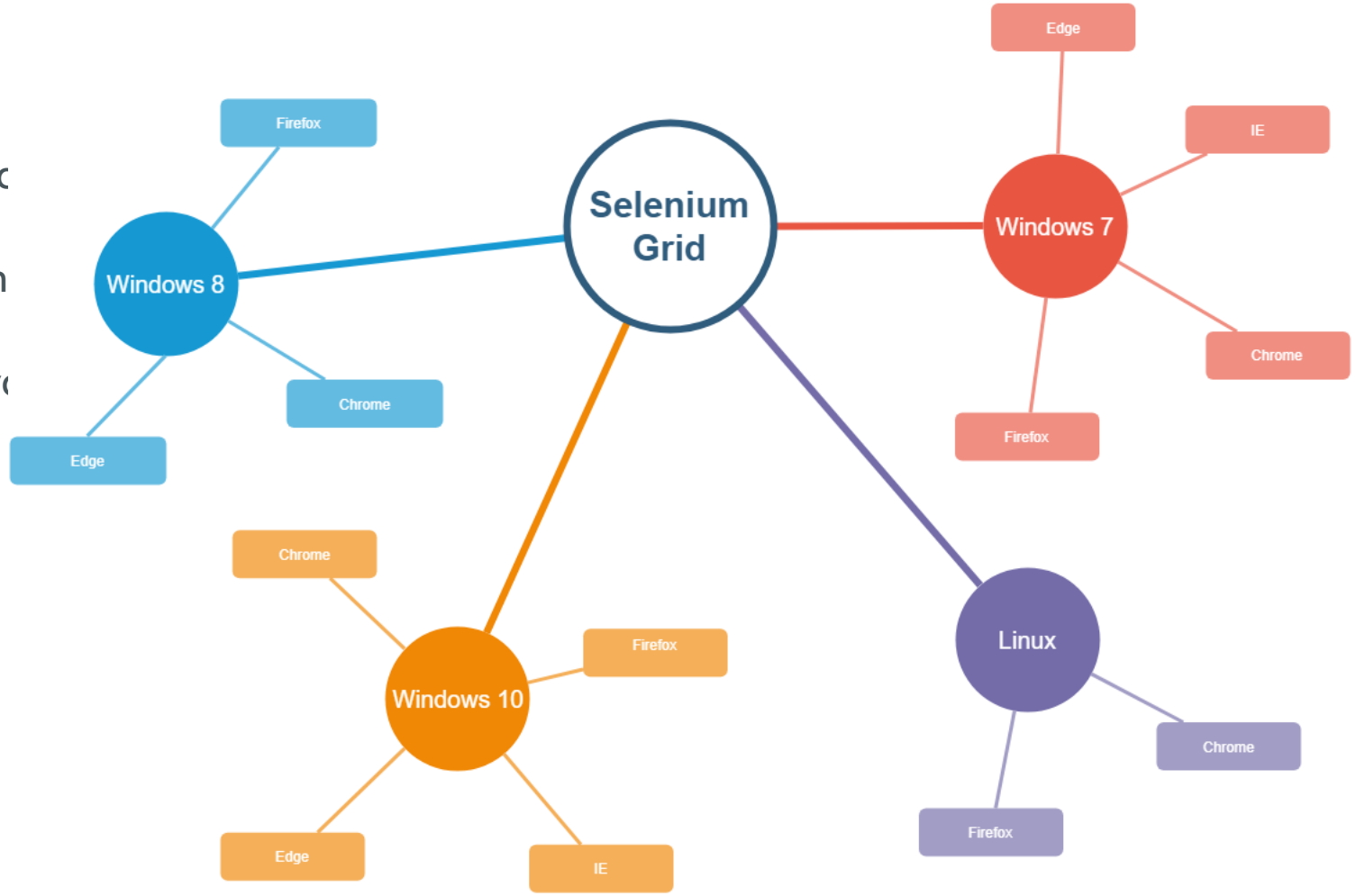
SELENIUM

ELŻBIETA SĄDEL

GRZEGORZ WITEK

SELENIUM GRID

- Dystrybuuje testy na wielu maszynac
- Zarządza środowiskami z jednego miejsca przez co ułatwia uruchamian testów na wielu kombinacjach przeglądarek i systemów operacyjny



SELENIUM SERVER

- Hub:

```
java -jar selenium-server-standalone-<version>.jar -role hub
```

```
09:05:10.966 INFO [GridLauncherV3.launch] - Selenium build info: version: '3.14.0', revision: 'aaccccе0'
09:05:10.972 INFO [GridLauncherV3$2.launch] - Launching Selenium Grid hub on port 4444
2018-09-05 09:05:11.336:INFO::main: Logging initialized @645ms to org.seleniumhq.jetty9.util.log.StdErrLog
09:05:11.560 INFO [Hub.start] - Selenium Grid hub is up and running
09:05:11.561 INFO [Hub.start] - Nodes should register to http://192.168.56.1:4444/grid/register/
09:05:11.562 INFO [Hub.start] - Clients should connect to http://192.168.56.1:4444/wd/hub
```

- Node:

```
java -jar selenium-server-standalone-<version>.jar -role node  -hub http://localhost:4444/grid/register
```

```
09:42:17.484 INFO [GridLauncherV3.launch] - Selenium build info: version: '3.14.0', revision: 'aaccccе0'
09:42:17.511 INFO [GridLauncherV3$3.launch] - Launching a Selenium Grid node on port 4993
2018-09-05 09:42:17.806:INFO::main: Logging initialized @716ms to org.seleniumhq.jetty9.util.log.StdErrLog
09:42:17.911 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 4993
09:42:17.911 INFO [GridLauncherV3$3.launch] - Selenium Grid node is up and ready to register to the hub
09:42:17.993 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
09:42:17.994 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://localhost:4444/grid/register
09:42:18.359 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
```

SELENIUM SERVER

Optional parameters

- `-port 4444` (4444 is default)
- `-host <IP | hostname>` specify the host name or IP. usually not needed and determined automatically. For exotic network configuration, network with VPN, specifying the host might be necessary.
- `-timeout 30` (300 is default) The timeout in seconds before the hub automatically releases a node that hasn't received any requests for more than the specified number of seconds. After this time, the node will be released for another test in the queue. This helps to clear client crashes without manual intervention. To remove the timeout completely, specify -timeout 0 and the hub will never release the node.

Note: This is NOT the WebDriver timeout for all ”wait for WebElement” type of commands.

- `-maxSession 5`(5 is default) The maximum number of browsers that can run in parallel on the node. This is different from the maxInstance of supported browsers (Example: For a node that supports Firefox 3.6, Firefox 4.0 and Internet Explorer 8, maxSession=1 will ensure that you never have more than 1 browser running. With maxSession=2 you can have 2 Firefox tests at the same time, or 1 Internet Explorer and 1 Firefox test).
- `-browser < params >` If -browser is not set, a node will start with 5 firefox, 1 chrome, and 1 internet explorer instance (assuming it’s on a windows box). This parameter can be set multiple times on the same line to define multiple types of browsers. Parameters allowed for -browser: browserName={android, chrome, firefox, htmlunit, internet explorer, iphone, opera} version={browser version} firefox_binary={path to executable binary} chrome_binary={path to executable binary} maxInstances={maximum number of browsers of this type} platform={WINDOWS, LINUX, MAC}
- `-registerCycle N` = how often in ms the node will try to register itself again.Allow to restart the hub without having to restart the nodes.
- Really large (>50 node) Hub installations may need to increase the jetty threads by setting -DPOOL_MAX=512 (or larger) on the java command line.

https://github.com/SeleniumHQ/selenium/wiki/Grid2

REMOTEWEBDRIVER

Potrzebny jeżeli chcemy używać Selenium Grid.

Kiedyś to było...

```
private static WebDriver driver;
DesiredCapabilities capability = DesiredCapabilities.chrome();
driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);
```

```
private static WebDriver driver;
ChromeOptions options = new ChromeOptions();
driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),
options);
```

Dokumentacja:
https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/remote/RemoteWebDriver.html
https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/chrome/ChromeOptions.html

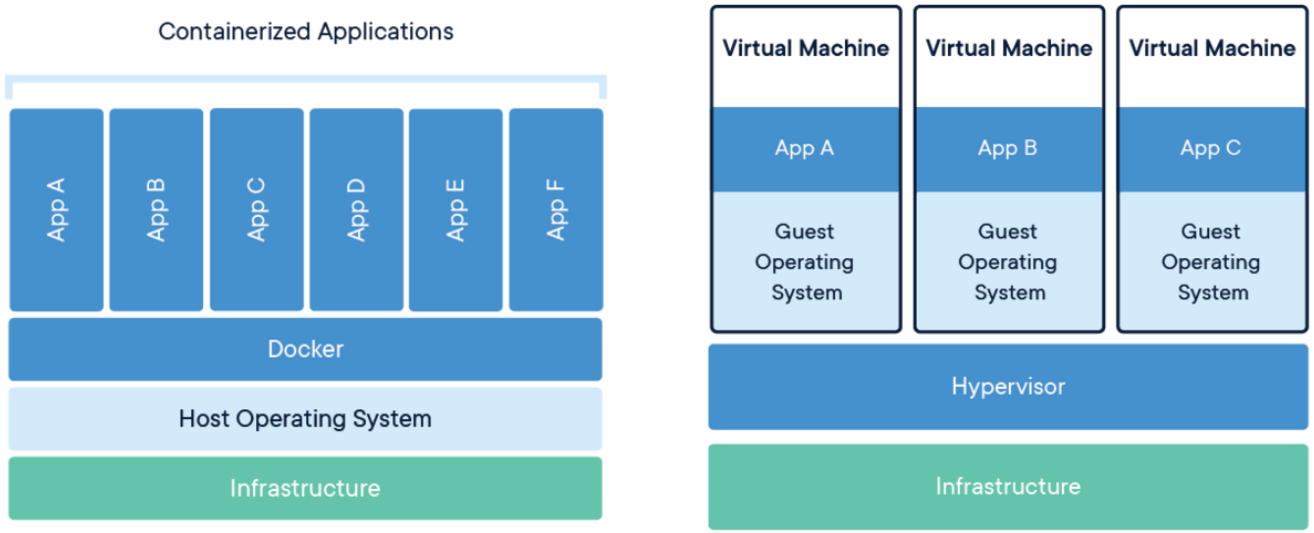
CO TO JEST DOCKER

- Docker to otwarte oprogramowanie służące jako platforma dla programistów i administratorów do tworzenia, wdrażania i uruchamiania aplikacji rozproszonych.

- Docker jest określany jako narzędzie, które pozwala umieścić program oraz jego zależności w lekkim, przenośnym, wirtualnym kontenerze, który można uruchomić na prawie każdym serwerze z systemem Linux.

CO TO JEST DOCKER

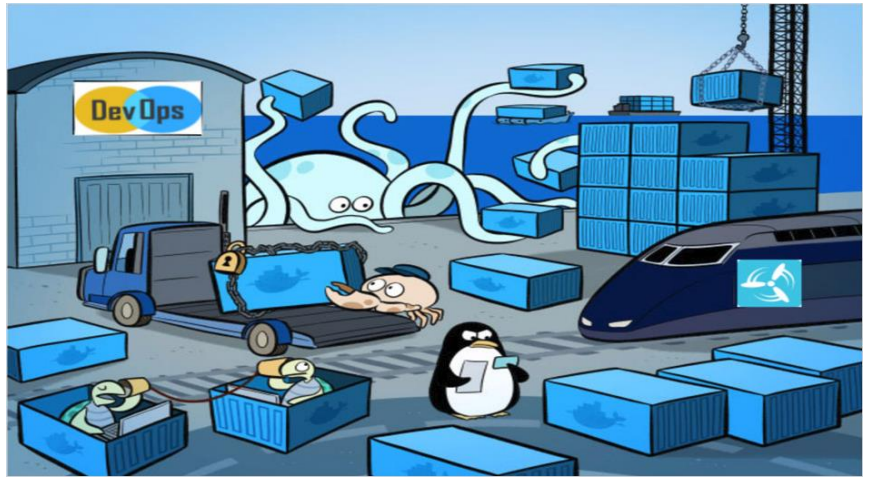
- **Kontener dockera to nie VM!!**
- Kontener działa podobnie, lecz wirtualizuje tylko system – nie całą maszynę
- VM jest pełną kopią systemu operacyjnego wraz ze wszystkimi binarkami
- Kontenery współdzielą jądro systemu, ale każdy używa wyizolowanch procesów
- Kontenery działają na bazie obrazów (image), które definiują jak jest on zbudowany



CO TO JEST DOCKER

- **Docker-file** – plik z przepisem jak zbudować obraz dockera
- **Image** – paczka zawierająca wszystkie niezbędne kompnenty (kod, biblioteki, zmienne środowiskowe, konfigurację
- **Kontener** – działający obraz dokera
- **Docker registry** – repozytorium z obrazami dockera
- **Docker hub** – oficjalne repozytorium
- **Docker swarm** – klaster zbudowany z wielu serwerów z dockerem
- **Docker-compose** – narzędzie do definiowania struktury złożonej z wielu kontenerów

```
FROM ubuntu:16.04
ENV DEBIAN_FRONTEND=noninteractive \
    DEBCONF_NONINTERACTIVE_SEEN=true
RUN apt-get -qqy update
COPY check-grid.sh \
       /opt/bin/
```



JAK UŻYĆ

```
ubuntu@ubuntu-GW:~$ sudo docker swarm init --advertise-addr 192.168.56.102
Swarm initialized: current node (vhn03rp241jq0y40bwnl4rl59) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-45g768in51442fh5bk7h5p9ucec8i3lja0ermoib05yxpqby0p-8d8iw26uaqwh4d6b5z2jusc0s 192.168.56.102:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

ubuntu@ubuntu-GW:~/SS$ sudo docker network create --driver overlay selenium_network
2oi2c8rl9dckwi3hgx4bt7yrr
ubuntu@ubuntu-GW:~/SS$ sudo docker stack deploy SS --compose-file ./docker-compose.yml
Creating service SS_FF
Creating service SS_CH
Creating service SS_hub
ubuntu@ubuntu-GW:~/SS$ sudo docker rm $(sudo docker ps -qa)
231406590a1c
ed5af9f1465d
4e370287802e
a1af03470af9
52b702dba29c
388db9a6715d
82225b13019e
4b03ef31dcd5
4c03693b690f
5a4d7bfbce74
551d688cb13c
```

DZIĘKUJEMY