

dr inż. Andrzej Sobecki



STUDENCKI KLASTER KATEDRALNY

dr inż. Andrzej Sobecki

STUDENCKI KLASTER KATEDRALNY

© Andrzej Sobecki, Politechnika Gdańska

Dokument opisuje klaster Katedry ASK ETI przeznaczony do realizacji projektów studenckich. W niniejszym przewodniku znajdziesz m.in. informacje o adresacji serwerów, nazwach użytkowników i zalecanych sposobach łączenia z klastrem. Przedstawiono również podstawowe usługi wspólne i zalecany sposób wdrażania własnych usług.

Katedra Architektury Systemów Komputerowych
Politechnika Gdańska

Klaster jest wykorzystywany przez ok. 200 studentów. Uszkodzenie klastra skutkuje oceną 2,0 dla zespołu, który doprowadził do awarii.

PROLOG

Materiały prezentowany w niniejszym dokumencie są aktualne w roku akademickim 2023/2024.

ROZDZIAŁ 1

OPIIS KLASTRA

Jeśli nie masz przy sobie tego dokumentu to szukaj plików README.md w katalogach serwera

Klaster składa się z 4 serwerów: student-swarm01.maas (10.40.71.115), student-swarm02.maas, student-swarm03.maas, student-swarm04.maas. Serwery zostały zwirtualizowane poprzez Ubuntu MAAS i działają pod kontrolą Ubuntu (22.04.3 LTS). Serwery współtworzą obszar do uruchamiania usług w formie skonteneryzowanej. Orkiestratorem kontenerów jest Docker Swarm. Wszystkie usługi wdrażane w klastrze muszą być skonteneryzowane, a zestawy usług opisane w formie kompozycji.

W celu połączenia z klastrem wymagane jest podłączenie do sieci Wydziału ETI - stacjonarnie lub poprzez wydziałowy VPN (z domu). Każdy student wydziału ETI może korzystać z wydziałowego VPN.



Po nawiązaniu połączenia VPN należy utworzyć połączenie SSH do serwera pośredniczącego (bastiona) 172.20.83.101, a następnie połączenie SSH do serwera docelowego tj. student-swarm01.maas. Z tego serwera można wydawać polecenia, które będą realizowane w całym klastrze.

Do logowania należy wykorzystać następujące konto:

- Dla serwera pośredniczącego 172.20.83.101: rsww z hasłem qwe123
- Dla serwera student-swarm01.maas: hdoop z hasłem qwe123.

Gdyby serwer student-swarm01.maas nie odpowiadał to proszę użyć jednego z pozostałych serwerów z puli. Wybrane węzły mają rolę managera w klastrze Swarm i tylko z nich możliwe będzie zlecanie poleceń dla klastra. Status managera może się zmieniać. Mogą Państwo sprawdzić, który węzeł został wskazany jako manager komendą:

```
docker node ls
```

ROZDZIAŁ 2

NOMENKLATURA

Jeśli tworzysz coś na klastrze to stosuj się do opisanego w tym rozdziale nazewnictwa.

Każdy obiekt, który powołujemy do życia w klastrze musi być oznaczony następującym prefiksem: ``<skrót przedmiotu>_<unikalny numer>``. Najłatwiej osiągnąć to wywołując komendę do wdrażania kompozycji, którą opisano poniżej.

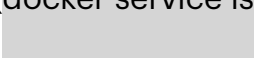
Skrót przedmiotu:

- RSWW - Rozproszone Systemy Wysokiej Wydajności,
- SI - Serwisy Internetowe .NET,
- BE - Biznes elektroniczny.

Numer zespołu jest równy najmniejszemu numerowi albumu przypisanego do członków zespołu tzn. jeśli mamy studentów z numerami albumu (1001, 1002, 1003) to numer zespołu będzie równy 1001.

Najczęściej będą Państwo powoływać do życia usługi, które oferują możliwość obsługi żądań. Możliwość przyjmowania żądań ze świata zewnętrznego przez usługę wymaga skonfigurowania portu do komunikacji. Ta operacja nazywa się wiązaniem portów (ang. binding).

Proszę stosować wiązanie portów jedynie, gdy jest to niezbędne. Do wyboru portu zewnętrznego proszę zastosować poniższą instrukcję.

1. Przyjmij, że **modyfikator** jest równy 0.
2. Weź wszystkie numery albumów członków Twojego zespołu.
3. Weź najmniejszy numer albumu z tego zbioru
4. Usuń ostatnią cyfrę i dodaj wartość modyfikatora.
5. Sprawdź czy **port** o podanym numerze jest wolny (docker service ls)

6. Jeśli port o podanym numerze jest zajęty to numer albumu kolejnej osoby w zespole i wróć do kroku 4.
7. Jeśli port jest wolny to możesz go zająć. [Koniec algorytmu]

8. Jeśli sprawdziłeś/sprawdziłaś wszystkie numery członków zespołu i nie znalazłeś/znalazłaś wolnego portu, to zwiększ modyfikator o 1 (modyfikator++) i przejdź do kroku 3.
9. [Koniec algorytmu]

ROZDZIAŁ 3

DOSTĘPNE NARZĘDZIA

Wdrażając swoje usługi przewiduj potencjalne awarie

Nie należy wdrażać swoich usług, instalując na serwerach dodatkowe pakiety i wiążąc usługi z konkretnym serwerem. W każdej chwili dowolny z serwerów może ulec awarii, podobnie jak w rzeczywistych warunkach.

Dane generowane przez aplikację, które są zapisywane w bazie danych powinny być szczególnie chronione. Dobrym rozwiązaniem będzie utworzenie i wykorzystanie wolumenów.

Klaster działa w trybie **docker swarm**, co oznacza że kontenery będą uruchamiane na jednym z serwerów podłączonych do klastra. Przy czym proces ten będzie dla nas przezroczysty, a za poprawne routowanie żądań do kontenera odpowiada Docker.

Wszystkie usługi jakie chcemy wdrażać na klaster muszą być opisane za pomocą pliku kompozycji (rozszerzony plik

docker-compose.yml). Do wdrożenia usług opisanych taką kompozycją proszę stosować następującą komendę:

```
docker stack deploy -c docker-compose.yml PREFIX --with-registry-auth
```

Zakładamy przy tym, że użytkownik znajduje się w folderze zawierającym plik docker-compose.yml. **PREFIX** jest to unikalny ciąg znaków dla zespołu, tworzony zgodnie z nomenklaturą opisaną powyżej. Do prefixu zostanie dołączona nazwa usługi, zgodnie z konfiguracją w pliku docker-compose.yml.

Po wywołaniu powyższej komendy, docker zarejestruje nowy stos usług tzw. stack. Listę uruchomionych stacków (kompozycji) można zobaczyć, wywołując komendę:

```
docker stack ls
```

Stack składa się z wielu usług, a te składają się z jednego lub więcej kontenera. Listę uruchomionych na klastrze usług można wygenerować, wywołując komendę:

```
docker service ls
```

Zabronione jest dokonywanie zmian lub wyłączanie stacków, usług i kontenerów należących do innych zespołów.

Usługi mogą mieć kilka instancji (kontenerów) takich jak repliki lub stare (wyłączone już) kontenery. Dlatego warto zajrzeć jakie kontenery są powiązane z usługą:

```
docker service ps nazwa_usługi
```

Przy każdym kontenerze zobaczymy unikalne ID_kontenera.

Każdy kontener może generować logi, które mogą być pomocne podczas poszukiwania błędu. Możemy zobaczyć zebrane logi z określonego kontenera, stosując komendę:

```
docker service logs ID_kontenera
```

W przypadku klastra swarm nie ma znaczenia, na którym serwerze kontener został uruchomiony. Sieć zapewnia komunikację informacji pomiędzy serwerami, w tym również logów.

Czasami jakaś usługa wymaga restartu wtedy najlepiej użyć komendy:

```
docker service update <NAZWA_USLUGI> --force
```

Po chwili oczekiwania usługa powinna się uruchomić ponownie. Stan restartu można sprawdzać listując usługi lub kontenery w ramach usługi.

ROZDZIAŁ 4

GDZIE UMIESZCZAĆ SWOJE PLIKI

Staraj się nie zaśmiecać klastra

Każdy zespół studencki może utworzyć swój folder na serwerze **student-swarm01.maas** w `/opt/storage/actina15-20/block-storage/students/projects/students-swarm-services`. Folder powinien mieć nazwę zgodną z nomenklaturą `<PREFIX>_<NR ZESPOLU>`. Wszystkie Twoje wolumeny oraz pliki w `/mnt/block-storage/students/...` nie powinny zajmować więcej niż 500MB.

Wewnątrz tego folderu można umieszczać jedynie pliki niezbędne do uruchomienia kompozycji. Oznacza to, że należy przedtem zbudować niezbędne obrazy, które będą zawierały pliki projektowe.

Zbudowane obrazy należy przekazać do rejestru obrazów. Następnie można je wskazać w plikach `docker-compose.yml` w swojej kompozycji.

Budowanie obrazu:

```
docker build . -t <adres registry>/PREFIX_nazwa
```

Wysyłanie zbudowanego obrazu do rejestru:

```
docker push <adres registry>/PREFIX_nazwa
```


ROZDZIAŁ 5

WSPÓLNY SERWER BAZODANOWY

Nie twórz serwera bazodanowego SQL i NoSQL specjalnie dla swojego projektu - oszczędzaj zasoby

W klastrze działają wspólne serwery bazodanowe. Mogą Państwo korzystać w swoich projektach z takich usług wspólnych jak: POSTGRESQL, MYSQL, MONGODB. Postgresql jest dostępny na porcie 5432, MySQL na porcie 3306, MongoDB na porcie 27017. Stan usług i oferowane porty można sprawdzić wywołując komendę:

```
docker service ls
```

Wszystkie zespoły są zobowiązane wykorzystywać jeden z tych serwerów jeśli ich kompozycja wymaga tego typu usługi. W tym celu należy utworzyć sobie bazę danych o nazwie zgodnej z opisaną powyżej nomenklaturą prefiksu.

Dla studentów nieobytych w konsoli, przygotowano interfejs webowy z dostępem do serwera bazodanowego. Został on

wystawiony na porcie 9080. Mogą też Państwo podłączyć swoje aplikacje GUI do opisanych serwerów.

Hasło do serwera bazodanowego Postgresql: **student**

Domyślny użytkownik dla serwera MongoDB to **root** z hasłem **student**. Dla osób poszukujących interfejsu GUI wystawiono również usługę mongo-express, dostępną na porcie 8081. Dane do logowania są identyczne. Swoje bazy danych należy również nazywać zgodnie z nomenklaturą z ew. dopiskiem _1, _2 itd.

Dla bazy MySQL udostępniono aplikację dostępną przez przeglądarkę na porcie 9099.

ROZDZIAŁ 6

WOLUMENY

Dane zapisane w kontenerze nie są trwałe

Jeśli projekt wymaga przechowywania danych pomiędzy wyłączeniami kontenerów to musimy utworzyć (w swoim stacku) wolumen i zamontować go do kontenera. W związku z użyciem serwerów przez wiele zespołów należy zwracać uwagę, aby pojedynczy zespół nie wykorzystywał więcej niż 500MB na wszystkie swoje wolumeny i przestrzeń w **/mnt/block-storage/students**.

Kontroluj zasoby poleceniem:

```
df -h
```

Najczęstsze awarie klastra w zeszłych latach były wywołane przez przepełnienie dysków. Wtedy zalogowanie po ssh również nie będzie możliwe. Inne problemy dotyczyły nieumiejętnego usuwania plików.

Pamiętaj również, że aktualizując swoją kompozycję wyłączasz automatycznie stare kontenery i tworzysz nowe. Jednak nikt automatycznie nie usunie starych kontenerów i

starych obrazów. Znajdź swoje stare kontenery i obrazy, a następnie je usuń.

```
docker images
```

```
docker service ps nazwa_usługi
```

Jeśli poszukujesz serwera, na którym uruchomiono kontener i chcesz to wykonać polecenia:

docker node ls - wyświetli listę serwerów w klastrze,

*docker node **nazwa_serwera** ps* - wyświetli kontenery uruchomione na wybranym serwerze.

Warto rozróżniać operacje wykonane per serwer (np. *docker ps*) od operacji wykonywanych per klaster (np. *docker service ps*).

ROZDZIAŁ 7

DOSTĘP DO USŁUG

Debugowanie jest równie proste

W przypadku klastra typu SWARM pomiędzy węzłami uruchomiona jest sieć typu MESH. Oznacza to, że wywołanie dowolnego adresu IP serwera należącego do klastra umożliwia wywołanie dowolnej usługi uruchomionej w tym klastrze - niezależnie od tego gdzie pracuje aktualnie kontener.

Dlatego jeśli wystawiono usługę httpd (port 80) -> (port 1001) to wywołanie tej usługi będzie możliwe na każdym adresie serwerów student-swarm01, student-swarm02, student-swarm03, student-swarm04 na porcie 1001.

Ze względu na konfigurację sieci, klaster nie jest publicznie dostępny z sieci Internet. Przed wywołaniem usługi należy przygotować sobie środowisko. W kolejnym rozdziale znajdą Państwo pomocne wskazówki co należy zrobić.

ROZDZIAŁ 8

TWORZENIE TUNELU

Do utworzenia tunelu wystarczy protokół SSH

Załóżmy, że chcemy zobaczyć usługę wystawioną w klastrze na porcie **80**. Pamiętaj, że mowa tutaj o porcie zbindowanym do klastra SWARM, a nie porcie na jakim usługa nasłuchuje w kontenerze. Oznacza to, że w swojej kompozycji musisz powiązać port z kontenera **XYZ** z portem w klastrze (w tym przypadku 80).

Następnie należy zestawić połączenie VPN ze swojego komputera do sieci wydziałowej. Każdy student wydziału ma prawo z niego korzystać. Cały ruch wychodzący powinien przechodzić przez VPN.

Następnie korzystając z konsoli (lub putty) należy wydać komendę podobną do tej:

```
ssh -L 5242:student-swarm01.maas:80 rsww@172.20.83.101
```

W systemie Windows należy uruchomić program putty lub podobny, a następnie przejść po lewej stronie (w drzewku) do opcji SSH -> Tunnels i ustawić opcje zgodnie z poleceniem powyżej.

