Politechnika Warszawska

WYDZIAŁ ELEKTRYCZNY

Instytut Elektrotechniki Teoretycznej
i Systemów Informacyjno-Pomiarowych
Zakład Elektrotechniki Teoretycznej
i Informatyki Stosowanej

Praca dyplomowa inżynierska

na kierunku Informatyka
w specjalności Inżynieria oprogramowania

Implementacja i testy wydajnosci środowiska Kubernetes na maszynach bezdyskowych

Krzysztof Nazarewksi
nr albumu 123456

promotor
mgr inż. Andrzej Toboła

WARSZAWA 2018

Implementacja i testy wydajnosci środowiska Kubernetes na maszynach bezdyskowych

Streszczenie

Celem tej pracy inżynierskiej jest przybliżenie czytelnikowi zagadnień związanych z uruchamianiem systemu Kubernetes na maszynach bezdyskowych.

Zacznę od wyjaśnienia pojęcia systemu bezdyskowego oraz sposobu jego funkcjonowania na przykładzie sieci uczelnianej wzorującego się na niej przygotowanie przeze mnie lokalnego środowiska.

Następnie opiszę problem izolacji i przydzielania zasobów systemowych na przykładzie wirtualnych maszyn, chroot, control group i kontenerów.

W głównej części dokumentu przedstawię pojęcie orkiestrami kontenerami, w jaki sposób odnosi się do wcześniej postawionych problemów. Opiszę alternatywy Kubernetes, jego architekturę oraz sposoby uruchamiania. Na koniec spróbuję uruchomić Kubernetes na maszynach bezdyskowych, problemy z tym związane oraz przedstawię wyniki.

Słowa kluczowe: praca dyplomowa, LaTeX, jakość

Implementing and testing Kubernetes running on diskless machines

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ac dolor scelerisque, malesuada ex vel, feugiat augue. Suspendisse dictum, elit efficitur vestibulum eleifend, mi neque accumsan velit, at ultricies ex lectus et urna. Pellentesque vel lorem turpis. Donec blandit arcu lacus, vitae dapibus tellus tempus et. Etiam orci libero, mollis in dapibus tempor, rutrum eget magna. Nullam congue libero non velit suscipit, vel cursus elit commodo. Praesent mollis augue quis lorem laoreet, condimentum scelerisque ex pharetra. Sed est ex, gravida a porta in, tristique ac nunc. Nunc at varius sem, sit amet consectetur velit.

Keywords: thesis, LaTeX, quality

WARSZAWA, 1 lutego 1234

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY

OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa inżynierska pt. Implementacja i testy wydajnosci środowiska Kubernetes na maszynach bezdyskowych:

- została napisana przeze mnie samodzielnie,
- nie narusza niczyich praw autorskich,
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej. Jestem świasdom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Krzysztof Nazarewksi.....

Spis treści

[1] `https://alpinelinux.org/`
[2] `https://clearlinux.org/`

Podziękowania

Dziękujemy bardzo serdecznie wszystkim, a w szczególności Rodzinom i Unii Europejskiej...

Zdolny Student i Pracowity Kolega

Rozdział 1

Test

The seminal work (Pizza i in. 2000)

Rozdział 2

Wstęp

2.1 System bezdyskowy

Rozdział 3

Systemy orkiestracji kontenerami

3.1 Fleet

Fleet[1] jest nakładką na systemd[2] realizująca rozproszony system inicjalizacji systemów.

3.2 Docker Swarm

Docker Swarm[3] jest rozwiązaniem orkiestracji kontenerami od twórców samego Docker'a. Proste w konfiguracji, nie oferuje tak dużych możliwości jak niżej wymienione.

3.3 Kubernetes

Kubernetes[4] jest jednym z najpopularniejszych narzędzi orkiestracji kontenerami. Stworzone przez Google i bazowane na wewnętrznym systemie Borg.

3.4 Mesos

Apache Mesos[5] zaawansowane narzędzie orkiestracji kontenerami.

[1] `https://coreos.com/fleet/docs/latest/launching-containers-fleet.html`
[2] `https://www.freedesktop.org/wiki/Software/systemd/`
[3] `https://docs.docker.com/engine/swarm/`
[4] `https://kubernetes.io/`
[5] `http://mesos.apache.org/`

3.5 Rancher

Rancher[6] jest platformą zarządzania kontenerami umożliwiającą między innymi zarządzanie klastrem Kubernetes. Od wersji 2.0 twórcy skupiają się na zarządzaniu Kubernetes porzucając inne silniki

[6] `https://rancher.com/`

Rozdział 4

Proces uruchamiania maszyn bezdyskowych

Na uruchamianie maszyn bezdyskowych w protokole PXE składają się 3 podstawowe elementy: 1. serwer DHCP, np. isc-dhcp-server lub dnsmasq 2. firmware wspierające PXE, np. iPXE 3. serwer plików (np. TFTP, HTTP, NFS)

Rozdział 5

Systemy operacyjne

5.1 cloud-init

cloud-init[1] jest standardem oraz implementacją konfiguracji wielu systemów działających w chmurze.

Standard polega na dostarczeniu pliku konfiguracyjnego w formacie YAML[2] w trakcie lub tuż po inicjalizacji systemu operacyjnego.

W niżej wymienionych dystrybucjach pojawiają się wzmianki o systemie, ale nie zawsze znaczą to samo, dlatego zawrę w nich uściślenia.

5.1.1 Implementacja

cloud-init zaimplementowany jest w Pythonie, co częściowo tłumaczy duży rozmiar obrazów przeznaczonych dla chmury. Po najmniejszych obrazach Python'a dla Docker'a[3] (python:alpine - 89MB i python2:alpine - 72 MB) wnioskuję, że nie istnieje mniejsza dystrybucja Python'a.

```
docker pull python:2-alpine > /dev/null
docker pull python:alpine > /dev/null
docker images | grep alpine
```

TODO: https://www.podcastinit.com/cloud-init-with-scott-moser-episode-126/

5.1.2 cloud-init CoreOS

Jest wychodzącą z użycia wewnętrzną implementacją czę

[1] `https://cloud-init.io/`
[2] `http://yaml.org/`
[3] `https://hub.docker.com/_/python/`

5.2 CoreOS

CoreOS[4] jest pierwszą dystrybucją linux'a dedykowaną zarządzaniu kontenerami. Zawiera dużo narzędzi dedykowanych klastrowaniu i obsłudze kontenerów, w związku z tym zajmuje 342 MB.

5.2.1 Konfiguracja

Konfiguracja przez Container Linux Config[5] transpilowany do Ignition[6]. Transpiler konwertuje ogólną konfigurację na przygotowaną pod konkretne chmury (AWS, GCE, Azure itp.). Dyskwalifikującą wadą tego typu konfiguracji jest brak wsparcia transpilatora dla systemów z rodziny BSD

Alternatywą Ignition jest dotychczasowy coreos-cloudinit[7], czyli częściowa implentacją cloud-init w języku Go. Niestety nie jest już aktywnie rozwijany i wychodzi z użytku.

5.3 RancherOS

RancherOS[8] jest systemem operacyjnym, w którym tradycyjny system inicjalizacji został zastąpiony trzema poziomami Docker'a[9]: - `bootstrap_docker` działający w initram'ie, czyli przygotowujący system, - `system-docker` zastępuje tradycyjnego init'a, zarządza wszystkimi programami systemowymi, - `docker` standardowy docker, interakcja z nim nie może uszkodzić działającego systemu,

Jego głównymi zaletami są mały rozmiar (45 MB) oraz prostota konfiguracji.

5.3.1 Konfiguracja

RancherOS jest konfigurowany przez cloud-init[10], czyli pochodną cloud-init'a[11] z CoreOS.

Znaczną przewagą nad oryginałem jest możliwość sekwencyjnego uruchamiania dowolnej ilości plików konfiguracyjnych.

[4] `https://coreos.com/`
[5] `https://coreos.com/os/docs/latest/provisioning.html`
[6] `https://coreos.com/ignition/docs/latest/`
[7] `https://github.com/coreos/coreos-cloudinit`
[8] `https://rancher.com/rancher-os/`
[9] `http://rancher.com/docs/os/latest/en/configuration/docker/`
[10] `http://rancher.com/docs/os/latest/en/configuration/`
[11] `https://github.com/rancher/os/commit/e2ed97648ad63455743ebc16080a82ee47f8bb0c`

Przydatne jest wyświetlenie kompletnej konfiguracji komendą `ros config export --full`[12].

5.4 Project Atomic

Project Atomic[13] jest grupą podobnie skonfigurowanych systemów operacyjnych dedykowaną środowiskom cloud i kontenerom.

Dystrybucje Project Atomic nazywają się Atomic Host, dostępne są następujące warianty: - Red Hat Atomic Host[14] - CentOS Atomic Host[15] - Fedora Atomic Host[16]

5.4.1 Konfiguracja

Atomic Host są konfigurowane systemem cloud-init[17],

5.5 Alpine Linux[18]

Minimalna dystrybucja Linux'a bazowana na musl-libc i busybox.

Niestety nie bootuje sie w trybie diskless ze wzgledu na buga, ktorego tworcy nie umieja naprawic.

5.5.1 Konfiguracja

`Alpine Backup`[19] - spakowane pliki wypakowywane w sekwencji boot'u

Alpine Configuration Framework[20]

5.6 ClearLinux[21]

- "bundle" zamiast pakietów systemowych aktualizowane z całym systemem

[12] `https://forums.rancher.com/t/good-cloud-config-reference/5238/3`
[13] `https://www.projectatomic.io/`
[14] `https://www.redhat.com/en/resources/enterprise-linux-atomic-host-datasheet`
[15] `https://wiki.centos.org/SpecialInterestGroup/Atomic/Download/`
[16] `https://getfedora.org/atomic/download/`
[17] `https://cloud-init.io/`
[18] `https://alpinelinux.org/`
[19] `https://wiki.alpinelinux.org/wiki/Alpine_local_backup`
[20] `http://wiki.alpinelinux.org/wiki/Alpine_Configuration_Framework_Design`
[21] `https://clearlinux.org/`

- skoncentrowany na wydajności na procesorach Intel
- niewygodny format dokumentacji (brak kompletnej mapy dokumentacji, duze zagniezdzenia)
- lokalizacja wszystkich modyfikacji w /var i /etc (prosty reset)

5.6.1 Linki

- https://www.infoworld.com/article/3159658/linux/6-key-points-about-intels-hot-new-linux-distro.html

Bibliografia

Pizza, Mariagrazia, Vincenzo Scarlato, Vega Masignani, Marzia Monica Giuliani, Beatrice Arico, Maurizio Comanducci, Gary T Jennings, i in. 2000. „Identification of vaccine candidates against serogroup B meningococcus by whole-genome sequencing”. *Science* 287 (5459). American Association for the Advancement of Science:1816–20.