Inštalačná príručka a pokyny na inštaláciu (úvodnú/opakovanú) - Národní katalog otevřených dat

Čtenář této příručky by měl být obeznámen s:

- Kubernetes
- Oracle Cloud Infrastructure (OCI)

1. ROZSAH PLATNOSTI A ÚČEL

Tato inštalačná príručka slouží k popisu Národního katalogu otevřených dat (NKOD), části projektu OD2.0.

2. DEFINÍCIA POJMOV A SKRATIEK

NKOD
Národní katalog otevřených dat

OCI

Oracle Cloud Infrastructure

OCID
Oracle Cloud Identifier

LP-ETL
LinkedPipes ETL

SPARQL Endpoint
Webová služba běžící nad RDF databází, přijímající dotazy v jazyce SPARQL, a vracející výsledky vyhodnocení dotazu nad daty v databázi.

OCPU
The Oracle CPU

3. POŽIADAVKY NA ZÁKLADNÉ HW / Cloud a SW PROSTREDIE

Tato instalační příručka počítá s existencí Kubernetes clusteru verze 1.24.1 v prostředí OCI.

Z hlediska hardwarových požadavků jsou směrodatné dvě výpočetně intenzivní komponenty:

- RDF úložiště
- Metadatový procesor - LinkedPipes ETL. Provoz každé z komponent požaduje 16GB operační paměti a jedno OCPU, které odpovídá zhruba dvěma CPU. Výše uvedené komponenty nelze v rámci clusteru replikovat. Z toho důvodu je postačují, aby měl cluster tři výpočetní instance (worker nodes).

Při instalaci se počítá s využitím Let's Encrypt pro získání HTTPS certifikátu. Z tohoto důvodu je nutné mít přístup k nastavení DNS pro cílovou doménu nasazení.

4. POSTUP INŠTALÁCIE (ÚVODNEJ / OPAKOVANEJ)

Instalace se prování na základě obsahu repozitáře NKOD-SW. Tento repozitář obsahuje větve main a develop. Větev main by měla být nasazena na produkční prostředí, větev develop pak na prostředí testovací.

4.1 Popis inštalácie SERVEROVEJ ČASTI

Serverová část řešení je nasazena do Kubernetes clusteru na OCI. Uvedené příkazy je potřeba pouštět OCI Cloud Shell připojeném do sítě Kubernetes. Připojení ke správné síti můžeme otestovat funkčností příkazu `kubectl` . Dále je třeba mít zvolené správný Compartment, do kterého budeme nasazení provádět.

4.1.1 Zjištění informací o clusteru

Pro potřeby instalace je třeba zjistit základní informace o OCI.

- *{Availability Domain}*
- *{Compartment}*
- *{Virtual Cloud Network}*
- *{Subnet}*
- OCI veřejného *{Subnet}* pro Load Balancer
- jméno domény
- email na kontaktní osobu pro HTTPS certifikát
- instance Kubernetes do které budeme nasazovat

Tyto hodnoty by nám měl sdělit správce OCI do kterého provádíme instalaci. Pro interakci s Kubernetes je téměř vždy třeba znát OCID daného zdroje, nikoliv jeho lidsky čitelné pojmenování.

Pro ilustraci, aktuální testovací prostředí má tyto parametry:

- *{Availability Domain}* `TREJ:EU-FRANKFURT-1-AD-1` , `TREJ:EU-FRANKFURT-1-AD-2` , `TREJ:EU-FRANKFURT-1-AD-3`
- *{Compartment}* `cloudmirri (root)/cmp-mirri/subcmp-opendata`
- *{Virtual Cloud Network}* `ocid1.vcn.oc1.eu-frankfurt-1.amaaaaaavxy2x2qaomns3mpzaph3w6ieferg25rbgmcatkdcbnmasncz4nea`
- *{Subnet}* `ocid1.subnet.oc1.eu-frankfurt-1.aaaaaaaais3n3chzuupml4yy2g2i7z2glxp7xt3u5fqtlqbvh74btnv66fvq`

Pro ilustraci, aktuální produkční prostředí má tyto parametry:

- *{Availability Domain}* `TREJ:EU-FRANKFURT-1-AD-1` , `TREJ:EU-FRANKFURT-1-AD-2` , `TREJ:EU-FRANKFURT-1-AD-3`
- *{Compartment}* `cloudmirri (root)/cmp-mirri/subcmp-opendata`
- *{Virtual Cloud Network}* `ocid1.vcn.oc1.eu-frankfurt-1.amaaaaaavxy2x2qa3fng7ymqvhwdqpwkm55cfheehdfeymzjrwwxklbrfpqq`

- *{Subnet}* `ocid1.subnet.oc1.eu-frankfurt-1.aaaaaaaa2ue3etifv7j3uxkx325efgmndftbqlmr655kuoy7lk2fi4bkxzmq`

4.1.2 OCI File Systems

Data NKOD jsou uložena na dvou typech úložišť *Block Storage* a *File System*. První typ je vhodný pro datově intenzivní operace, například databáze. Druhý pak spíše pro sdílení souborů.

Oba druhy úložišť vytvořit z prostředí Kubernetes. Nicméně z důvodu snazší instalace a lepší použitelnosti jsou úložiště *File System* vytvořena v prostředí OCI a následně použita v Kubernetes. Samotný *File System* poskytuje pouze datové úložiště, pro jehož použití musíme definovat *Export Path*, která je dostupná přes *Mount Target*. Neb je *Mount Target* schopen obsloužit více *File System*, využijeme pro jednoduchost pouze jedné jeho instance.

Do správy *File System* je možné se navigovat z hlavního menu přes *Storage* > *File System*. Následující seznam obsahuje údaje nezbytné pro vytvoření *File System*, budeme vytvářet `File System for NFS`:

- NODC-Website
  - Name: NODC-Website
  - Availability Domain: *{Availability Domain}*
  - Compartment: *{Compartment}*
  - Export path: /website
  - Mount name: MountTarget-NODC
  - VCN: *{Virtual Cloud Network}*
  - Subnet: *{Subnet}*
- NODC-Registration
  - Name: NODC-Registration
  - Availability Domain: *{Availability Domain}*
  - Compartment: *{Compartment}*
  - Export path: /registration
  - Vyberme existující *Mount Target* vytvořený pro první *File System*
- NODC-LinkedPipes-Storage
  - Name: NODC-LinkedPipes-Storage
  - Availability Domain: *{Availability Domain}*
  - Compartment: *{Compartment}*

    - Export path: /linkedpipes-storage
    - Vyberme existující *Mount Target* vytvořený pro první *File System*
- NODC-Certificate
    - Name: NODC-Certificate
    - Availability Domain: *{Availability Domain}*
    - Compartment: *{Compartment}*
    - Export path: /certificate
    - Vyberme existující *Mount Target* vytvořený pro první *File System*

Pro další kroky v instalaci bude třeba si zjistit následující informace o vytvořených objektech:

- OCID pro NODC-Storage
- OCID pro NODC-Registration
- OCID pro NODC-Website
- OCID pro NODC-Certificate
- IP adresu vytvořeného *Mount Target*

Všechny tyto hodnoty je možné zjistit skrze webové rozhraní v detaily jednotlivých objektů.

4.1.3 Nastavení Cloud Shell

V dalších krocích budeme využívat OCI Cloud Shell.

Za účelem práce s Kubernetes je třeba být připojený do odpovídající sítě a nastavit připojení ke Kubernetes. K nastavení instance Kubernetes je třeba provést následující kroky:

- Navigovat se na *Developers Services* > *Kubernetes Clusters (OKE)*.
- Zde si najdeme cílovou instanci Kubernetes a otevřeme její detail.
- Po stisku tlačítka *Access Cluster* se zobrazí instrukce pro konfiguraci OCI Cloud Shell.

4.1.4 Projektový repozitář

Pro potřeby instalace vyžijeme definice z repozitáře NKOD-SW. Nejsnazším řešením je repozitář naklonovat do domovského adresáře v OCI Cloud Shell. V případě nasazení do testovacího prostředí je třeba změnit branch ve druhém řádku z `main` na `develop`.

```
git clone https://github.com/datova-kancelaria/nkod-software.git
```

```
cd nkod-software
git checkout main
```

V dalších krocích předpokládáme, že se uživatel nachází v kořeni naklonovaného repozitáře, tedy adresáři `./nkod-software/`.

4.1.5 Kubernetes jmenné prostory

Všechny Kubernetes objekty NKOD jsou umístěny ve jmenném prostoru. Ten je možné vytvořit následujícím příkazem:

```
kubectl apply -f ./k8s/namespace.yaml
```

4.1.6 Propojení Kubernetes a OCI

V tomto kroku budeme vytváře Kubernetes objekty, které se napojují na objekty v OCI. Do této kategorie patří zejména datová úložiště vytvořená v 4.1.2 OCI File Systems ale i další OCI specifické objekty.

Vytvoření potřebných tříd pro *Block Storage* je možné pomocí následujících příkazů:

```
kubectl apply -f k8s/oci/oci-block-storage-balanced.yaml
kubectl apply -f k8s/oci/oci-block-storage-high.yaml
```

Poznámka: V základní konfiguraci se využívá pouze balanced *Block Storage* a tvorba druhého typu *Block Storage* tak vlastně není nutná. Tento typ úložiště se používá v `./k8s/graphdb/graphdb-pvc.yaml` a `./k8s/linkedpipes/executor-pvc.yaml`. Zde je možné přejít na výkonnější třídu úložiště v případě nedostatečného výkonu.

Dále je třeba napojit Persistent Volume Claims z Kubernetes na *File System* z OCI. Za tímto účelem bude třeba upravit soubory Persistent Volume Claims definující soubory:

- `./k8s/oci/linkedpipes-storage-pv.yaml`
- `./k8s/oci/registration-pv.yaml`
- `./k8s/oci/website-pv.yaml`
- `./k8s/oci/certificate-pv.yaml`

V každém souboru je třeba provést substituci hodnot získaných v předchozích krocích. Jedná se vždy o hodnotu pro `spec.csi.volumeHandle`. Úprava souborů je až na použité hodnoty totožná, z tohoto důvodu popíšeme úpravu pouze pro `./k8s/oci/website-pv.yaml`. Ostatní soubory je třeba upravit podobným způsobem.

V tomto souboru je nutné nahradit `# {NODC-Website-OCID}:{MountTargetIP}:/website`, včetně začátku komentáře `#` za konkrétní hodnoty. Mějme například hodnoty:

- `NODC-Website-OCID = ocid1.filesystem.oc1.prague.sd_223`
- `MountTargetIP = 10.0.1.123`

Řádek po úpravě bude vypadat následovně:

```
volumeHandle: ocid1.filesystem.oc1.prague.sd_223:10.0.1.123:/website
```

Při substituci je třeba nezapomenout na zachování odsazení řádku, které určuje strukturu YAML dokumentu.

Jakmile jsou soubory upraveny můžeme na jejich základě vytvořit definice:

```
kubectl apply -f ./k8s/oci/linkedpipes-storage-pv.yaml
kubectl apply -f ./k8s/oci/registration-pv.yaml
kubectl apply -f ./k8s/oci/website-pv.yaml
kubectl apply -f ./k8s/oci/certificate-pv.yaml
```

Dalším specifickým zdrojem je veřejný Load Balancer. Hned na úvod je třeba upozornit na skutečnost, že smazáním Load Balanceru po jeho vytvoření ztratíme veřejnou IP adresu. Z výše uvedeného důvodu doporučujeme po vytvoření Load Balanceru nemazat.

Podobně jako u souborových systémů je třeba i zde upravit definici zdroje. V tomto případě je třeba nahradit hodnotu `# {Subnet}` v souboru `./k8s/oci/website-load-balancer.yaml`. Dle sekce 4.1.1 Zjištění informací o clusteru jsem získali lidsky čitelné jméno subnetu pro Load Balancer. Zde je nicméně třeba vložit OCID, to je možné zjistit ze webového rozhraní *Networking*, *Virtual Cloud Networks* sekce subnets.

Jakmile budou soubor upraven, můžeme vytvořit definici:

```
kubectl apply -f ./k8s/oci/website-load-balancer.yaml
```

4.1.7 Nastavení DNS

Na konci předchozího kroku jsem vytvořili veřejný Load Balancer. V tomto kroku je zapotřebí nastavit odpovídající DNS záznam. Veřejnou IP adresu Load Balanceru je možné zjistit pomocí příkazu:

```
kubectl get service nodc-public --namespace-nodc
```

4.1.8 Konfigurace a Secret

Konfigurační soubory jsou umístěné v adresáři `./configuration`. Ve zbytku této sekce popíšeme význam jednotlivých konfiguračních souborů a jejich naplnění. Nakonec uvedeme příkazy pro vytvoření konfigurace v Kubernetes z uvedených konfiguračních souborů.

dockerconfig.json tento soubor slouží pro nastavení přístupu k OCI Container Registry. Tuto konfiguraci potřebujeme, neb Kubernetes nemá v základu přístup do OCI Container Registry. Popis vytvoření soubor je možné najít v návodu Pull an Image from a Private Registry. Přihlášení do OCI Container registry je popsáno v Logging in to Oracle Cloud Infrastructure Registry, případně Push an Image to Oracle Cloud Infrastructure Registry.

Kroky k vytvoření souboru jsou následující:

- Ze sekce *Identity* > *My profile* si zobrazíme záložku *Auth Tokens*.
- Zde vytvoříme nový token, který si poznamenáme.-
- Následně provedeme přihlášení do Docker repository. To je možné s následujícím příkazem po dosazení hodnot:

```
docker login fra.ocir.io {object-storage-namespace}/{identity-domain}/{user-name}
```

Kde `{object-storage-namespace}` je možné získat pomocí příkazu `oci os ns get`. Hodnota `{identity-domain}` je součástí uživatelského jména.

- Jako heslo následně zadáme získaný token.
- Tímto dojde k přihlášení k Docker repository a zápisu přihlašovacích údajů do `~/.docker/config.json`.

nginx.conf konfigurační soubor pro NginX, je možné nechat ve výchozí podobě. V tomto souboru jsou uložené cesty k certifikátům.

nodc-configuration.properties konfigurace komponent v rámci NKOD. Význam jednotlivých nastavení je popsán v konfiguračním souboru.

nodc-secret.properties tento soubor obsahuje konfiguraci hesel. Význam jednotlivých nastavení je popsán v konfiguračním souboru.

Jakmile máme všechny konfigurační soubory připravené můžeme vytvořit konfigurace a Secret:

```
kubectl create configmap nodc-configuration --namespace=nodc --from-env-file=./co
kubectl create secret generic nodc-secret --namespace=nodc --from-env-file=./conf
```

Pro vytvoření konfigurace pro Webovou komponentu využijeme s výchozí konfigurací je možné použít příkaz:

```
kubectl create configmap nodc-nginx --namespace=nodc --from-file=website.nginx=./
```

Výchozí konfigurace poskytuje přístup pouze k veřejným komponentám využívá HTTPS. Alternativně je možné využít konfigurační soubory `nginx-no-https.conf` či `nginx-no-vpn.conf` . První slouží k zpřístupnění všech komponent bez využití HTTPS. Teto soubor je tedy možné využít pokud není k dispozici vhodná doména, nebo není třeba využití HTTPS. Druhý konfigurační soubor pak zpřístupní všechny komponenty skrze HTTPS. Ani jedna z těchto konfigurací by neměla být nasazena na produkci, neb poskytuje přístup k interním komponentám NKOD.

V následujícím příkazu je nutné aby cesta vedla k souboru `config.json` pro Docker. V cestě není možný využít zkratky pro domovský adresář. Soubor se nachází v umístění `~/.docker/config.json` . Neb tento soubor obsahuje přihlašovací údaje je třeba sní dle toho nakládat.

```
kubectl create secret generic oci-registry-secret --namespace=nodc --type=kuberne
```

Ačkoliv je konfigurace propagována automaticky stejně jako Secret, většina komponent čte konfiguraci pouze při vytvoření. Z tohoto důvodu je třeba po změně konfigurace smazat Pody, nebo restartovat Deployment. Za tímto účelem je možné využít následující příkaz:

```
kubectl rollout restart deploy {deployment-name} --namespace=nodc
```

Kde `{deployment-name}` je třeba nahradit za jméno Deploymentu, jehož Pody se mají smazat a znovu vytvořit.

4.1.9 Sestavení a publikace Docker image pro RDF úložiště

Tento krok není možné provádět v prostředí OCI, dojde k selhání Docker build. Pro účely stažení souboru je naopak lepší využít lokálního zařízení. V takovém případě je třeba opět vytvořit kopii repozitáře NKOD-SW.

Pro potřeby RDF úložiště je zvolena databáze Ontotext GraphDB Free. Tato verze bohužel nemá dostupný použitelný Docker image. Navíc verze ani není veřejně dostupná pro stažení a proto není možné Docker image automaticky vytvořit a publikovat na GitHubu. Z tohoto důvodu je třeba GraphDB image sestavit na jiném stroji a publikovat ho do OCI Container Registry. Za účelem publikace je třeba mít zřízen přístup. Pro získání přístupu lze využít obdobný postup jako pro vytvoření *dockerconfig.json* souboru v sekci 4.1.6 Konfigurace a Secret.

Chybějící soubor je možné stáhnout přímo ze stránek Ontotext GraphDB Free po registraci. Je třeba stáhnout `platform independent distribution package` verzi, jméno se musí shodovat s definici v `./components/graphdb/Dockerfile`. Stažený soubor je nutné umístit do adresáře `./components/graphdb`. V současné verzi s verzí 10.0.0 a tedy názvem soubor `graphdb-free-10.0.0-dist.zip`. V případě změny je nutné upravit `GRAPHDB_URL` v souboru `./components/graphdb/Dockerfile`.

Po přidání chybějícího souboru je pak možné z adresáři `./components/graphdb` provést sestavení a publikaci pomocí níže uvedených příkazů. Hodnotu `{object-storage-namespace}` je možné získat pomocí příkazu `oci os ns get`. Hodnota `{identity-domain}` je součástí uživatelského jména.

```
docker login -u '{object-storage-namespace}/{identity-domain}/{user-name}' fra.oc
docker build -t fra.ocir.io/{object-storage-namespace}/graphdb:10.0.0 .
docker push fra.ocir.io/{object-storage-namespace}/graphdb:10.0.0
```

4.1.9 Nasazení RDF úložiště GraphDB

Nasazení se provede pomocí příkazu:

```
kubectl apply -f ./k8s/graphdb/
```

Po nasazení je vhodné zkontrolovat stav pomocí příkazu:

```
kubectl get pod --namespace=nodc
```

4.1.10 Nasazení Metadatového procesoru - LinkedPipes ETL

Nasazení se provede pomocí příkazu:

```
kubectl apply -f ./k8s/website/website-pvc.yaml
kubectl apply -f ./k8s/linkedpipes/
```

Po nasazení je vhodné zkontrolovat stav pomocí příkazu:

```
kubectl get pod --namespace=nodc
```

4.1.11 Získání HTTPS certifikátu

Aby mohla probíhat HTTPS terminace uvnitř prostředí NKOD, je zapotřebí získat HTTPS certifikát. V rámci řešení NKOD je certifikát ulože v `certificate-pvc`. Ten je nutné vytvořit pomocí příkazu:

```
kubectl apply -f ./k8s/certificate-manager/certificate-pvc.yaml
```

Vydavatelem certifikátu byl zvolen Let's Encrypt, který omezuje platnost certifikátu na 3 měsíce. Let's Encrypt využije HTTP ověření přístupem na URL domény, pro které je certifikát požadován.

Za tímto účelem je využití Kubernetes Job, který využívá Docker container z `./components/certificate-manager`. Vytvořený job je rozpoznán jako cíl pro Load Balancer a je na něj po čas běhu vedena komunikace z portu 80 pro ověření. Z tohoto důvodu nesmí nesmí v době běhu být spuštěna komponenta Webová stránka.

Job je možné vytvořit následujícím příkazem.

```
kubectl apply -f ./k8s/certificate-manager/create-certificate.yaml
```

Po nasazení je vhodné zkontrolovat stav Jobu pomocí příkazu:

```
kubectl get pod,job --namespace=nodc
```

Volitelně je pak možné smazat záznam o Jobu a jeho definici z Kubernetes pomocí příkazu:

```
kubectl delete job nodc-create-job --namespace=nodc
```

4.1.12 Nasazení Webové stránky

Jakmile máme k dispozici certifikát a běží všechny ostatní služby je možné nasadit Webovou stránku. Důvodem pro toto omezení je vlastnost NginX, který při startu ověřuje dostupnost služeb pro které funguje jako proxy. Nasazení se provede následujícím příkazem:

```
kubectl apply -f ./k8s/website/website.yaml
```

Po nasazení je vhodné zkontrolovat stav Jobu pomocí příkazu:

```
kubectl get pod --namespace=nodc
```

Součástí konfigurace webové stránky je seznam SPARQL dotazů. Tento seznam je uložen v souboru `./components/website/www/src/sparql-queries.yaml`. Po zapsání změn v tomto souboru do repozitáře dojde k automatickému sestavení a publikaci Docker image. V případě změny konfigurace při běžící komponentě Webové stránky je nutné vynutit její přenasazení. To je možné pomocí následujícího příkazu:

```
kubectl rollout restart deploy nodc-website-deployment --namespace=nodc
```

4.1.13 První spuštění

Po první instalaci je třeba vytvořit kopii dat pro další zpracování v Metadatovém procesoru - LinkedPipes ETL. Před vytvořením kopie těchto dat je nutné provést inicializaci modulu. Toho je možné dosáhnout s pomocí Kubernetes Job, které je možné spustit následujícím příkazem:

```
kubectl -f ./k8s/synchronize.yaml
```

Stejně jako dříve je vhodné zkontrolovat úspěšné doběhnutí Jobu. Po jeho doběhnutí by měla instance LinkedPipes ETL obsahovat pipeline. Následně je třeba provést ručním spuštění pipeline `00 - Cache`. Je vhodné opět zkontrolovat správné doběhnutí pipeline.

Volitelně je pak možné smazat záznam o Jobu a jeho definici z Kubernetes pomocí příkazu:

```
kubectl delete job nodc-synchronize-job --namespace=nodc
```

TODO: je třeba upravit po zprovoznění VPN

4.1.14 Manažer

O pravidelné zpracování dat se stará komponenta Manažer. Tuto komponentu lze nasadit pomocí příkazu:

```
kubectl apply -f ./k8s/manager.yaml
```

Čas zpracování je nastaven v souboru `./k8s/manager.yaml` v časové zóně UTC+0.

Zpracování je také možné pustit ručně. V příkazu je nutné nahradit placeholder `{job-name}` unikátním identifikátorem jobu. Identifikátor může být například `nodc-manager-cronjob-manual-20230110`, výsledný příkaz pak vypadá následovně:

```
kubectl create job --namespace=nodc --from=cronjob/nodc-manager-cronjob {job-name
```

4.2 Popis Inštalácie KLIENTSKEJ ČASTI

Klientskou částí je webová stránka pro dotazování nad RDF databází. Tuto část není třeba instalovat.

5. CHYBOVÉ STAVY PRI KONFIGURÁCIÍ

Tato sekce popisuje možné chybové stavy a komplikace v procesu instalace.

5.1 Nelze se přihlásit do OCI Container Registry

V případě pokusu o nahrání GraphDB obrazu můžeme dostat následující chybu: `Error response from daemon: Get "https://eu-frankfurt-1.ocir.io/v2/": unknown: Unauthorized` Důvodem může být absence object-storage-namespace v uživatelském jménu. Uživatelské jméno má vzor `{object-storage-namespace}/{identity-domain}/{user-name}` kde Hodnotu `{object-storage-namespace}` je možné získat pomocí příkazu `oci os ns get` . Hodnota `{identity-domain}` je zadána v procesu přihlášení do OCI.

Podobnou chybu můžeme získat i při nasazení GraphDB do Kubernetes, kdy není možné image stáhnout z OCI Container Registry. V takovém případě je třeba zkontrolovat přihlašovací údaje a Secret `oci-registry-secret` .

6. UMIESTNENIE ZAKLADNEJ DOKUMENTÁCIE

https://github.com/datova-kancelaria/nkod-dokumentacia

Give feedback