

Příjemci podpory:

Vysoké učení technické v Brně
Fakulta informačních technologií

Poskytovatel:

Ministerstvo vnitra ČR

Integrovaná platforma pro zpracování digitálních dat z bezpečnostních incidentů (TARZAN)

Identifikační kód VI20172020062

Název předkládaného výsledku: Big Data Containerized Cluster

Typ výsledku dle UV č. 837/2017	Evidenční číslo (příjemce)	Rok vzniku
R software		2019
ISBN-ISSN	Webový odkaz na výsledek	Kde a kdy publikováno
	https://www.fit.vut.cz/research/product/622/	

Anotace k výsledku:

Tento software slouží pro sestavení big data výpočetního clusteru založeného na kontejnerech a patřičných management technologiích jako jsou Docker Swarm a Kubernetes. Umožňuje nasazení a správnou kooperaci jednotlivých služeb, které se používají pro zpracování velkých dat.

Řešitelský tým: Petr Matoušek (manažer a hlavní řešitel), Kamil Jeřábek (realizační tým)

Big Data Containerized Cluster

Big Data výpočetní cluster založený
na Docker Swarm a Kubernetes

Instalační příručka

Kamil Jeřábek



Dokument k projektu TARZAN
Fakulta informačních technologií, Vysoké učení technické v Brně

Naposledy změněno: 3. července 2020

Big Data cluster - Instalační příručka

Kamil Jeřábek

Vysoké učení technické v Brně, email: ijerabek@fit.vutbr.cz

Abstrakt V dnešní době hledáme stále nové způsoby jak zjednodušit nasazení aplikací a služeb jednotným způsobem nezávisle na hardware. V tomto ohledu se dostává do popředí kontejnerizace. S kontejnery se můžeme setkat taktéž v Cloudu, kde jsou masově nasazovány v podobě tzv. microservices. Nicméně pro správu více strojů se musíme poohlédnout po cluster management nástroji, který je schopný řídit a spravovat nasazování jednotlivých kontejnerů. Takovýmito technologiemi v současné době jsou Docker Swarm a Kubernetes.

1 Úvod

Big Data cluster je založen na technologii Docker v kombinaci s cluster management nástroji Docker Swarm a Kubernetes. Poskytuje konfigurační soubory pro nasazení jednotlivých služeb potřebných pro sestavení Big Data processing pipeline. Tyto služby zahrnují konfigurační soubory pro Apache Spark, Apache Ignite, Apache Cassandra, Apache Hadoop, Apache Kafka, Apache Zeppelin, Nginx.

Jednotlivé služby jsou sestaveny pomocí kontejnerových obrazů dostupných na oficiálním docker repository DockerHub¹. O správu clusteru se stará Kubernetes verze 1.15.3 a Docker verze 18.09.6. Pro jednotlivé služby byly použity kontejnery, ve kterých běží služby těchto verzí Spark v2.2.0 s Hadoop v2.8 a java v8, Hadoop Hadoop v2.7.4 java v8, Ignite v2.6.0, Cassandra v3.11, Zeppelin v0.8.1. V tomto složení jsou schopny všechny služby mezi sebou interagovat bez sebemenších problémů.

Pro cluster management je možné zvolit buď Docker Swarm nebo Kubernetes, nikoli obě zároveň. Docker Swarm poskytuje jednoduché nasazení, Kubernetes poskytuje možnost nasazení většího množství služeb a zároveň lepší udržitelnosti.

Dokument je členěn následovně: Sekce 2 popisuje hardwarové a softwarové podmínky nutné ke správnému chodu konfigurovaného systému. Sekce 3 pak zevrubně popisuje postup k úspěšnému nasazení všech dílčích částí systému. Sekce 4 obsahuje závěrečné poznámky, a dokument je pak uzavřen seznamem s tématem souvisejících odkazů a bibliografických zdrojů.

¹ <https://hub.docker.com/>

2 Prerekvizity

Konfigurační soubory clusteru jsou k dispozici v repozitáři `kubernetes-cluster-setup` [4].

Konfigurace předpokládá cluster o čtyřech propojených serverech. Níže uvedený seznam prerekvizit uvažuje linuxový serverový systém založený na CentOS 7.*, nicméně obecně lze použít libovolný operační systém, u něž jsme schopni zajistit následující:

1. Povolené směrování provozu v rámci virtuálních síťových adaptérů
`/sbin/sysctl -w net.ipv4.ip_forward=1`
2. Vypnutý swap, nutno odebrat záznam s položkou swap ze souboru `/etc/fstab`, následně použít příkaz `mount -a`.
3. Nainstalované pomocné provozy pro kontejnerizační prostředí jako LVM2
`yum install -y yum-utils device-mapper-persistent-data lvm2`
4. Nainstalovanou minimální verzi Docker 18.*
`yum-config-manager -add-repo
https://download.docker.com/linux/centos/docker-ce.repo
yum install -y docker-ce`

V případě použití Docker Swarm je nutné následující nastavení:

1. Na zvoleném stroji pro Docker Swarm master inicializovaný Swarm
`docker swarm init -advertise-addr <HOST_IP_ADDRESS>`
2. Na ostatních strojích pro Docker Swarm worker připojení workeru do clusteru
`docker swarm join -token <UNIQUE_TOKEN> <MASTER_IP>:<PORT>`

V případě použití Kubernetes je nutné doinstalovat následující:

1. Přidaný repozitář pro kubernetes
`cat <EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes
-el7-x86_64`
2. Nainstalovanou verzi Kubernetes v1.15.* a pomocné nástroje
`yum install -y kubelet kubeadm kubectl`
3. Spuštěnou službu pro kubernetes
`systemctl start kubelet && systemctl enable kubelet`
4. Kubernetes uvedený do původního nastavení
`kubeadm reset`
5. Na zvoleném stroji pro kubernetes master inicializovaný kubernetes
`kubeadm init -pod-network-cidr=10.244.0.0/16`
6. Na ostatních strojích pro kubernetes worker připojení workerů do clusteru
`kubeadm join <MASTER_IP_ADDRESS>:6443 -token <UNIQUE_TOKEN>
-discovery-token-ca-cert-hash sha256:<HASH>`
7. Označené jednotlivé stroje pomocí label
`kubectl label nodes <MASTER-NODE> worker=false pro master
kubectl label nodes <WORKER-NODE> worker=true pro workery`

3 Instalace

K provozu clusteru je potřeba několik služeb, které jsou schopny následně kooperovat. Tyto služby mohou běžet zároveň v různých kombinacích. Přičemž je můžeme rozdělit do dvou částí, a to uložště (Hadoop File System, Cassandra Database, Ignite In Memory Storage), a zpracovávací část (Kafka, Spark, Ignite). V poslední řadě zde pak máme Apache Zeppelin, který poskytuje interaktivní webové rozhraní ve formě notebooků pro ovládání jednotlivých služeb.

3.1 Docker Swarm

V případě použití Docker Swarm nastavení je nutné použít konfigurační soubory typu `docker-compose`. Taktéž není možné nasadit všechny služby a jsme nuceni se omezit pouze na Cassandra, Hadoop a Spark. Tyto jsou schopny spolu interagovat za použití Docker Swarm cluster managementu.

Konfigurační soubory pro tyto služby se nacházejí v adresáři `swarm`. Podrobný postup pro nasazení lze nalézt v uživatelském manuálu [3].

3.2 Kubernetes

Jednotlivé služby se nacházejí v samostatných adresářích. Každý adresář obsahuje skript pro nasazení dané služby `create.sh` a taktéž skript pro zastavení či odstranění služby `delete.sh`. Podrobný postup pro nasazení lze nalézt v uživatelském manuálu [3].

4 Závěr

Instalační příručka podrobně dokumentuje proces nasazení a instalace Big Data Výpočetního Clusteru. Tato konfigurace byla vyvinuta v rámci projektu TAR-ZAN [1].

Tato instalační příručka navazuje na naše další publikace:

- Technickou dokumentaci [2];
- uživatelský a programátorský manuál [3];

Odkazy

- [1] *Integrovaná platforma pro zpracování digitálních dat z bezpečnostních incidentů*. <https://www.fit.vut.cz/research/project/1063/>. navštíveno: 2019-09-10.
- [2] Kamil Jeřábek. *Big Data Vypocetni Cluster: Technická dokumentace*. Tech. zpr. 2019.
- [3] Kamil Jeřábek. *Big Data Vypocetni Cluster: Uzivatelcky a programatorsky manual*. Tech. zpr. 2019.
- [4] *Kubernetes Cluster Setup*. <https://github.com/kjerabek/kubernetes-cluster-setup>. navštíveno: 2019-12-10.

Big Data Containerized Cluster

Big Data výpočetní cluster založený
na Docker Swarm a Kubernetes

Uživatelský a programátorský manuál

Kamil Jeřábek



Dokument k projektu TARZAN
Fakulta informačních technologií, Vysoké učení technické v Brně

Naposledy změněno: 3. července 2020

Big Data cluster - Uživatelská příručka

Kamil Jeřábek

Vysoké učení technické v Brně, email: ijerabek@fit.vutbr.cz

Abstrakt V dnešní době hledáme stále nové způsoby jak zjednodušit nasazení aplikací a služeb jednotným způsobem nezávisle na hardware. V tomto ohledu se dostává do popředí kontejnerizace. S kontejnery se můžeme setkat taktéž v Cloudu, kde jsou masově nasazovány v podobě tzv. microservices. Nicméně pro správu více strojů se musíme poohlédnout po cluster management nástroji, který je schopný řídit a spravovat nasazování jednotlivých kontejnerů. Takovými technologiemi v současné době jsou Docker Swarm a Kubernetes.

1 Úvod

Big Data cluster je založen na technologii Docker v kombinaci s cluster management nástroji Docker Swarm a Kubernetes. Poskytuje konfigurační soubory pro nasazení jednotlivých služeb potřebných pro sestavení Big Data processing pipeline. Tyto služby zahrnují konfigurační soubory pro Apache Spark, Apache Ignite, Apache Cassandra, Apache Hadoop, Apache Kafka, Apache Zeppelin, Nginx.

Jednotlivé služby jsou sestaveny pomocí kontejnerových obrazů dostupných na oficiálním docker repository DockerHub¹. O správu clusteru se stará Kubernetes verze 1.15.3 a Docker verze 18.09.6. Pro jednotlivé služby byly použity kontejnery, ve kterých běží služby těchto verzí Spark v2.2.0 s Hadoop v2.8 a Java v8, Hadoop Hadoop v2.7.4 Java v8, Ignite v2.6.0, Cassandra v3.11, Zeppelin v0.8.1. V tomto složení jsou schopny všechny služby mezi sebou interagovat bez sebemenších problémů.

Jako hlavní výpočetní framework je myšlen Apache Spark, nicméně je možné použít taktéž Apache Ignite. Apache Ignite je možno použít taktéž jako úložiště ve formě klíč-hodnota, či cache vrstva. Apache Cassandra je zde myšlena jako primární úložiště pro zpracovaná data. Apache Hadoop zde slouží jako distribuovaný souborový systém pro ukládání zdrojových dat. V neposlední řadě je zde Apache Kafka, která taktéž může sloužit jako vstupní prvek do systému. Apache Zeppelin pak slouží jako interaktivní rozhraní pro přístup k jednotlivým službám.

Pro cluster management je možné zvolit buď Docker Swarm nebo Kubernetes, nikoli obě zároveň. Docker Swarm poskytuje jednoduché nasazení, Kubernetes poskytuje možnost nasazení většího množství služeb a zároveň lepší udržitelnosti.

¹ <https://hub.docker.com/>

Dokument je členěn následovně: Sekce 2 popisuje zprovoznění a práci s nasazením služeb pro Docker Swarm. Sekce 3 pak popisuje zprovoznění a práci s nasazením služeb pro Kubernetes. Sekce 4 obsahuje závěrečné poznámky, a dokument je pak uzavřen seznamem s tématem souvisejících odkazů a bibliografických zdrojů.

2 Docker Swarm

První možností je využití Docker Swarm pro konfiguraci clusteru. Docker Swarm ovšem poskytuje omezenější možnosti co se týče nasazení služeb. Například Apache Ignite lze nasadit pomocí tohoto manageru velmi obtížně a nefunguje kompletně, chybí možnost nalezení a spárování jednotlivých uzlů mezi sebou. Nicméně pro nasazení Hadoop, Spark a Cassandra je možné jej využít bez problému.

K nasazení služeb slouží `docker-compose` soubory, tyto soubory nalezneme v adresáři `swarm`. V těchto souborech můžeme nalézt popis několika služeb najeďnou, následně se nasazují současně jako stack.

U každé služby je nicméně potřeba nakonfigurovat počet replik, kdy je počítáno s jedním kontejnerem na každém stroji. Je nutné mít označené jednotlivé uzly jako master a worker, toto je popsáno v instalační příručce. Následně je nutné vytvořit sdílenou síť pomocí příkazu:

```
docker network create -d overlay --attachable spark-net.
```

Pro nasazení služeb Spark/hadoop/cassandra je potřeba nastavit pro worker kontejnery počet replik. Je v souboru `docker-compose-{spark|hadoop|cassandra}.yml` nutno upravit následující řádek:

```
replicas: 3.
```

Pro služby, které pracují s diskem (hadoop, cassandra), je nutné nastavit také `volume`. Tedy upravit řádek obsahující komentář `#change this to your path`.

Následně je možno nasadit službu pomocí příkazu:

```
docker stack deploy -c docker-compose-{spark|hadoop|cassandra}.yml {spark|hadoop|cassandra}.
```

Pro destrukci stacku je možné použít příkaz:

```
docker stack rm {spark|hadoop|cassandra}.
```

3 Kubernetes

Druhou možností pro cluster management založen na kontejnerech je Kubernetes. Tento manager nám umožňuje nasadit větší množství služeb, včetně Apache

Ignite, který má pro Kubernetes implementován mechanismus pro objevování uzlů.

Pro nasazení každé ze služeb je v adresáři s konfiguračními soubory přiložen skript `create.sh`. Pro destrukci skript `delete.sh`, který zároveň resetuje `physical volumes`.

Jelikož každá z těchto služeb pracuje s úložištěm je nutné na každém stroji zvlášť vytvořit adresáře pro jednotlivé služby (záleží nám na datové lokalitě) a následně vytvořit tzv `physical volumes` pro kubernetes. K tomuto účelu je zde adresář `manual-storage-configs`, který obsahuje adresáře s konfiguračními soubory pro všechny `Physical Volumes` pro jednotlivé služby. Každý adresář obsahuje soubory `persistentvolume{,2,3,4}.yaml`. V těchto souborech je nutné změnit řádky obsahující:

```
path: /specify/your/own/path/ #specify your own path
```

tak aby cesta odpovídala připravenému adresáři pro patřičné volume dané služby na daném stroji. Následně pak řádek obsahující:

```
- host1 #change this to hostname
```

tak aby odpovídal hostname daného stroje. Následně je zde skript `create.sh` pro vytvoření všech volumes a taktéž skript `delete.sh` pro jejich destrukci.

3.1 Apache Spark

S polu se službou Apache Spark je možné nasadit samostatný kontejner `spark-submit`, pomocí kterého je možné spouštět spark shell a nasazovat spark joby do clusteru. Konfigurační soubory pro tento kontejner a jeho nasazení jako služby je možné v adresáři `spark/spark-submit/`.

3.2 Apache Ignite

Pro nasazení Apache Ignite je nejprve potřeba spustit službu `nginx` a vybrané konfigurační soubory, které se nacházejí v adresáři `ignite/ignite-configs/` zkopírovat do adresáře, který je prolinkovaný s volume pro `nginx`. Následně je nutné před nasazením služby upravit soubor `ignite/ignite-configs/ignite-deployment.yml`, konkrétně hodnotu pro proměnnou prostředí `CONFIG_URI`, tak aby ukazovala na správný název zvoleného konfiguračního souboru pro službu Apache Ignite.

3.3 Apache Cassandra

V případě nasazení služby pro `cassandra`, je možné nastavit některé parametry, které se nacházejí v konfiguračním souboru `cassandra/cassandra-statefulset.yml`, v části `env`. Podstatné parametry pro nastavení výkonnosti jsou `MAX_HEAP_SIZE` a `HEAP_NEWSIZE`.

3.4 Reverzní proxy

Pro zpřístupnění webových rozhraní mimo cluster je nutné použití reverzní proxy. V tomto případě je použita reverzní proxy Traefik². Konfigurační soubory se nacházejí v adresáři `traefik/`.

V konfiguračním souboru `traefik-spark.yaml` je nutné upravit řádky, které obsahují komentář `#specify yours`. Na těchto řádcích je potřeba upravit doménové jména podle toho, jak jsou přiřazena jednotlivým službám a strojům.

4 Závěr

Uživatelská příručka podrobně dokumentuje uživatelské nastavení a provoz Big Data Výpočetního Clusteru. Tato konfigurace byla vyvinuta v rámci projektu TARZAN [1].

Tato uživatelská příručka navazuje na naše další publikace:

- Technickou dokumentaci [3];
- instalační příručku [2];

Odkazy

- [1] *Integrovaná platforma pro zpracování digitalních dat z bezpečnostních incidentů*. <https://www.fit.vut.cz/research/project/1063/>. navštíveno: 2019-09-10.
- [2] Kamil Jeřábek. *Big Data Vypocetni Cluster: Instalacni prirucka*. Tech. zpr. 2019.
- [3] Kamil Jeřábek. *Big Data Vypocetni Cluster: Technická dokumentace*. Tech. zpr. 2019.

² <https://docs.traefik.io/>