

Wirtualizacja – Hands On infoShare Academy



HELLO

Michał Jerzy Wierzbicki

in case you've forgotten





Agenda

- 1. Tworzymy VM alpine ręcznie
- 2. Tworzymy VM alpine przez Vagranta
- 3. Szybki omówienie docker swarm (tylko na potrzeby laba)
- 4. Tworzymy klaster docker swarm









Tworzymy VM alpine ręcznie

Zamiast apt-get install jest apk add. rc-service "\${service}" start/stop

polecenie	znaczenie
apkupdate add \${pkgs}	<pre>apt-get update && apt-get install \${pkgs}</pre>
<pre>rc-service "\${service}" start/stop</pre>	Odpal/zatrzymaj serwis
<pre>rc-update add "\${service}" "\${runlevel}"</pre>	Ustaw serwis żeby włączał się na trigger z runlevel (np. default, boot)
adduser "\${username}"	Stwórz użytkownika
adduser "\${username}" "\${group}"	Dodaj użytkownika do grupy
sudo / doas	Czasowe podniesienie przywilejów do wheel





krótkie przypomnienie





Tworzymy VM alpine przez Vagranta

Plik musi nazywać się dokładnie "Vagrantfile"

Polecenia wydawać z katalogu w którym jest Vagrantfile

Vagrantfile to ruby, więc można korzystać ze zmiennych, pętli, etc.

polecenie	znaczenie
vagrant up	Stwórz VM
vagrant provision	Skonfiguruj OS wewnątrz VM
vagrant upprovision	vagrant up + vagrant provision
vagrant destroy	Zniszcz VM
vagrant reload	Jeśli istnieją VM, zniszcz i vagrant up
vagrant reloadprovision	Jeśli istnieją VM, zniszcz i vagrant upprovision
\$foo=Bar	(w ruby) globalna zmienna foo o wartości Bar
\$baz="#{\$foo}bara"	(w ruby) globalna zmienna baz o wartości Barbara
<pre>vagrant upprovision vagrant destroy vagrant reload vagrant reloadprovision \$foo=Bar</pre>	vagrant up + vagrant provision Zniszcz VM Jeśli istnieją VM, zniszcz i vagrant up Jeśli istnieją VM, zniszcz i vagrant upprovision (w ruby) globalna zmienna foo o wartości Bar









Szybkie omówienie docker swarm

Docker swarm służy do sterowania wieloma hostami z dockerem.

Tak jak docker compose, korzysta z compose.yml (ale mają różnice w składni).

K	oolecenie	znaczenie
_	<pre>locker swarm init \listen-addr "\${ip}:\${port}"advertise- addr "\${ip}:\${port}"</pre>	Stwórz nowego swarma, ten host jest managerem
C	locker swarm join-tokenquiet worker	(na managerze) Wydrukuj token potrzebny do przyłączenia się jako worker
	locker swarm join \ token "\${token}" "\${ip}:\${port}"	(na workerze) przyłącz się do istniejącego swarma
	<pre>locker stack deploy \compose-file "\${fpath}" "\${stack_name}"</pre>	Postaw na swarmie zestaw kontenerów
C	locker stack rm "\${stack_name}"	Zniszcz ze swarma zestaw kontenerów









Tworzymy klaster docker swarm

Najlepiej wszystko w 1 Vagrantfile

1+ manager VM (zacznijcie od 1, nie ma co szaleć od początku)

2+ worker VM (vagrant + Virtualbox nie działa za szybko)

polecenie	znaczenie
set -e	(w shellu) jeśli którekolwiek polecenie da wynik negatywny, zakończ działanie skrytpu natychmiast
set +e	(w shellu) wyłącz powyższe
<pre>docker stack deploy \compose-file "\${fpath}" "\${stack_name}"</pre>	Postaw na swarmie zestaw kontenerów
<pre>docker stack rm "\${stack_name}"</pre>	Zniszcz ze swarma zestaw kontenerów









PYTANA infoShare Academy

infoShareAcademy.com



THANK YOU FOR YOUR ATTENTION

infoShareAcademy.com