

1. Pierwszym krokiem było umieszczenie na Dockerhub'ie obrazu docker'owego z komunikatorem.

Otagowanie obrazu i umieszczenie na Dockerhub'ie:

```
suhar@Suhar-PC:~$ sudo docker tag 9041a34b5a30 mrsuhar/komunikator-build:lab11
suhar@Suhar-PC:~$ sudo docker push mrsuhar/komunikator-build:lab11
The push refers to repository [docker.io/mrsuhar/komunikator-build]
da5a0ab0a55c: Pushed
ffb649051097: Pushing [=====>] 579.1MB/947.7MB
501149a27c86: Pushed
8169d0601aeb: Pushed
84ba2fd70d4f: Pushed
1a56ae6dc8b4: Pushed
e8fc7988bb8b: Pushed
d6b9195f39e3: Pushed
01dc7c7a3d44: Pushed
af5b7ecd4d95: Pushed
ae6fc1f6045a: Pushed
8324dcc2315f: Pushed
31e73fdd65f0: Pushed
093501b0a9e2: Pushing [=====>] 463.8MB/510MB
b1169e57b139: Pushed
b3577d595e75: Pushed
```

Ze względu na niepowodzenie w stage('Deploy') na Dockerhub'ie umieściłem działający obraz stworzony lokalnie.

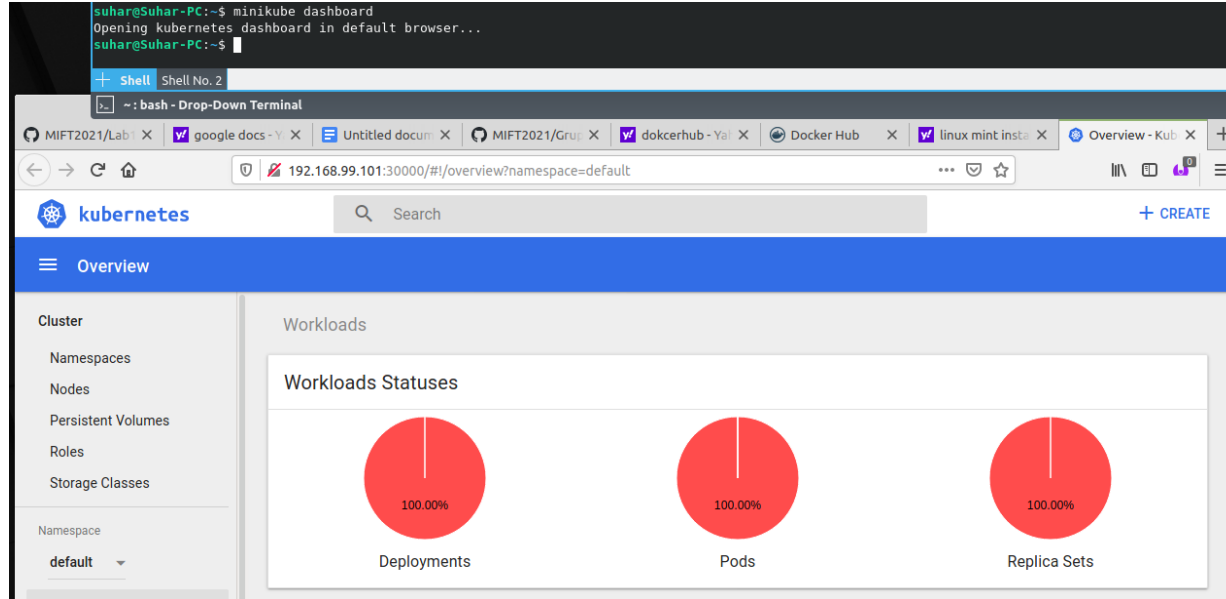
2.Kolejnym krokiem była instalacja Minicube. W tym celu konieczna była zmiana ustawienia SVM w BIOS'ie , instalacja Virtualbox'a. Niestety nie udokumentowano konkretnych poleceń wykorzystanych podczas tego kroku.

3. 1 Uruchomienie klastra,minicube dashboard.

Uruchomienie klastra:

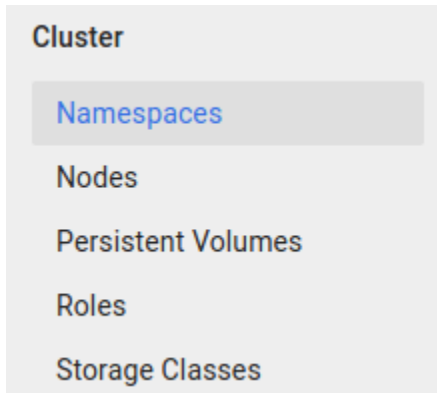
```
suhare@Suhar-PC:~$ minikube start
Starting local Kubernetes v1.9.0 cluster...
Starting VM...
Getting VM IP address...
Moving files into cluster...
Setting up certs...
Connecting to cluster...
Setting up kubeconfig...
Starting cluster components...
Kubectl is now configured to use the cluster.
Loading cached images from config file.
```

Minicube Dashboard:



3.2 Co można zrobić w dashboard'zie?

Przeglądanie parametrów klastra:



Namespace'y ułatwiają podział zadań w klastrze.

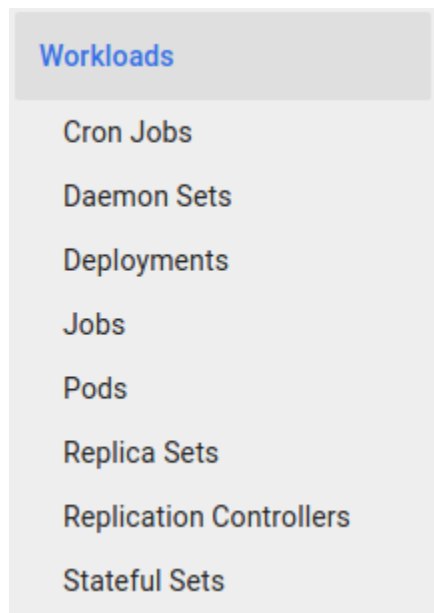
Node'y to elementy klastra wykonujące zadania i kontrolowane przez panel.

Persistent Volume służy do zarządzania pamięcią do przechowywania danych w klastrze (sprawia, że sposób żądania i przydzielania pamięci jest uniwersalny niezależnie od systemu operacyjnego w konkretnym nod'zie) .

Role's to zarządzanie rolami w klastrze.

Storage Class służy do określenia rodzaju przechowywanych danych w klastrze

Przeglądanie zadań:

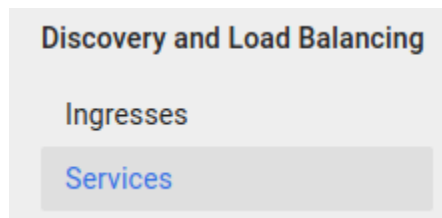


Pod'y to konkretne zadaniq wykonywane przez klaster,którym przydzielana jest jakaś część zasobów obliczeniowych i pamięciowych klastra.

(Cron) Jobs służą do automatycznego uruchamiania konkretnych pod'ów.

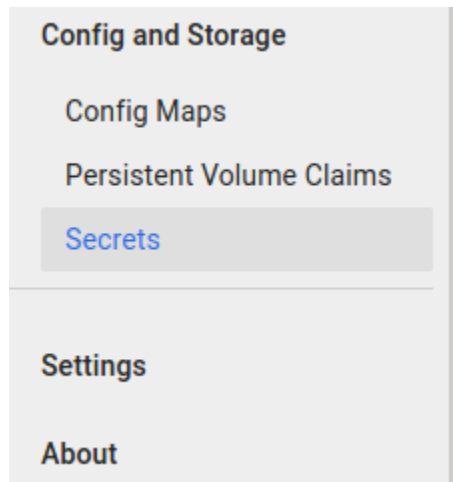
Set'y służą do wykorzystywania pod'ów w określony sposób zależnie od ich typu (Daemon,Replica,Stateful).

Przeglądanie serwisów:



Ingress to API zarządzające zewnętrznym dostępem do klastra.
Services to lista usług obecnie oferowanych przez klaster.

Ustawienia przechowywania danych, i tokenów dostępu.



Config Map's służą do przechowywania niezaszyfrowanych danych w formie klucz:wartość, są wykorzystywane do umożliwienia przenoszenia aplikacji/serwisu między różnymi środowiskami.

Persistent Volume Claims pozwala ustawić żądania dostępu do pamięci (nie w sensie obliczeniowym, a jako miejsca przechowywania danych)

Secrets zawiera zaszyfrowane dane np. Token dostępu do Kubernetes'a.

4. Uruchomienie kontenera w Kubernetes jako usługi:

Terminal:

```
suhar@Suhar-PC:~$ kubectl create deployment komunikator-build --image=mrsuhar/komunikator-build:lab11
deployment "komunikator-build" created
suhar@Suhar-PC:~$ kubectl expose deployment komunikator-build --type=NodePort --port=7891
service "komunikator-build" exposed
suhar@Suhar-PC:~$ kubectl get services komunikator-build
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
komunikator-build   NodePort    10.106.8.108  <none>         7891:30115/TCP   32s
suhar@Suhar-PC:~$ minikube svc hello-minikube
Error: unknown command "svce" for "minikube"

Did you mean this?
    service

Run 'minikube --help' for usage.
suhar@Suhar-PC:~$ minikube service komunikator-build
Waiting, endpoint for service is not ready yet...
```

Dashboard:

The screenshot shows the Kubernetes Dashboard interface. The top navigation bar includes the Kubernetes logo, a search bar, and a '+ CREATE' button. The main content area is titled 'Services' and displays a table of services. The table has columns for Name, Labels, Cluster IP, Internal endpoints, External endpoints, and Age. Two services are listed: 'komunikator-build' and 'kubernetes'. The 'komunikator-build' service is highlighted with a green checkmark and shows its internal and external endpoints. A terminal window is visible in the foreground, showing the command 'minikube service komunikator-build' being executed.

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
komunikator-build	app: komunikator-build	10.106.8.108	komunikator-build:7891 TCP komunikator-build:30115 TCP	-	a minute
kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	6 hours

Działania kontenera w klastrze:

Details

Name: komunikator-build-89f4b4c9b-7sqxv

Namespace: default

Labels: app: komunikator-build pod-template-hash: 459060756

Creation Time: 2021-05-26T13:50 UTC

Status: Running

QoS Class: BestEffort

Network

Node: [minikube](#)

IP: 172.17.0.5

Containers

komunikator-build

Image: mrsuhar/komunikator-build:lab11

Environment variables: -

Commands: -

Args: -

Błędy podczas działania:

[illegible]

Jak widać udało się uruchomić obraz w klastrze niestety ze względu na błędy nie działa on w pełni poprawnie.