Docker learnings

Learnings from using Docker after 3 years of usage

Jan Baer

30. April 2019

This talk is not about Kubernetes or Docker Swarm

Security

Security

- You shouldn’t run your container as root

Security

- You shouldn’t run your container as root
  - Create user in Dockerfile

Security

- You shouldn't run your container as root
  - Create user in Dockerfile
  - Take care about **uid**, map to an existing uid on the host if needed

Security

- You shouldn't run your container as root
  - Create user in Dockerfile
  - Take care about **uid**, map to an existing uid on the host if needed
- Give only the privileges you really need

Security

- You shouldn't run your container as root
  - Create user in Dockerfile
  - Take care about **uid**, map to an existing uid on the host if needed
- Give only the privileges you really need
  - Docker documentation

```
docker run -d --cap-drop CHOWN alpine
```

Security

- You shouldn't run your container as root
  - Create user in Dockerfile
  - Take care about **uid**, map to an existing uid on the host if needed
- Give only the privileges you really need
  - Docker documentation

```
docker run -d --cap-drop CHOWN alpine
```

- Using **tmpfs** for sensitive data which shouldn't be saved outside of the container

```
docker run ... \
--tmpfs /tmp/${CONTAINER_NAME}:uid=1000,gid=1000 \
  ...
```

Building images

How to build smaller images

- Use multistage builds

How to build smaller images

- Use multistage builds
- Use build cache (copy package.json and yarn.lock in an extra step before yarn install)

How to build smaller images

- Use multistage builds
- Use build cache (copy package.json and yarn.lock in an extra step before yarn install)
- Remove deveveloper node_modules before copy

Maintaining

Cleanup

- Remove dangling images with **docker image prune**
- Remove stopped containers with **docker container prune**
- Same for network and volume or all in once with **docker system prune**
- Autoremove a container after it's stopped with **docker –rm...**

Keep an eye on your logfiles

How to grep the logs

```
docker logs {container} | grep {term}
```

This might will not work as expected

How to grep the logs

```
docker logs {container} | grep {term}
```

This might will not work as expected

This can happen if the container is logging to **stderr**. Piping works only for **stdout**. So you have to redirect **stderror** to **stdout** before your can pipe it to grep.

How to grep the logs

```
docker logs {container} | grep {term}
```

This might will not work as expected

This can happen if the container is logging to **stderr**. Piping works only for **stdout**. So you have to redirect **stderror** to **stdout** before your can pipe it to grep.

```
docker logs {container} 2>&1 | grep {term} | less
```

Take care about the size of your Docker log files

- When using JSON File logging driver (which is the default)
  - Using **/etc/docker/daemon.json**

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "10m",
    "max-file": "3"
  }
}
```

Take care about the size of your Docker log files

- Or using commandline option

```
docker run --rm -it --log-opt max-size=10m alpine
```

- docs.docker.com

Cleanup the Docker logs

To delete all log files, you can use the following command

```
find /var/lib/docker/containers/
    -type f -name "*.log" -delete
```

More tips - Using a UI in the browser or terminal

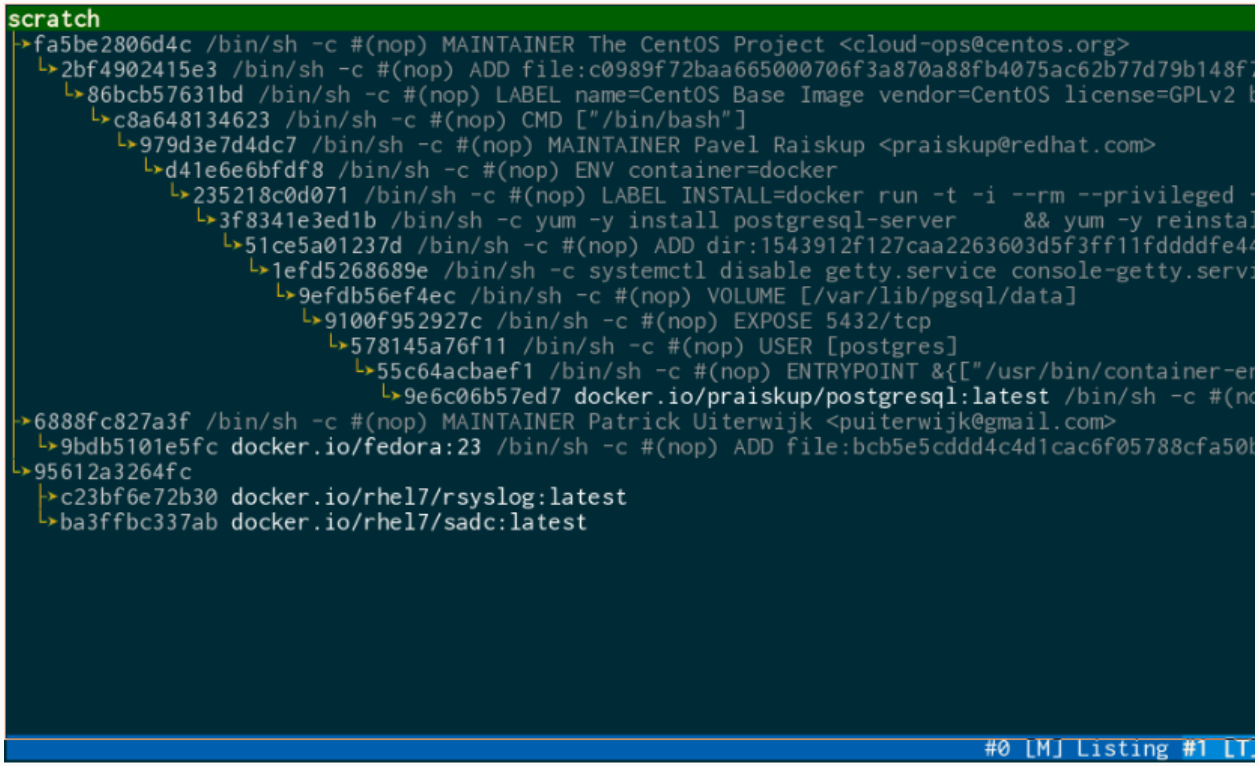
Use portainer locally without a password

```
alias portainer="docker run --rm -d
  -p 9000:9000
  --name=portainer
  -v /var/run/docker.sock:/var/run/docker.sock
  portainer/portainer --no-auth"
```

Hint: call it in an anonymus browser session

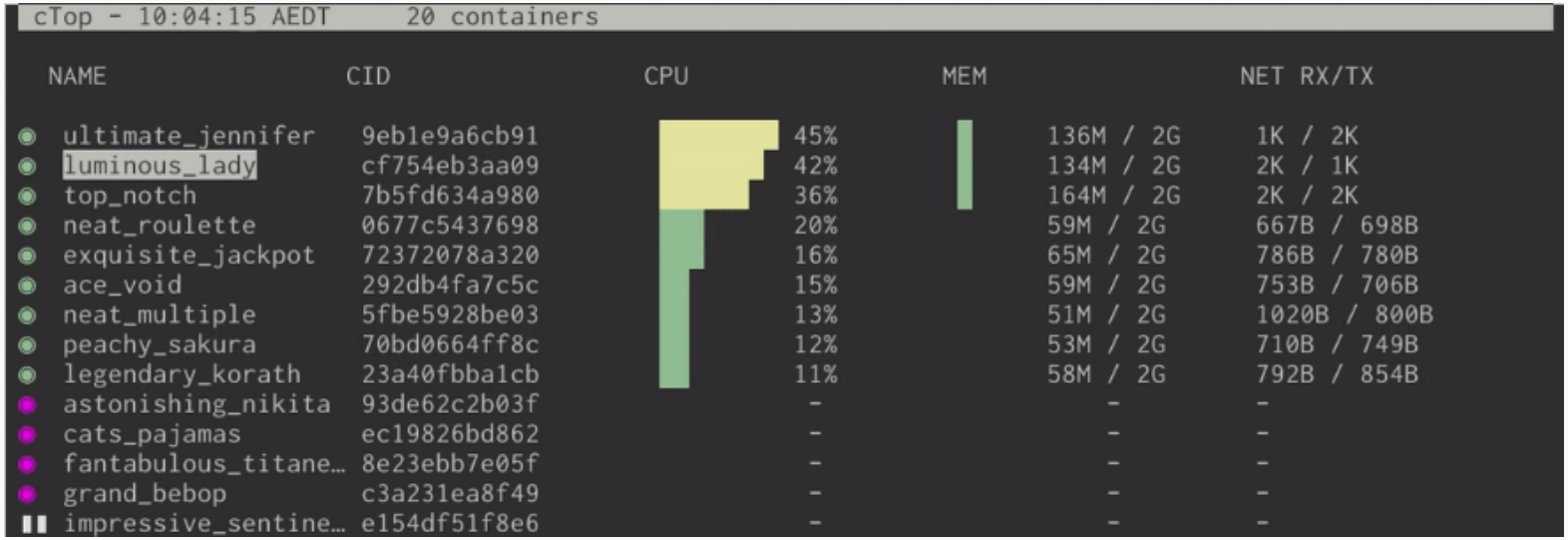
Use sen as terminal ui to inspect the layers of your images

```
alias sen="docker run --rm --name=sen
  -v /var/run/docker.sock:/run/docker.sock
  -ti -e TERM tomastomecek/sen"
```



Use ctop for monitoring your local containers

- github ctop



Last tipp

You won’t need to be sudo always

To prevent to make all Docker related operations with the root user, add yourself to the Docker group

```
sudo usermod -a -G docker $(whoami)
```

Thank you!