



Docker

Інструменти для Docker




Docker

Introduction



Олег Заревич
Senior DevOps Engineer

 Oleg Zarevych

 OlegZarevych

 OlegZarevych



Docker

Тема уроку

Інструменти для Docker

Docker

План уроку

1. Аналіз Docker образу
2. stop
3. dockly
4. dive
5. Lazydocker
6. hadolint
7. checkov

Docker

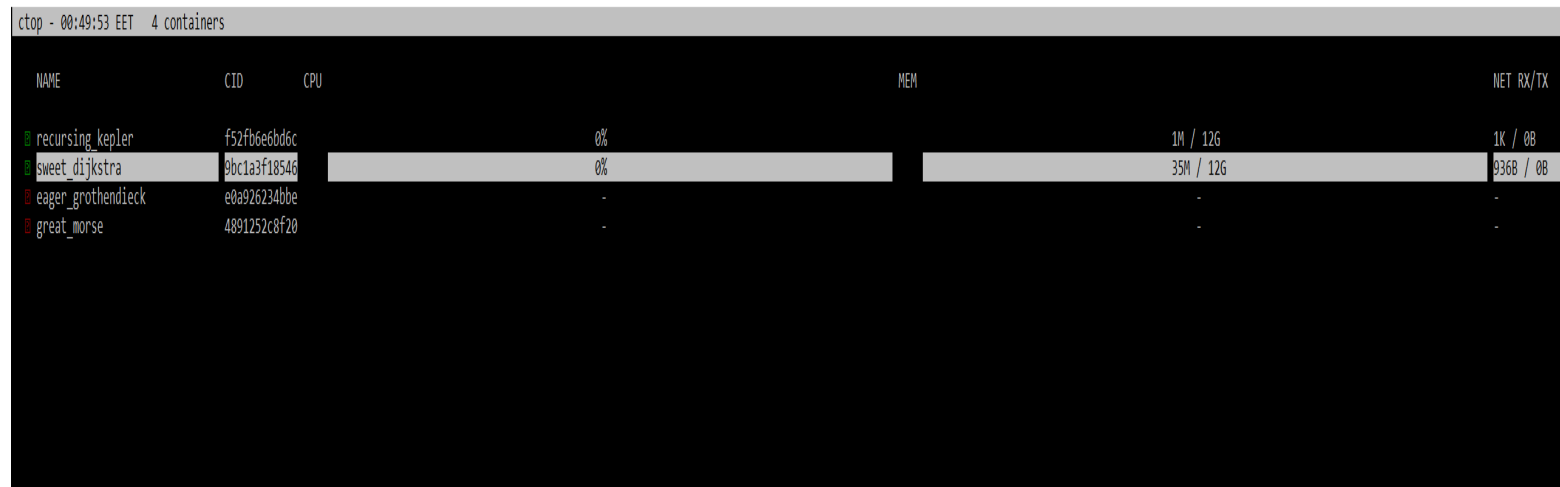
Аналіз Docker

- `docker inspect image_name`
- `docker scan image_name`

Docker

ctop

- Моніторинг контейнерів у реальному часі
- <https://github.com/bcicen/ctop>



NAME	CID	CPU	MEM	NET RX/TX
recurring_kepler	f52fb6e6bdc	0%	1M / 12G	1K / 0B
sweet_dijkstra	9bc1a3f18546	0%	35M / 12G	936B / 0B
eager_grothendieck	e0a926234bbe	-	-	-
great_morse	4891252c8f20	-	-	-

Docker

dockly

- Термінал для роботи з Docker
- <https://github.com/lirantal/dockly>

Containers		
Id	Name	Image
9bc1a	/sweet_dijkstra	py-hello
f5276	/recurring_kepler	ubuntu
e8a92	/eager_grothendieck	ubuntu
48912	/great_morse	ubuntu

dive

- Інструмент дослідження для Docker
- <https://github.com/wagoodman/dive>

```

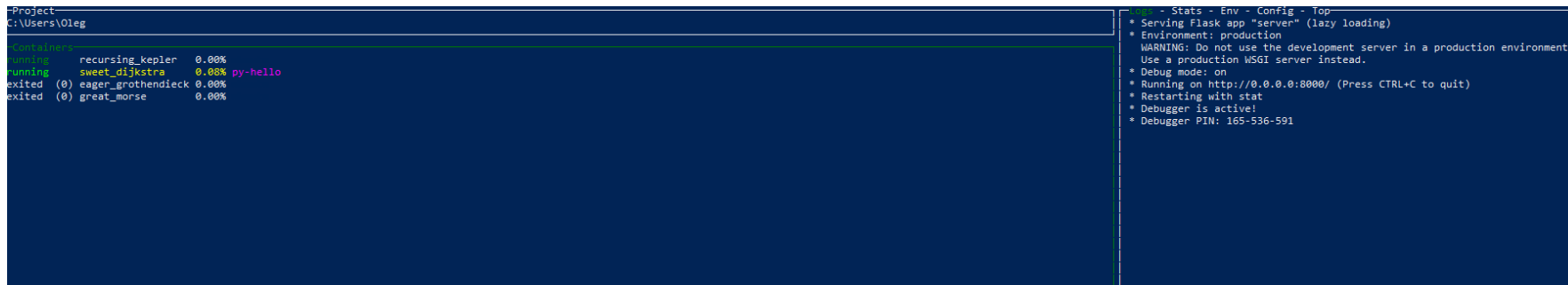
Layers |
+-----+
mp      Size  Command
5.5 MB FROM 128866eddcccc40
1.8 MB set -eux; Bapkg add --no-cache @lca-certificates @tzdata @;
30 MB set -eux; Bapkg add --no-cache --virtual build-deps @gnuip @tar @x: @bluez-dev @dbus2-dev @dpkg-dev @expat-dev @findutils @gcc @gdbm-dev @libc-dev @libffi-dev @libnl-dev @libtirpc-dev @linux-headers @make @ncurses-dev @open
0 B set -eux; Bfor src in $(ls pydoc3 python3-config); do Bdst=$(echo $src | tr -d $); B@ -s "/usr/local/bin/$dst" ); Bln -svl $src "/usr/local/bin/$dst"; Bdone
10 MB set -eux; Bwget -O get-pip.py "$PYTHON_GET_PIP_URL"; Becho "$PYTHON_GET_PIP_SHA256" get-pip.py | sha256sum -c -; Bexport PYTHONNOXTHLIBPYECHO=1; Bpython get-pip.py B--disable-pip-version-check B--no-cache-dir B--no-compile Bpip=PYTHONL
0 B WORKDIR /usr/src/app
98 B COPY requirements.txt . # buildkit
12 MB RUN /bin/sh -c pip install -qr requirements.txt # buildkit
269 B COPY server.py . # buildkit
-----+-----+
Layer Details |
Tags: (unavailable)
Id: 7079ad9dc4110b21e7be5150ee5f8d6674f6311fd043e392d682eb660477a0
Digest: sha256:5057f54009668d1d3c891cef2a3c6607367833bf5496247b157d477e3d62560
Command:
COPY server_bye.py . # buildkit
-----+-----+
Image Details |
Total Image size: 60 MB
Potential wasted space: 651 kB
Image efficiency score: 99 %
-----+-----+
Count  Total Space  Path
2      428 kB      /etc/ssl/certs/ca-certificates.crt
3      184 kB      /lib/apk/db/installed
3       36 kB      /lib/apk/db/scripts.tar
3      500 B      /lib/apk/db/triggers
3      255 B      /etc/apk/world
2       0 B      /usr/bin/ncat
2       0 B      /usr/bin/ncat
2       0 B      /usr/bin/find
2       0 B      /usr/bin/mk
3       0 B      /tmp
2       0 B      /bin/tar
3       0 B      /var/cache/misc
2       0 B      /usr/bin/strings
2       0 B      /usr/bin/unlzip
3       0 B      /lib/apk/db/lock
2       0 B      /usr/bin/unxz
2       0 B      /usr/bin/wget
2       0 B      /usr/bin/lzma

```


Docker

lazydocker

- Ще одна консоль для роботи з контейнерами
- <https://github.com/jesseduffield/lazydocker>



The screenshot displays the lazydocker application interface. On the left, a list of containers is shown with their status and CPU usage:

Container	Status	CPU Usage
recursing_kepler	running	0.00%
sweet_dijkstra	running	0.00%
eager_grothendieck	exited (0)	0.00%
great_morse	exited (0)	0.00%

On the right, a terminal window shows the output of the application, including a warning about the production environment and the Flask app status:

```
* Serving Flask app "server" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 165-536-591
```

Docker

hadolint

- Простий лінтер для Dockerfile
- <https://github.com/hadolint/hadolint>

```
PS C:\Users\Oleg\source\repos\Trainings\docker-py-hello-world> hadolint Dockerfile
Dockerfile:5 DL3013 warning: Pin versions in pip. Instead of `pip install <package>` use `pip install <package>==<version>` or `pip install --requirement <requirements file>`
Dockerfile:5 DL3042 warning: Avoid use of cache directory with pip. Use `pip install --no-cache-dir <package>`
PS C:\Users\Oleg\source\repos\Trainings\docker-py-hello-world>
```

Docker

checkov

- Checkov – статичний аналізатор коду, в тому числі і Dockerfile-y
- <https://github.com/bridgecrewio/checkov>

```
C:\Users\Oleg\source\repos\Trainings\docker-py-hello-world> checkov --file Dockerfile
File association not found for extension .py
[+] sca_image framework ]: 100%|██████████| [[1/1], Current File Scanned=Dockerfile
[+] dockerfile framework ]: 100%|██████████| [[1/1], Current File Scanned=Dockerfile
[+] secrets framework ]: 100%|██████████| [[1/1], Current File Scanned=Dockerfile

checkov
by bridgecrew.io | version: 2.2.31

Passed checks: 6, Failed checks: 2, Skipped checks: 0

Check: CKV_DOCKER_11: "Ensure From Alias are unique for multistage builds."
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-docker-from-alias-is-unique-for-multistage-builds

Check: CKV_DOCKER_7: "Ensure the base image uses a non latest version tag"
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-the-base-image-uses-a-non-latest-version-tag

Check: CKV_DOCKER_1: "Ensure port 22 is not exposed"
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-port-22-is-not-exposed

Check: CKV_DOCKER_9: "Ensure that APT isn't used"
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-docker-apt-is-not-used

Check: CKV_DOCKER_5: "Ensure update instructions are not use alone in the Dockerfile"
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-update-instructions-are-not-used-alone-in-the-dockerfile

Check: CKV_DOCKER_10: "Ensure that WORKDIR values are absolute paths"
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-docker-workdir-values-are-absolute-paths

Check: CKV_DOCKER_2: "Ensure that HEALTHCHECK instructions have been added to container images"
PASSED for resource: Dockerfile
Guide: https://docs.bridgecrew.io/docs/ensure-that-healthcheck-instructions-have-been-added-to-container-ima

1 | FROM python:3.9-alpine
2 | WORKDIR /usr/src/app
3 | EXPOSE 8080
4 | COPY requirements.txt .
5 | RUN pip install -qr requirements.txt
6 | COPY server.py .
7 | COPY server_bye.py .
8 | ENTRYPOINT [ "python3" ]
9 | CMD [ "./server.py" ]
10 |
11 |
```

Інформаційний відеосервіс для розробників програмного забезпечення



Перевірка знань

TestProvider.com



Перевірте як Ви засвоїли даний матеріал на [TestProvider.com](https://testprovider.com)

TestProvider – це online сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT спеціаліста.

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.

Docker

Дякую за увагу! До нових зустрічей!



Олег Заревич
Senior DevOps Engineer

