Settings

Add template

Select a category

Show container templates

Search...

Portainer Agent

Manage all the resources in your Swarm cluster

Update Delete

portainer

OpenFaaS

Serverless functions made simple

Update Delete

serverless

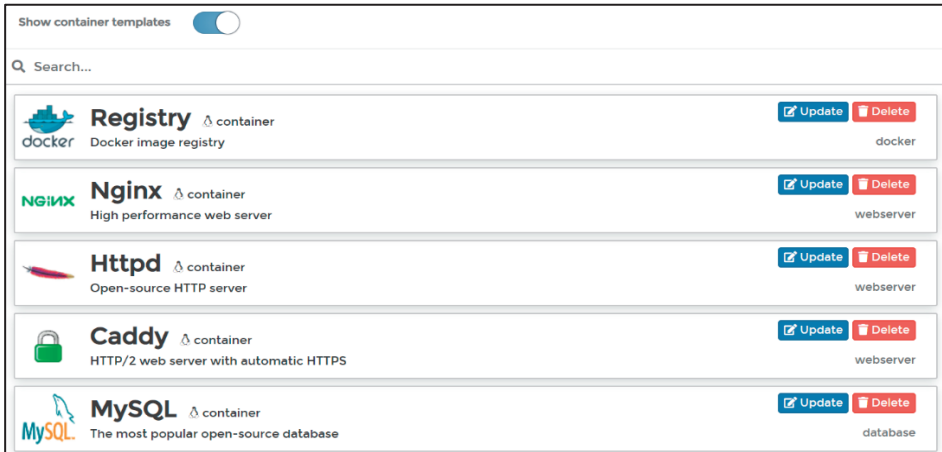
IronFunctions

Open-source serverless computing platform

Update Delete

serverless

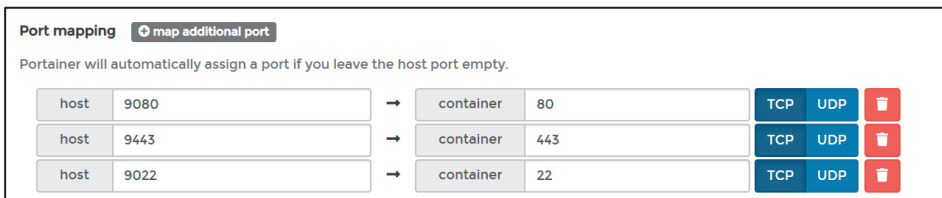
Rysunek 11.32. Szablony aplikacji w interfejsie narzędzia Portainer



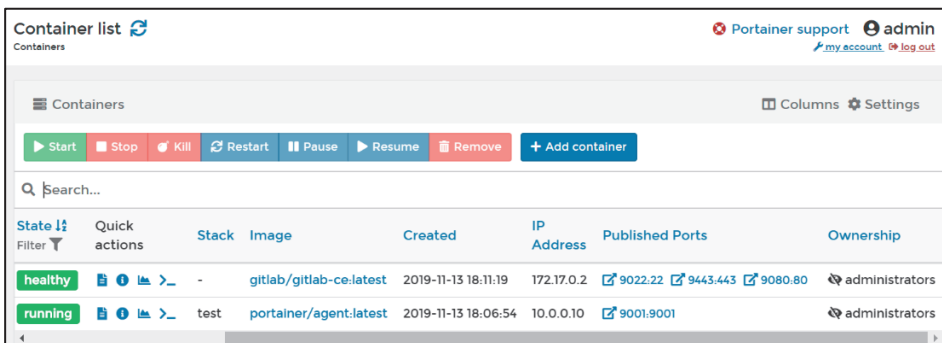
Rysunek 11.33. Szablony kontenerów w interfejsie narzędzia Portainer



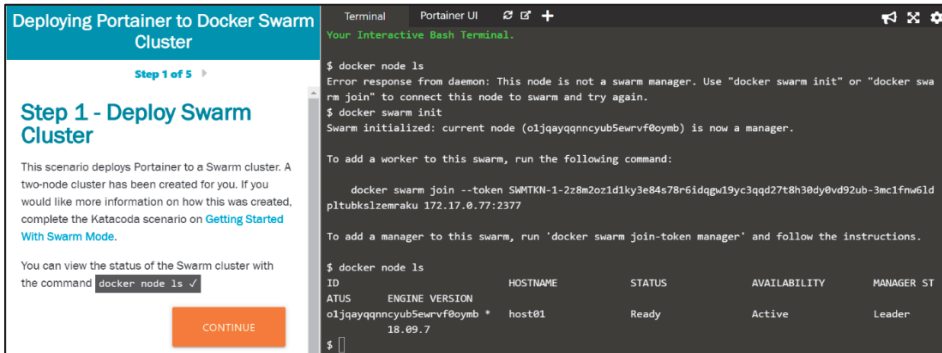
Rysunek 11.34. Szablony narzędzia GitLab w interfejsie narzędzia Portainer



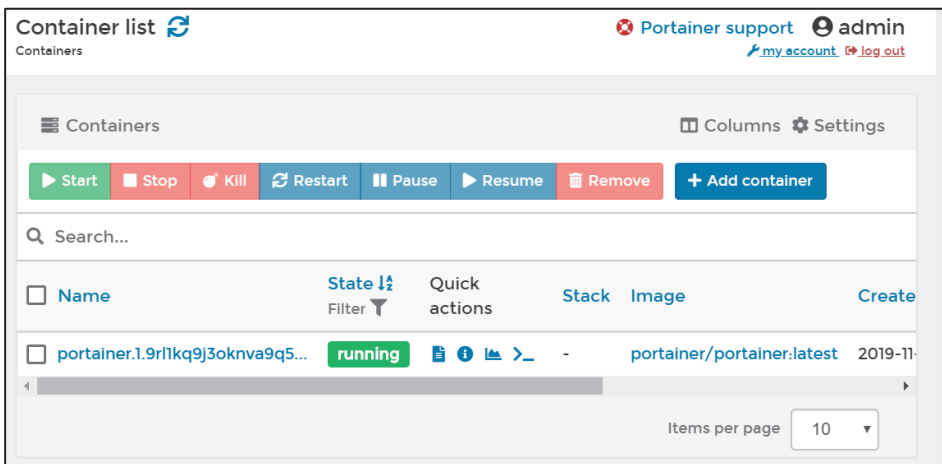
Rysunek 11.35. Wiązanie portów w interfejsie narzędzia Portainer



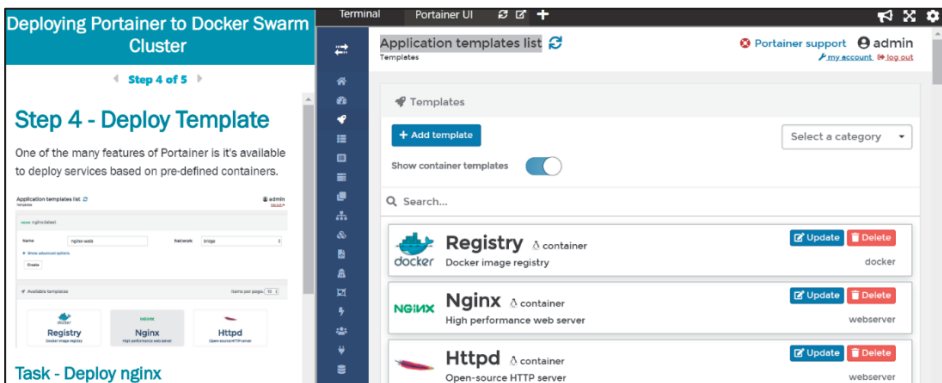
Rysunek 11.36. Lista kontenerów w interfejsie narzędzia Portainer



Rysunek 11.37. Wdrożenie narzędzia Portainer w klastrze Docker Swarm



Rysunek 11.38. Uruchomiony kontener Portainer



Rysunek 11.39. Wdrożenie szablonu serwera Nginx

Configuration

Name

nginx-web

Network

bridge

Access control

Enable access control

?

Administrators

I want to restrict the management of this resource to administrators only

Restricted

I want to restrict the management of this resource to a set of users and/or teams

Hide advanced options

Port mapping

map additional port

Portainer will automatically assign a port if you leave the host port empty.

host

80

→

container

80

TCP

UDP

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
0lw40ii82gslxgfjtm5h2ncr *	docker-master1	Ready	Active	Leader
1vvjz6c4bfe4ez16kzkyuc3io	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
rhjxlrqimj7      portainer_agent     global              4/4                 portainer/agent:latest
rs5wrgflhmax     portainer_portainer replicated          1/1                 portainer/portainer:lates
oxj7bf6e36kj     tomcat_tomcat      global              4/4                 tomcat:8.0
```

Rysunek 11.43. Lista usług w klastrze Docker Swarm

Services					
<div><div>Update</div><div>Remove</div><div>+ Add service</div></div>					
Q Search...					
<input type="checkbox"/>	> Name <i>Id</i>	Stack	Image	Scheduling Mode	Published Ports
<input type="checkbox"/>	> httpd_httpd	httpd	httpd:2.4	global <i>4 / 4</i>	80:80 443:443
<input type="checkbox"/>	> jboss_jboss	jboss	jboss/wildfly:latest	global <i>4 / 4</i>	8080:8080 9990:9990
<input type="checkbox"/>	> portainer_agent	portainer	portainer/agent:latest	global <i>4 / 4</i>	-
<input type="checkbox"/>	> portainer_portainer	portainer	portainer/portainer:latest	replicated <i>1 / 1</i> Scale	9000:9000
<input type="checkbox"/>	> tomcat_tomcat	tomcat	tomcat:8.0	global <i>4 / 4</i>	8180:8080

Rysunek 11.44. Aktywne usługi w klastrze Docker Swarm

Cluster status

Nodes

Docker API version

Total CPU

Total memory

Go to cluster visualizer

Nodes

Search...

Name ID	Role	CPU	Memory	Engine	IP Address	Status
docker-master1	manager	1	2.1 GB	18.09.1	192.168.1.131	ready
docker-master2	manager	1	2.1 GB	18.09.1	192.168.1.132	ready
docker-worker1	worker	1	1 GB	18.09.1	192.168.1.133	ready
docker-worker2	worker	1	1 GB	18.09.1	192.168.1.134	ready

Rysunek 11.45. Węzły tworzące klastr Docker Swarm