nAxiom

Wdrożenie nAxiom w Openshift - krok po kroku

Wersja nAxiom: 1.11.3

1. Wprowadzenie

W tym artykule opisano kroki procedury wdrożenia witryny nAxiom z użyciem oprogramowania Red Hat OpenShift Container Platform. W procedurze używane są pliki konfiguracyjne z ustawieniami domyślnymi. Informacje uzupełniające na temat uruchamiania nAxiom w kontenerach zawiera artykuł Wdrożenie nAxiom z obrazów Docker.

2. Procedura wdrożenia

1. Zainstaluj platformę Red Hat OpenShift (https://www.redhat.com/en/technologies/cloud-computing/openshift).

2. Skopiuj pliki wymagane do uruchomienia aplikacji i utwórz następującą strukturę w katalogu wdrożenia (na przykład *C:/OpenShift*):

```
├───certs
│       o7t-internal.crt
│       o7t-internal.key
│       proxy-cert.crt
│       proxy-cert.key
│
├───configs
│   ├───admin
│   │       config-custom.json*
│   │
│   ├───api
│   │       nlog-custom.config*
│   │
│   ├───documentation
│   │       appsettings-custom.json*
```

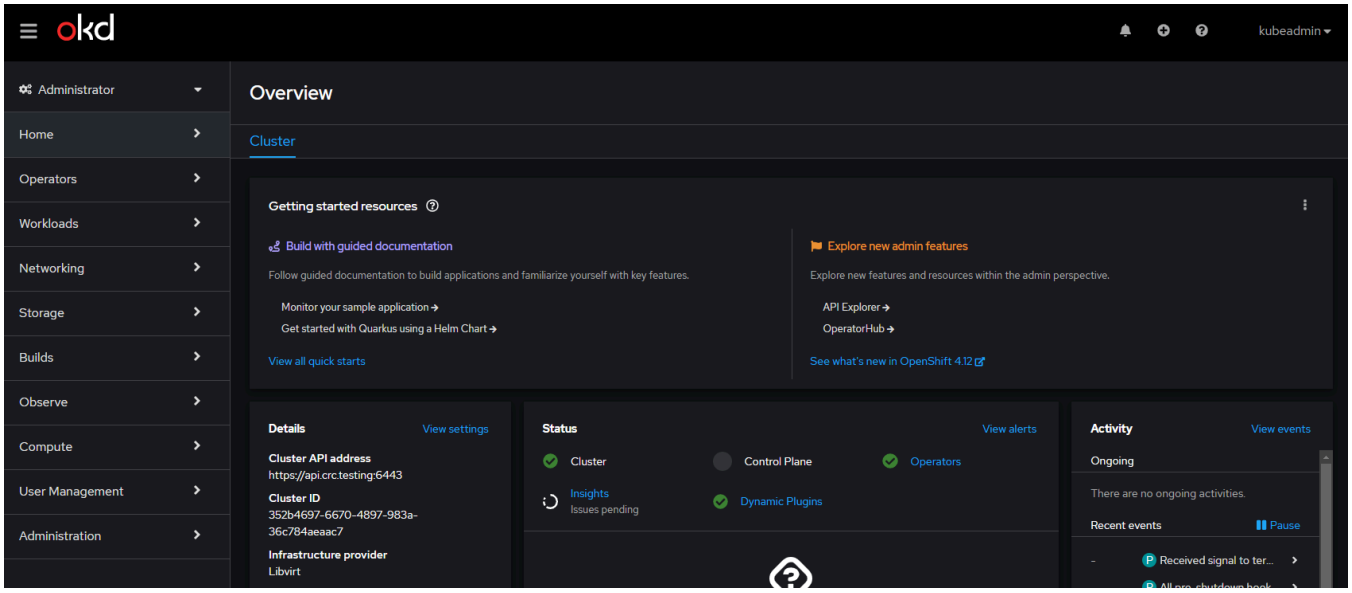
```
|   |
|   ├───front
|   |       config-custom.json*
|   |
|   ├───mobile-api
|   |       appsettings-custom.json*
|   |
|   ├───nginx
|   |       naxiom-buffers.conf
|   |       naxiom-certificate.conf
|   |       naxiom-headers.conf
|   |       naxiom-openshift.conf
|   |
|   ├───public-api
|   |       appsettings-custom.json*
|   |
|   ├───task-service
|   |       appsettings-custom.json*
|   |
|   ├───telerik-reports
|   |       appsettings-custom.json*
|   |
|   ├───tenant-admin
|   |       config-custom.json*
|   |
|   └───workflow
|           config-custom.json*
|
├───deployments
|       admin-deployment.yaml
|       api-deployment.yaml
|       auth-deployment.yaml
|       documentation-deployment.yaml
|       front-deployment.yaml
|       mobile-api-deployment.yaml
|       mssql-deployment.yaml
|       nginx-deployment.yaml
```

```
│       public-api-deployment.yaml
│       syncfusion-api-deployment.yaml
│       task-service-deployment.yaml
│       telerik-reports-deployment.yaml
│       tenant-admin-deployment.yaml
│       tenant-api-deployment.yaml
│       workflow-deployment.yaml
│
├───pvc**
│       mssql-persistent-volume-claim.yaml
│
├───routes
│       naxdev-local-route.yaml
│       wildcard-naxdev-local-route.yaml
│
├───scc
│       scc.yaml
│
├───scripts
│   ├───syncfusion-api
│   │       custom-entrypoint.sh*
│   │
│   └───telerik-reports
│           custom-entrypoint.sh*
│
├───secrets
│   ├───api
│   │       appsettings-custom.json*
│   │
│   ├───auth
│   └───tenant-api
│           appsettings-custom.json
│
└───services
        admin-service.yaml
        api-service.yaml
        auth-service.yaml
```

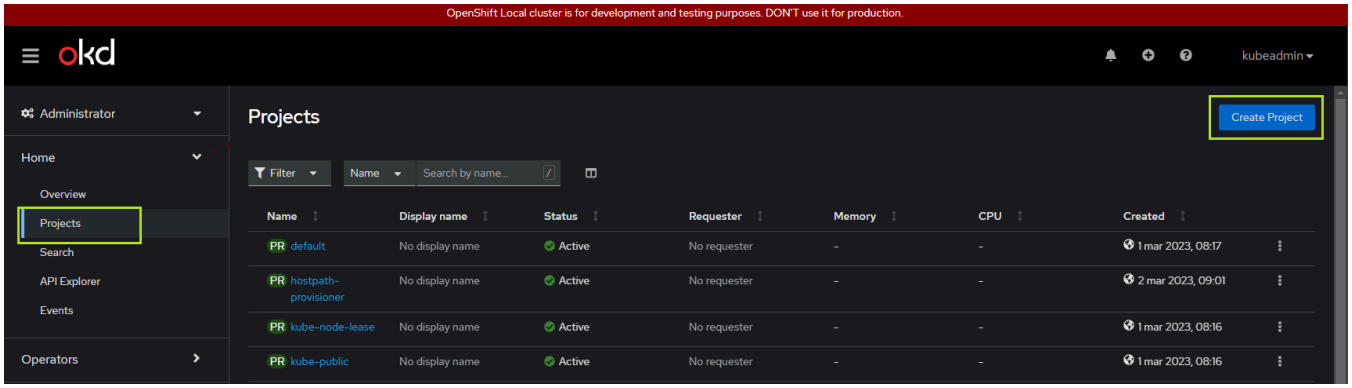
```
        documentation-service.yaml
        front-service.yaml
        mobile-api-service.yaml
        mssql-service.yaml
        nginx-service.yaml
        public-api-service.yaml
        syncfusion-api-service.yaml
        task-service-service.yaml
        telerik-reports-service.yaml
        tenant-admin-service.yaml
        tenant-api-service.yaml
        workflow-service.yaml
```

* pliki opcjonalne z niestandardowymi ustawieniami konfiguracyjnymi
** folder *PVC* (Persistent Volume Claim) jest potrzebny do zapisu danych z instancji serwera SQL uruchamianej w kontenerze (*mssql*). W przypadku korzystania z innej instancji ten folder nie jest potrzebny, a w procedurze można pominąć kroki 11, 12 i 13.

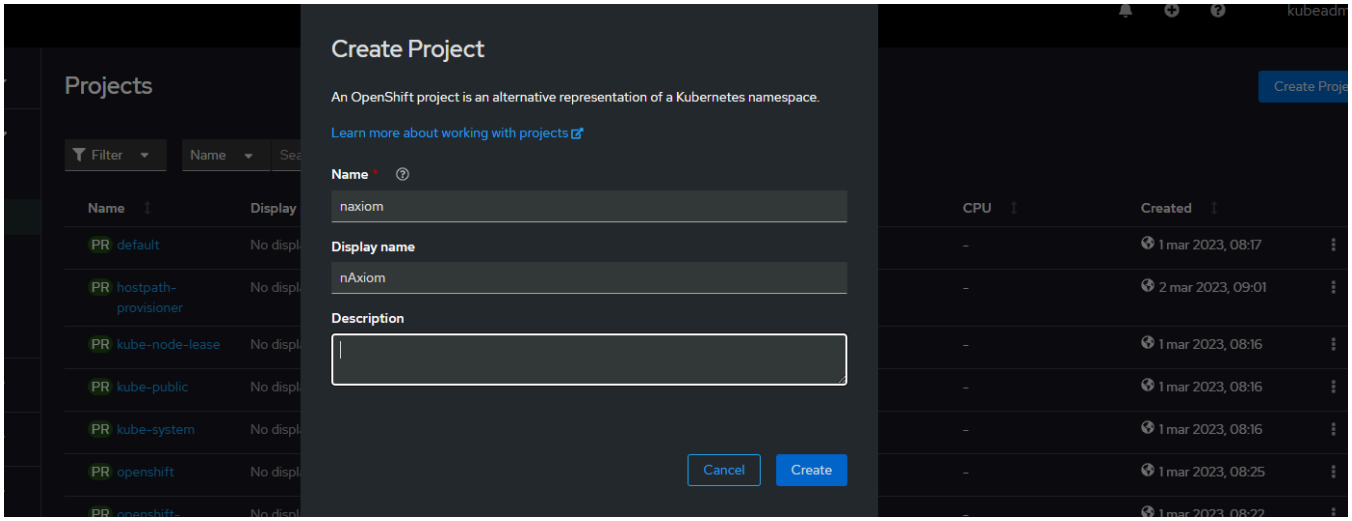
3. Uruchom konsolę Openshift w przeglądarce, pod adresem URL https://console-openshift-console.apps-crc.testing/. Zostanie wyświetlona następująca strona:



4. Utwórz nowy projekt:



Nadaj mu dowolną nazwę, na przykład *nAxiom*, i kliknij przycisk `Create`.

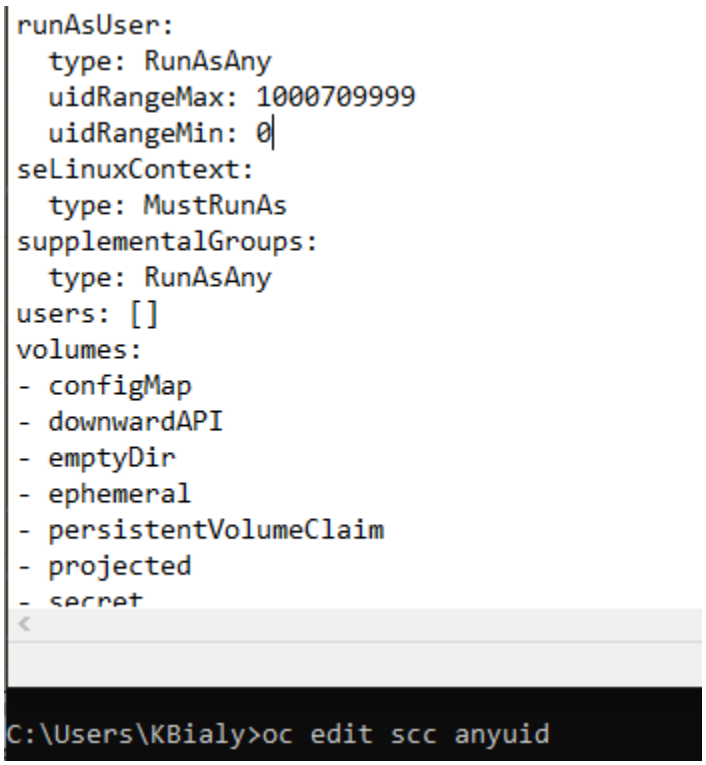


5. Wczytaj obrazy do rejestru lokalnego (do nowej przestrzeni nazw *naxiom*).

6. (Wymagane uprawnienia administratora). Uruchom terminal i wpisz:

```
oc edit scc anyuid
```

znajdź sekcję *runAsUser* i dodaj następujące linie:



7. Wczytaj obiekty *SecurityContextConstraints*. W tym celu uruchom następujące polecenie, aby wczytać plik *scc.yaml* do platformy OpenShift (wymagane uprawnienia administratora):

```
oc create -f ./scc/scc.yaml -n naxiom
oc adm policy add-scc-to-user run-as-root -z default -n naxiom
```

Ten plik powinien mieć następującą zawartość:

```
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: run-as-root
  allowPrivilegedContainer: false
runAsUser:
  type: RunAsAny
  uidRangeMax: 1000709999
  uidRangeMin: 0
seLinuxContext:
  type: MustRunAs
fsGroup:
```

```
  type: MustRunAs
  supplementalGroups:
  type: RunAsAny
```

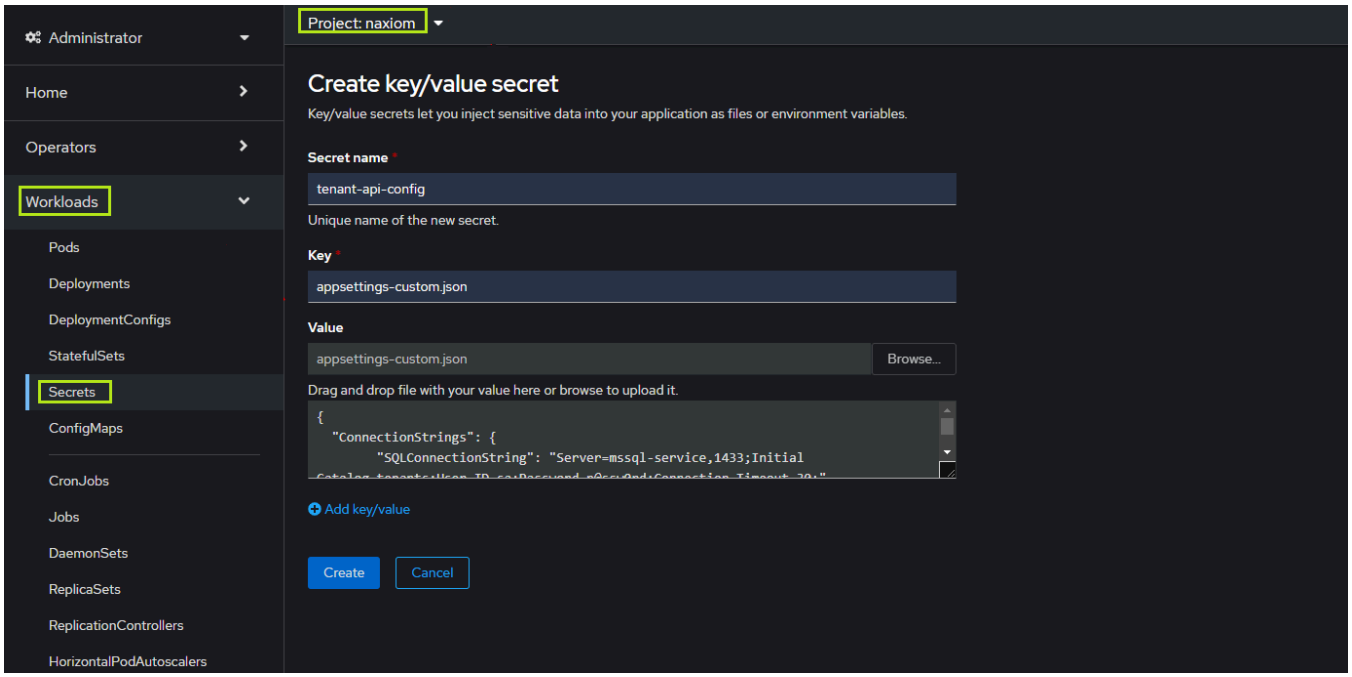
8. Jeśli nie masz certyfikatów, wygeneruj je, na przykład używając polecenia `openssl` dla środowiska lokalnego. Można także użyć programu `letsencrypt` lub innego płatnego urzędu certyfikacji. Szczegóły dotyczące certyfikatów opisano w osobnym artykule Wdrożenie nAxiom z obrazów Docker.
Następnie uUtwórz obiekty *tls secret* dla certyfikatów. Trzeba to zrobić z terminala, ponieważ interfejs przeglądarkowy nie ma takiej opcji. Uruchom następujące polecenia:

```
oc create secret tls cert-naxiom-internal --cert=./certs/o7t-internal.crt
    --key=./certs/o7t-internal.key -n naxiom

oc create secret tls cert-naxiom-proxy --cert=./certs/proxy-cert.crt
      --key=./certs/proxy-cert.key -n naxiom
```

9. Teraz utwórz niezbędne obiekty wdrożenia. Można do tego użyć poleceń terminala `oc` lub interfejsu przeglądarkowego.
Najpierw utwórz obiekt *secret* wymagany do uruchomienia aplikacji (plik *appsettings-custom.json* dla serwisu *tenant-api* z parametrami połączenia z bazą danych). Z menu `Workloads` z lewej strony wybierz pozycję `Secrets`, kliknij przycisk `Create` i wybierz `key/value secret`. W wyświetlone pola wpisz wartości jak na ilustracji (wskaż plik *appsettings-custom.json* dla serwisu *tenant-api*).

> Uwaga: pamiętaj, aby tworzyć obiekty *secret* i inne w przestrzeni nazw swojego projektu (w tym opisie *naxiom*).

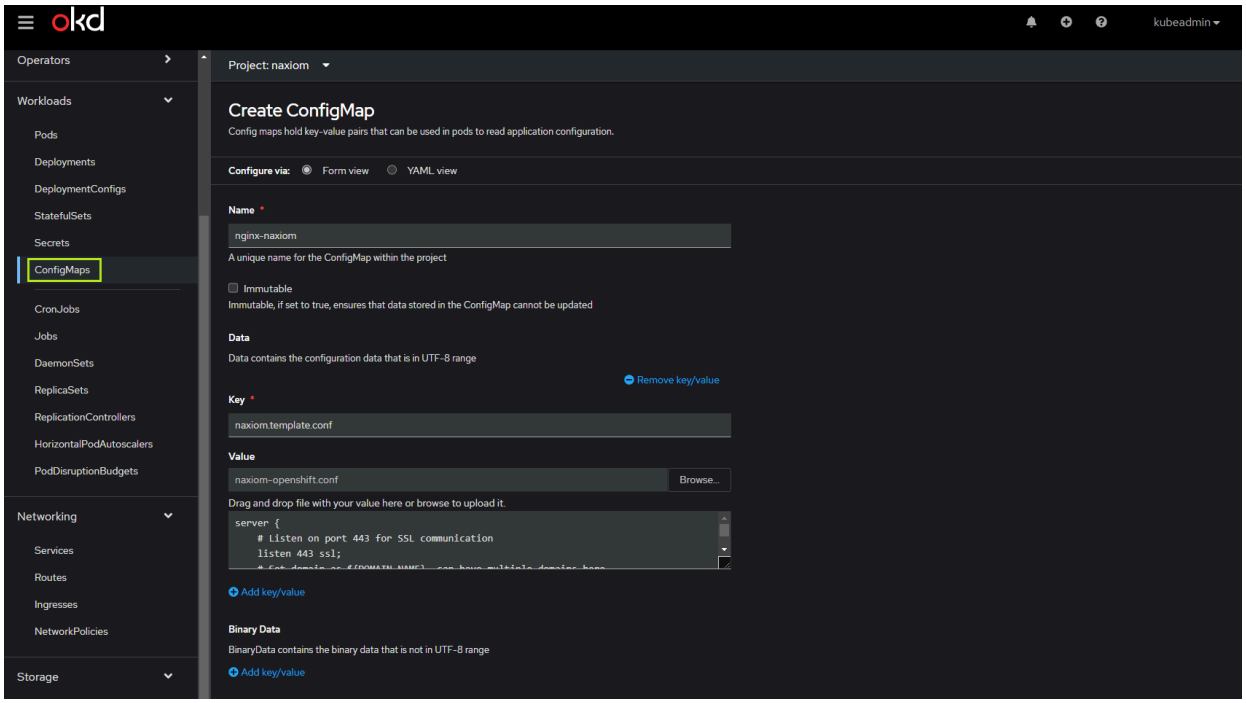


10. Utwórz obiekty *configmaps*. Jest to wymagane dla usługi *nginx*. Dla innych serwisów te obiekty nie są wymagane, o ile mają one działać w konfiguracji domyślnej zapisanej w obrazach. Serwis *nginx* używa 4 obiektów *configmap*. Można je utworzyć w konsoli. Jeden z tych obiektów jest szablonem rozwiązywanym podczas uruchamiania kontenera nginx przy użyciu polecenia *envsubst*. (Szczegóły znajdziesz we wpisie *command*, w pliku *deployment* dla serwisu nginx). Utwórz następujące obiekty *configmap*:

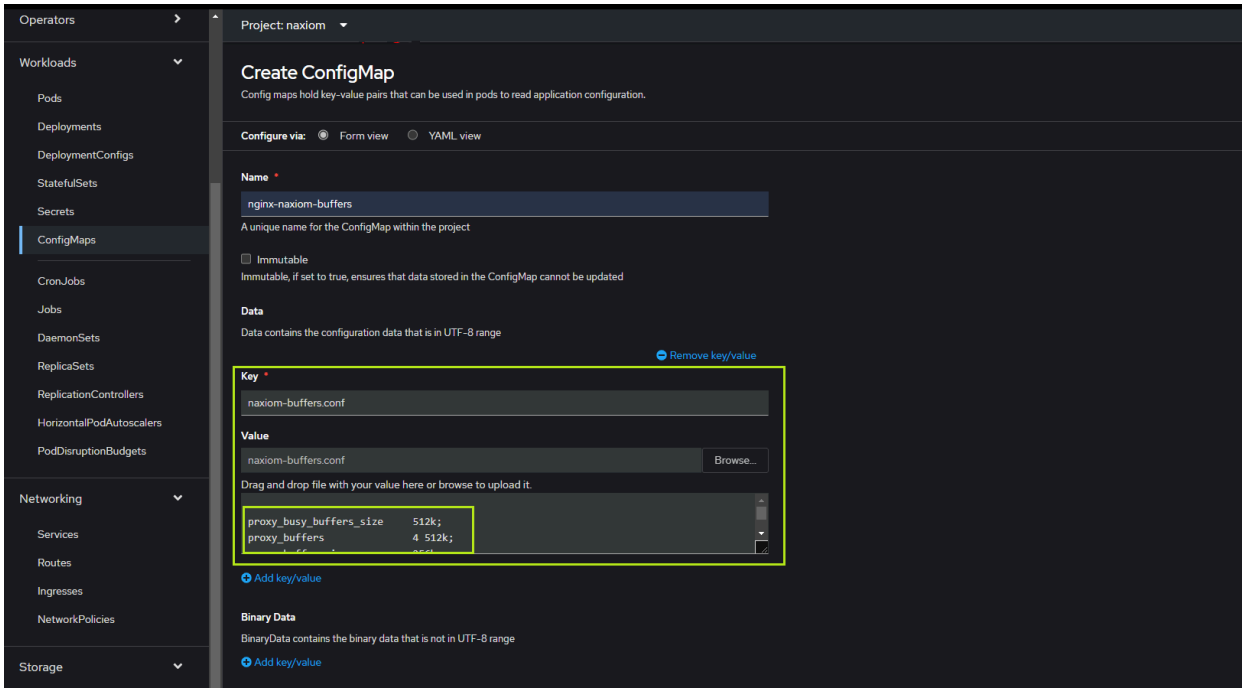
- *nginx-naxiom*:

> Uwaga: ze względu na dłuższy czas przetwarzania rozbudowanych żądań zaleca się wydłużenie limitu czas dla operacji odczytu, nawiązywania połączenia i wysyłania z domyślnych 60 s na 300 s. W tym celu w konfiguracji serwera nginx należy dodać następujące wpisy:

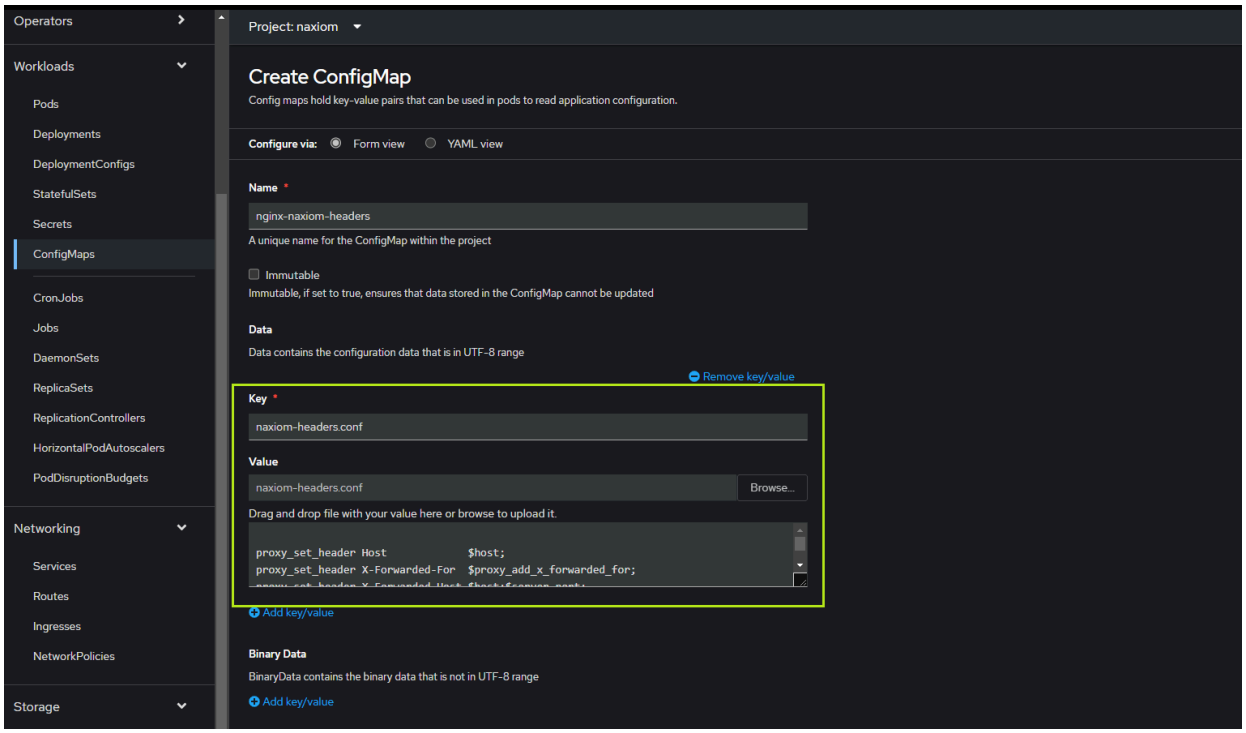
```
proxy_read_timeout 300s;
proxy_connect_timeout 300s;
proxy_send_timeout 300s;
```



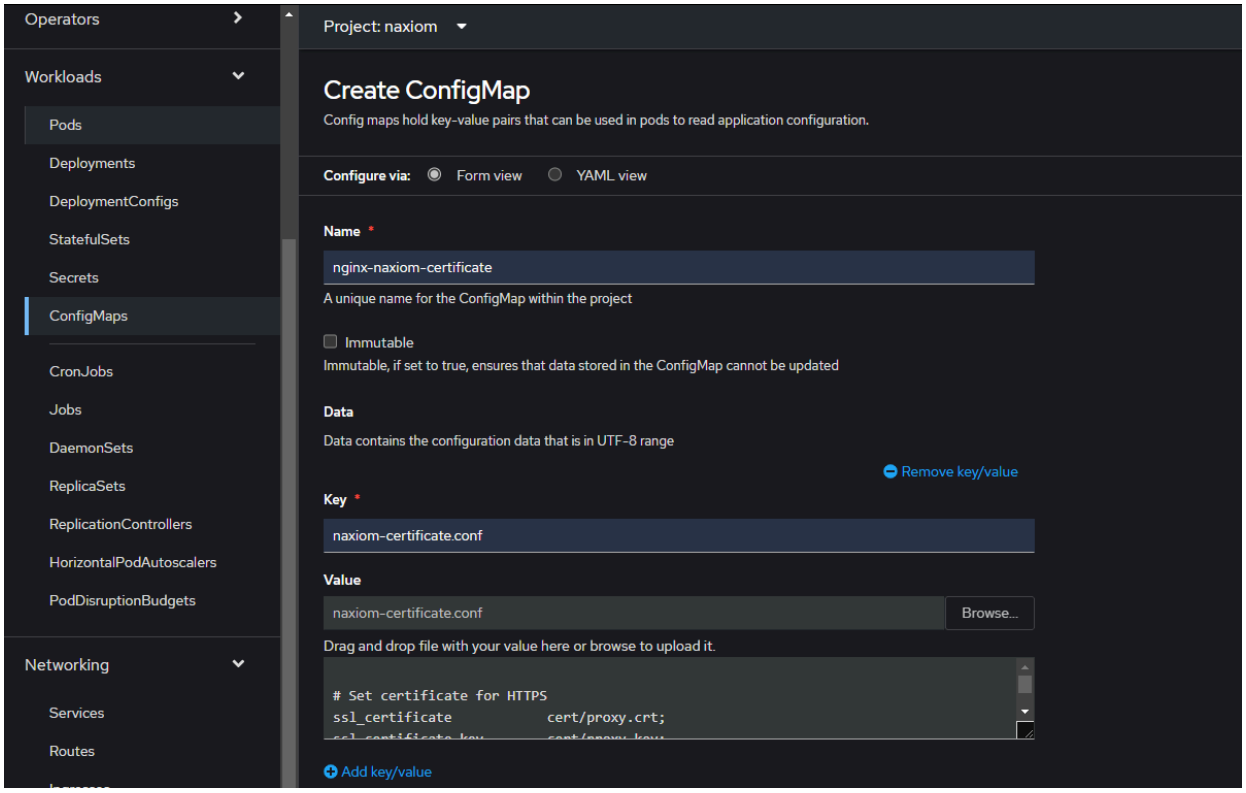
- *nginx-naxiom-buffers*:



- *nginx-naxiom-headers*:

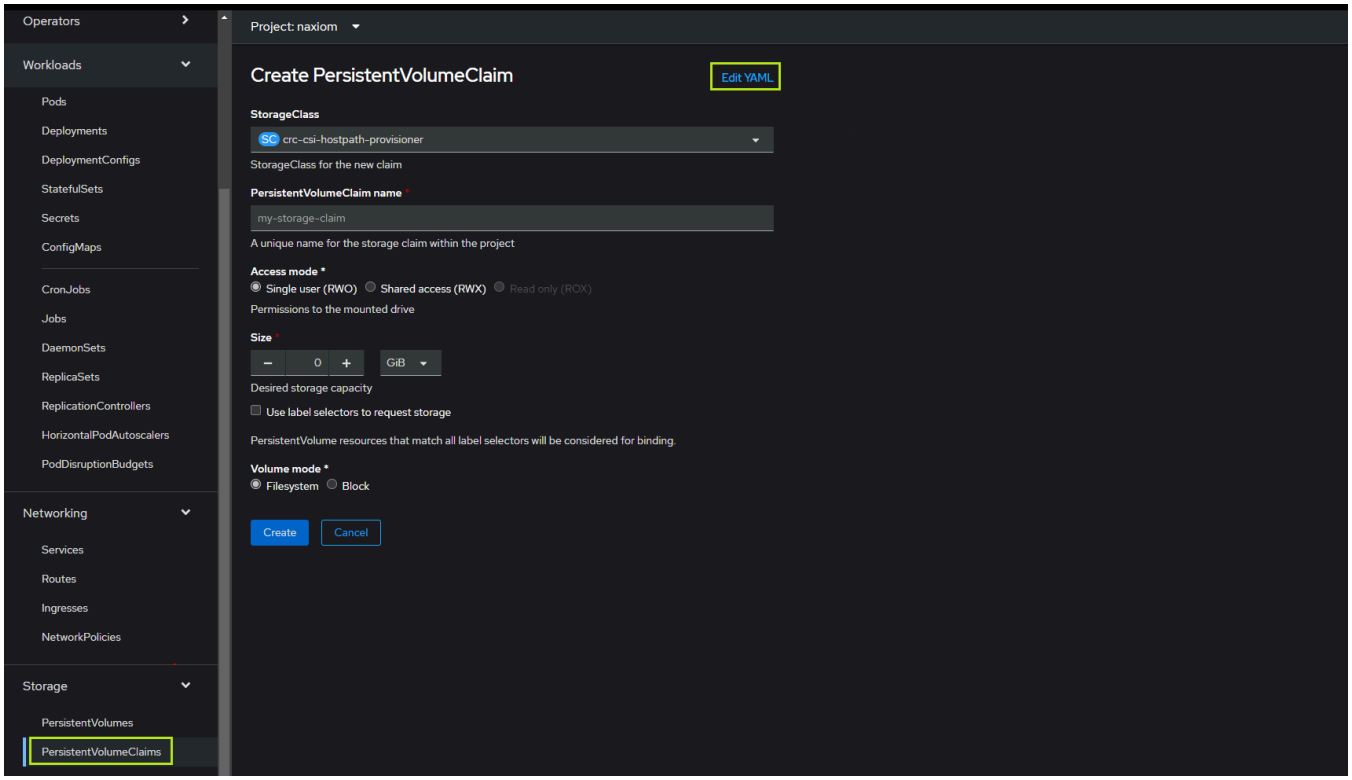


- *nginx-naxiom-certificate*:

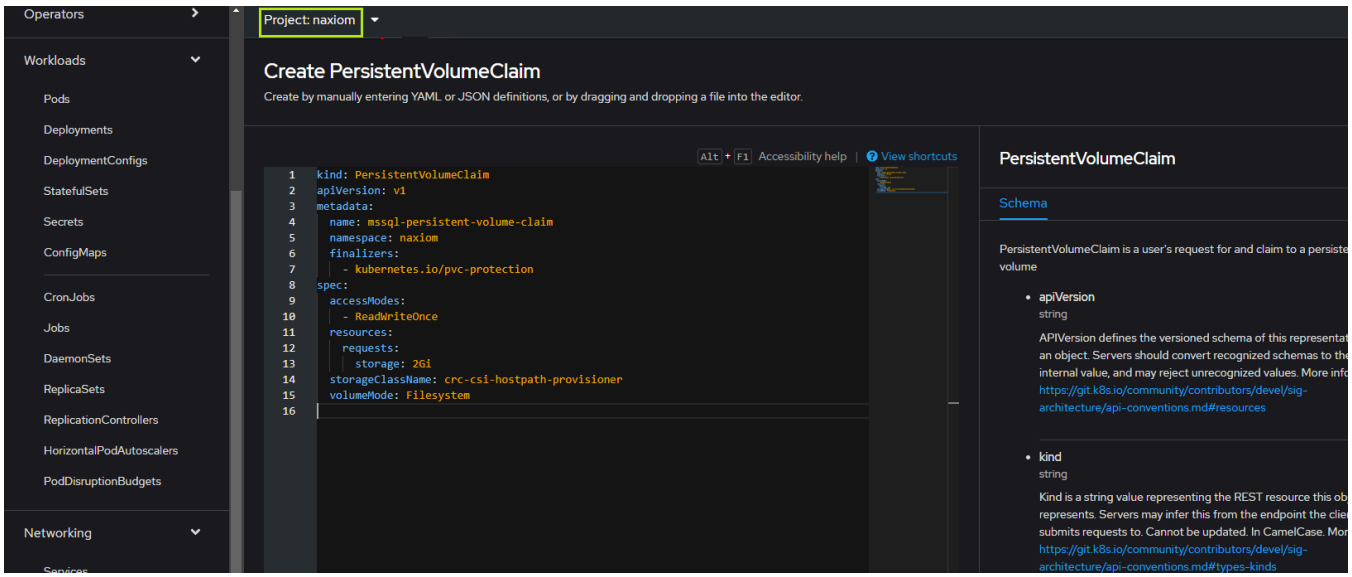


Dopilnuj, aby nazwy kluczy zostały wpisane poprawnie (tak jak na ilustracjach powyżej).
W opisywanym przykładzie wdrożenia serwis nginx jest używany w przestrzeni nazw wdrażanej witryny. Można go jednak skonfigurować do działania we osobnej przestrzeni nazw i obsługę wielu witryn nAxiom. Wymaga to jednak skonfigurowania kierowania ruchu do kilku domen.

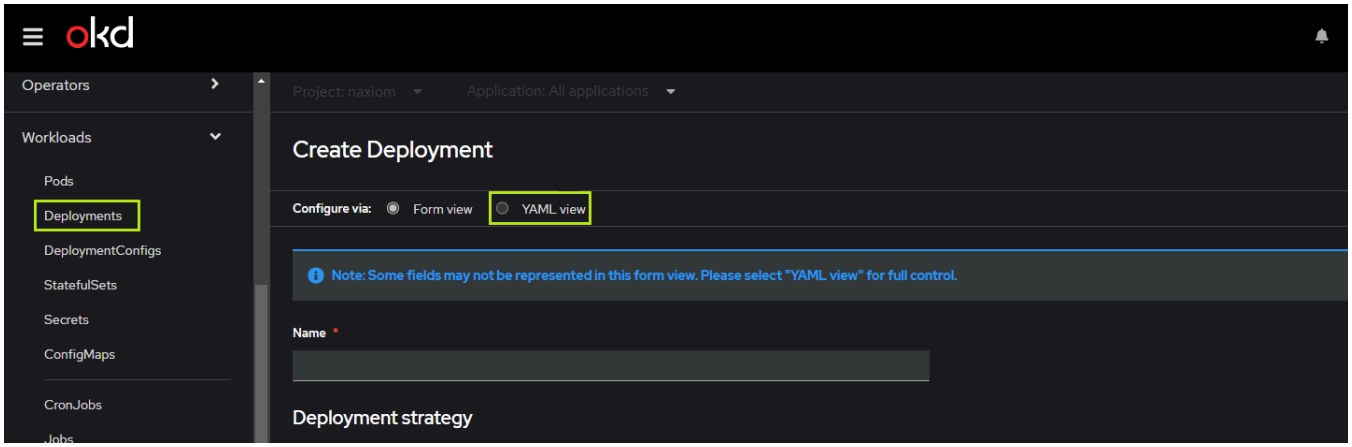
11. Utwórz zasób PVC (*persistent volume claim*) dla serwisu MSSQL. Ten krok (i dwa następne) można pominąć, jeśli zamiast poda MSSQL w klastrze OpenShift ma być używana lokalna instancja SQLEXPRESS. Kliknij pozycję `Persistent Volume Claim` w menu `Storage`. Kliknij przycisk `Create PersistentVolumeClaim` i kliknij `Edit YAML` (parametry zostaną podane w pliku YAML w folderze *pvc*).



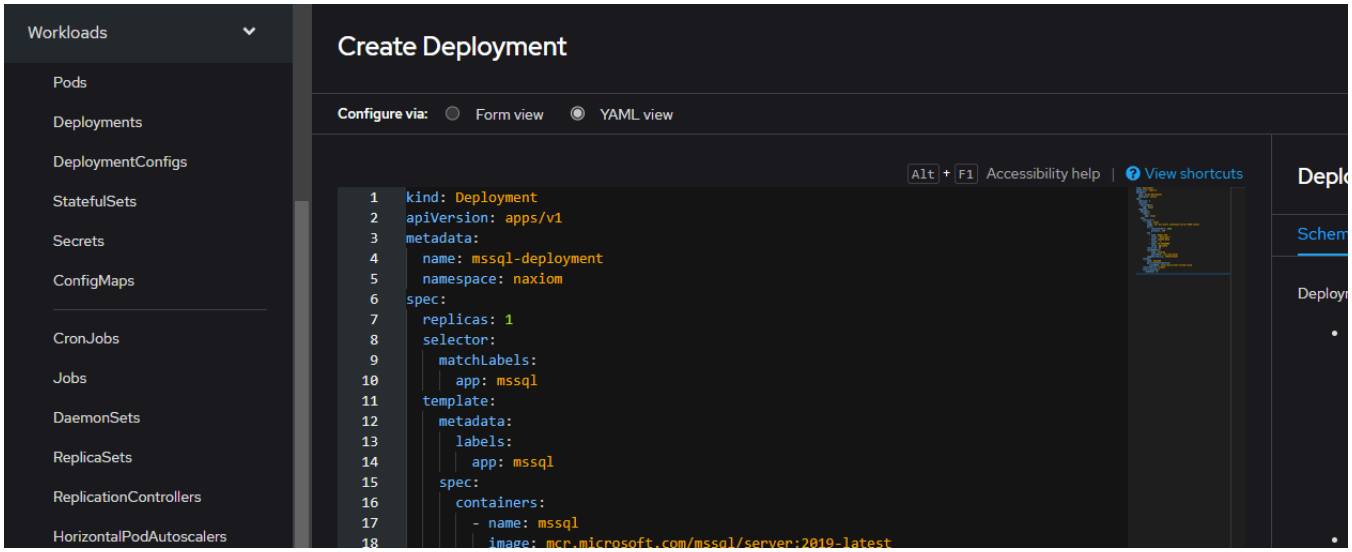
Wklej zawartość pliku *yaml* do edytora (zwróć uwagę, czy przestrzeń nazw to *naxiom*) i kliknij przycisk `Create`. Zasób PVC będzie miał status *Pending*, co oznacza, że jest gotowy do użycia.



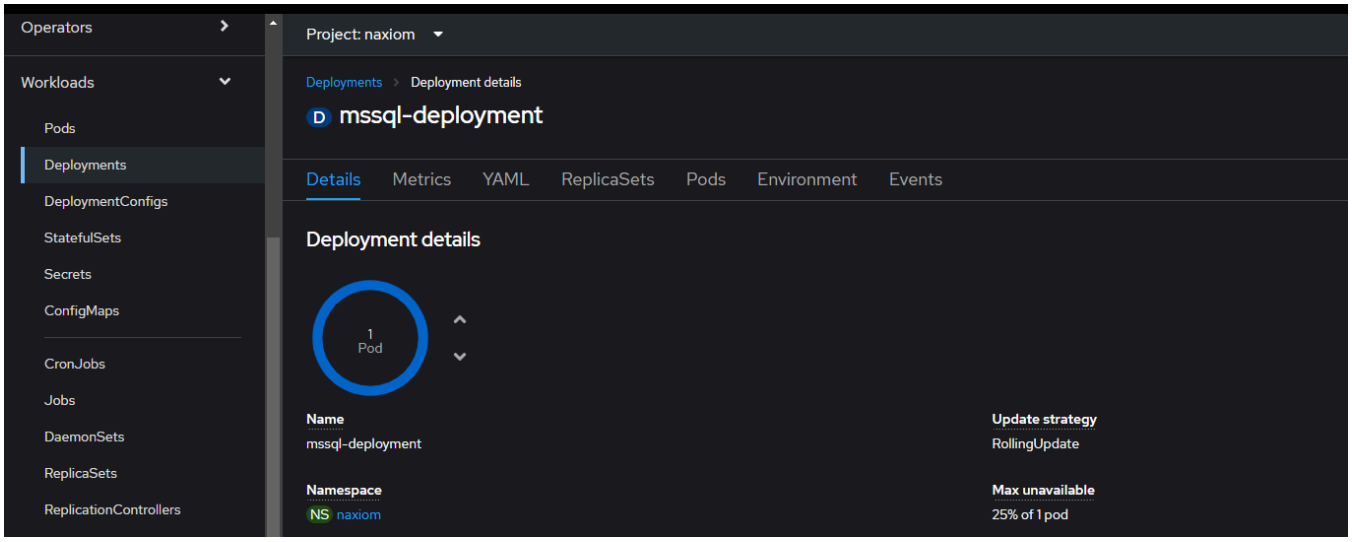
12. Utwórz obiekty *deployment* i *service*. Najpierw dla serwisu *mssql*. Kliknij pozycję *Deployments*, kliknij przycisk *Create* i zaznacz opcję `YAML view`.



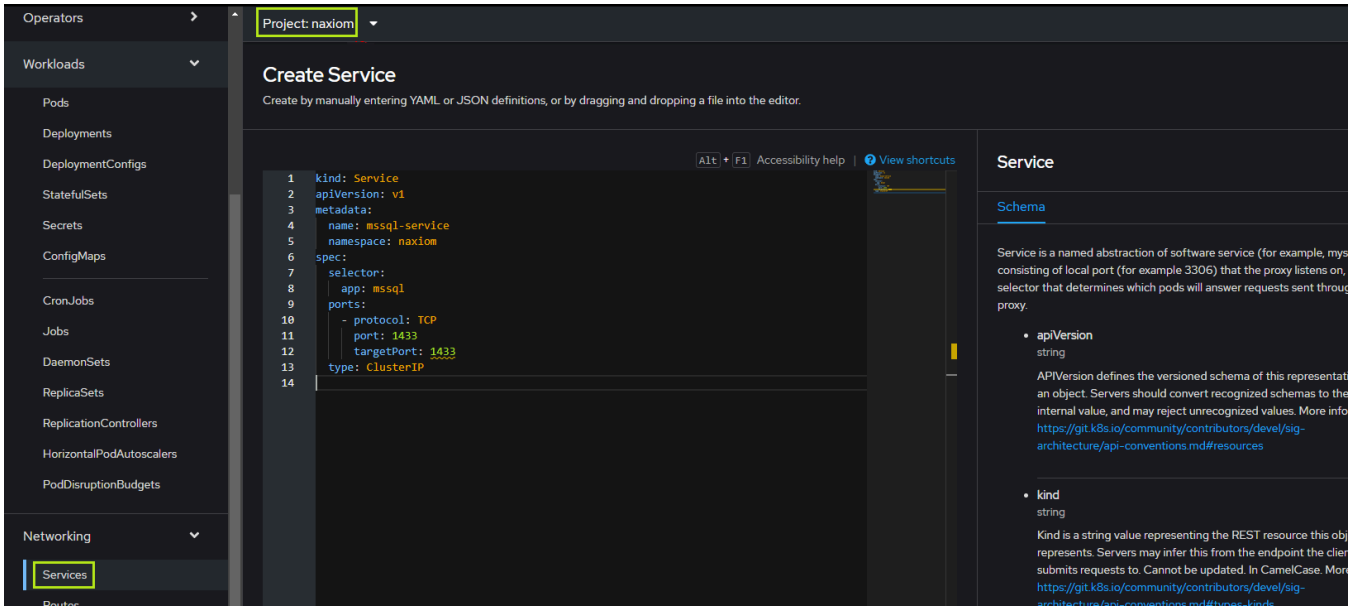
Wklej zawartość pliku *mssql-deployment* i kliknij przycisk `Create`.



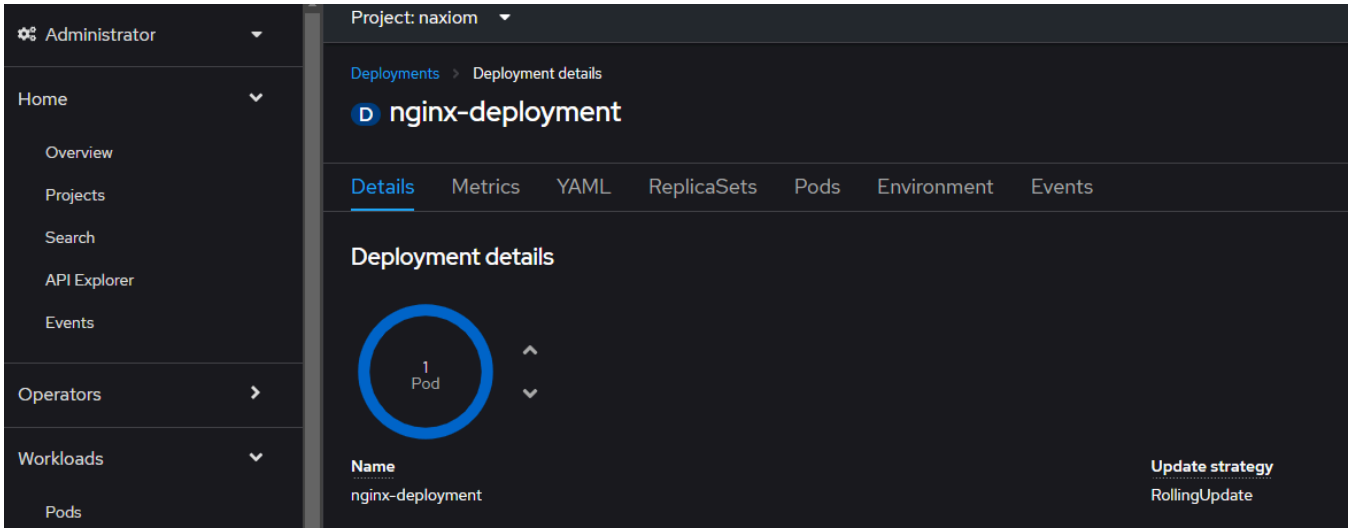
Zostanie wyświetlony pulpit wdrożenia. Poczekaj aż system pobierze obraz z repozytorium mssql. Kiedy wdrożenie będzie gotowe do uruchomienia, zostanie wyświetlony niebieski okrąg:



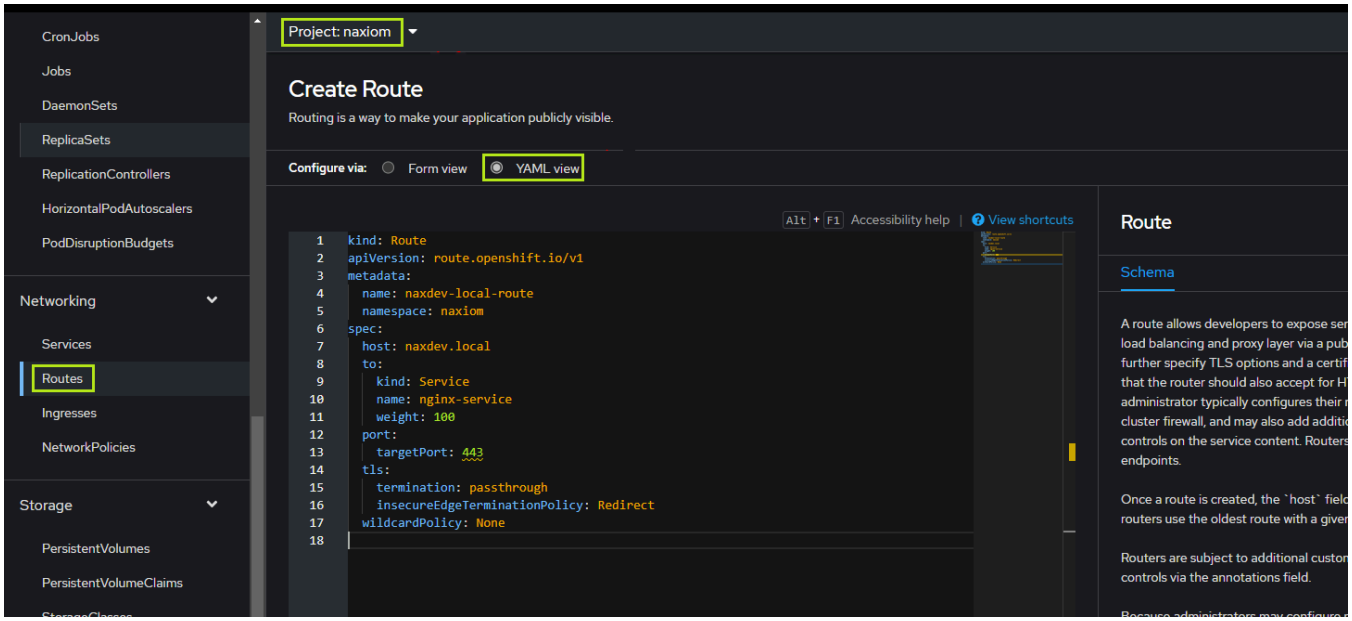
13. Utwórz obiekt *service* dla poda *mssql*, aby umożliwić komunikację z innymi podami. Kliknij pozycję `Services` w menu `Networking` i wklej zawartość pliku *mssql-service.yaml*.



14. Utwórz obiekty *deployment* i *service* dla serwisu nginx w analogiczny sposób jak w krokach 12 i 13.



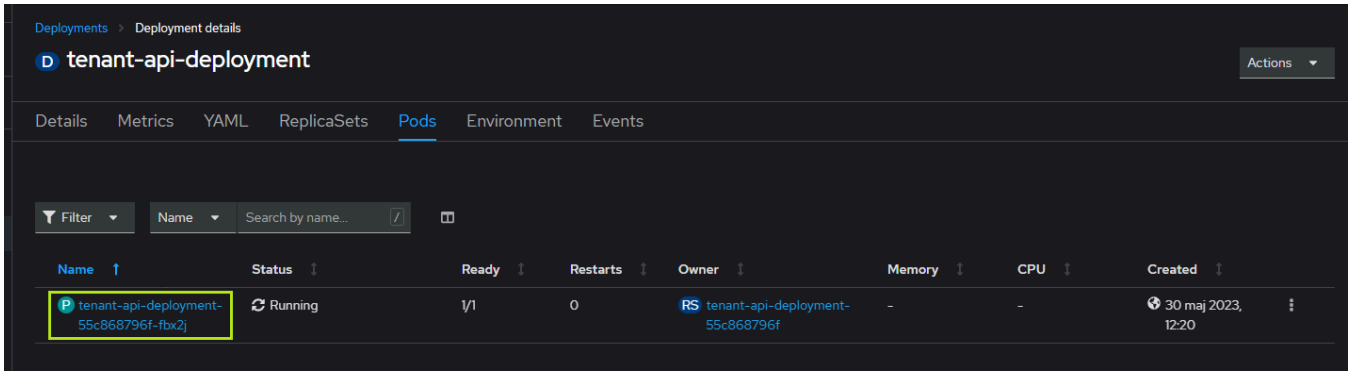
15. Utwórz obiekty *route*, które będą odpowiadać za kierowanie ruchu do serwisu nginx. Kliknij pozycję `Routes` w menu `Networking`, kliknij przycisk `Create Route` i zaznacz opcję `YAML view`.



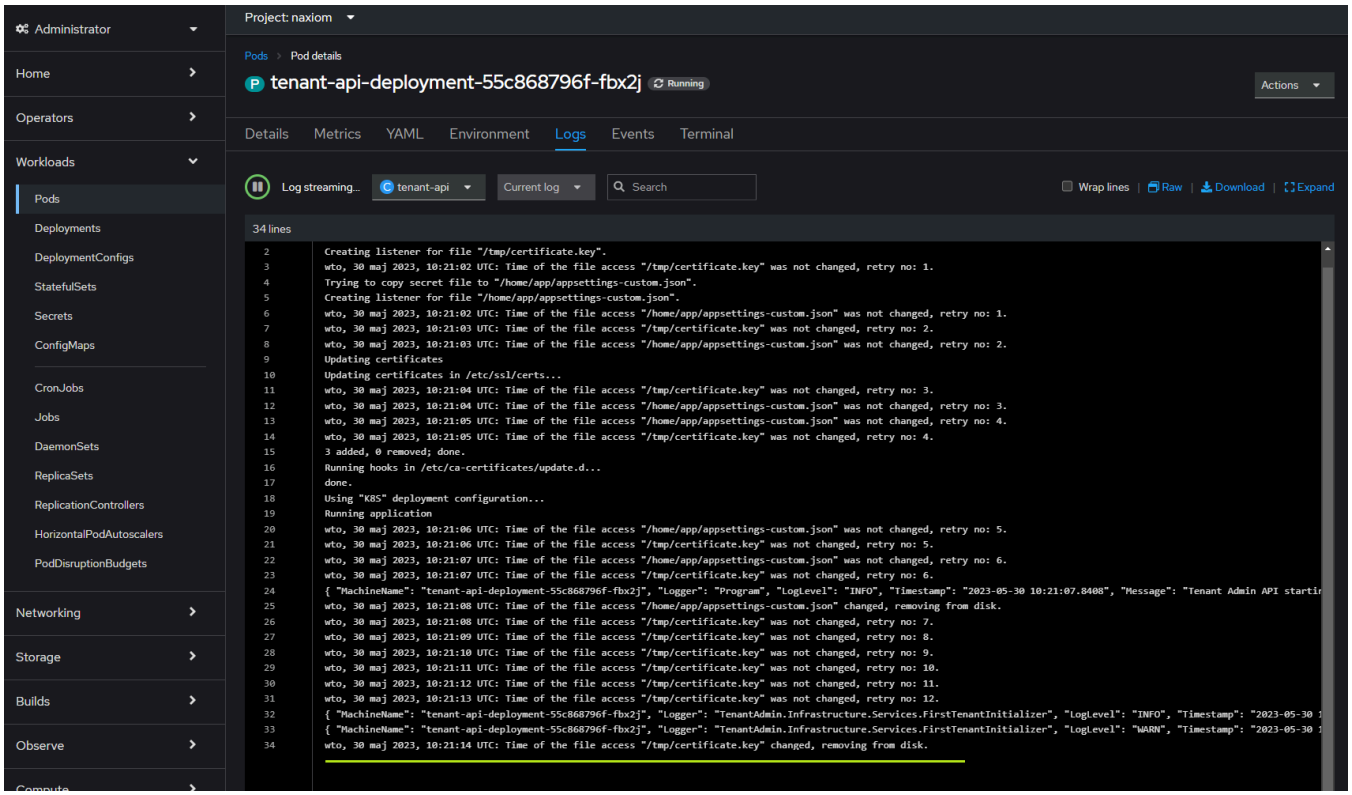
Dodaj dwa obiekty *route* zdefiniowane w plikach w folderze *routes*: jeden dla domeny i jeden z obsługą poddomen (wymaganą do obsługi wielu tenantów).

16. Sprawdź, czy domena *naxdev.local* jest dostępna w przeglądarce (certyfikaty samopodpisane nie będą działać w przeglądarce Firefox, która korzysta z własnego magazynu certyfikatów). Błąd 502 jak na ilustracji poniżej oznacza, że serwis nginx działa, należy tylko utworzyć obiekty *deployment* i *service* dla serwisów nAxiom. Wystąpienie innego błędu będzie oznaczało, że ruch nie dociera do serwisu nginx.

17. Utwórz obiekty *deployment* i *service* dla serwisów *tenant-api* i *tenant-admin* analogicznie jak dla serwisów *mssql* i *nginx*. Aby sprawdzić, czy konfiguracja w pliku *tenant-api-deployment* jest poprawna, kliknij zakładkę Pods:

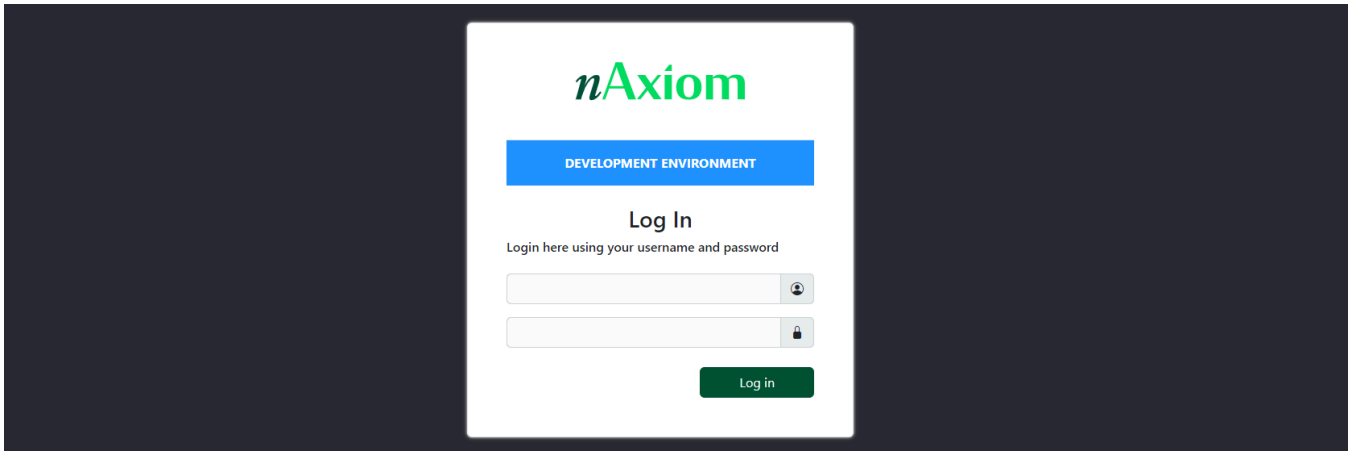


Kliknij oznaczony pod, a następnie kliknij zakładkę Logs, jak na ilustracji poniżej:



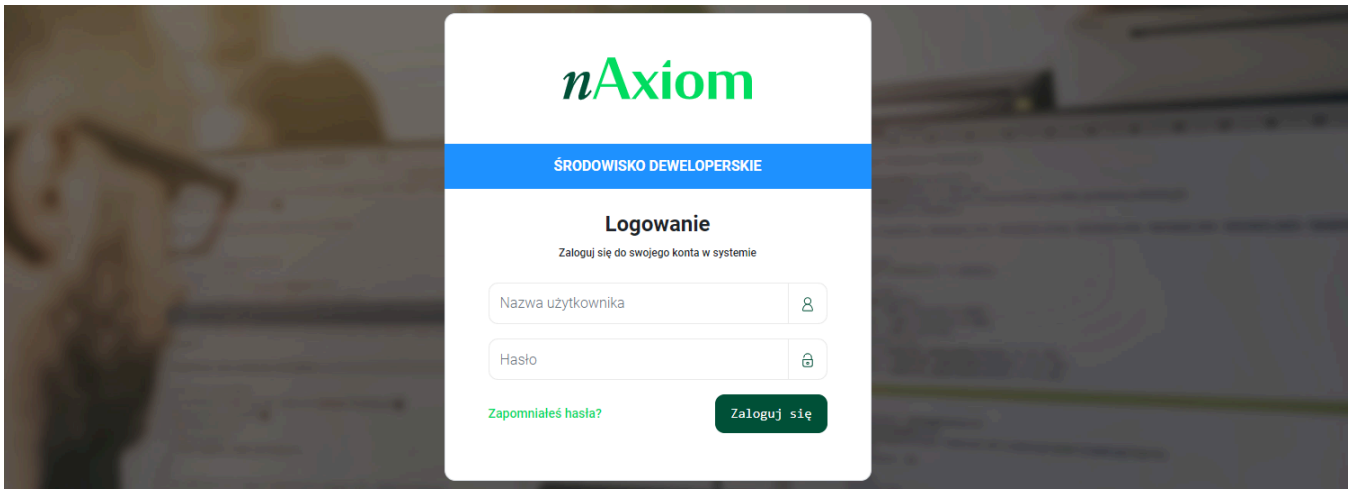
Jeśli log zawiera wpis oznaczony na ilustracji, serwis *tenant-api* działa prawidłowo.

18. Wpisz w przeglądarce adres https://naxdev.local/tenatnsadmin. Powinna zostać wyświetla strona logowania do serwisu *tenant-admin*:

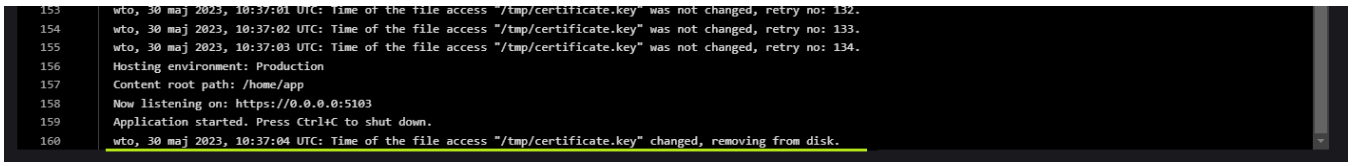


Zaloguj się i utwórz pierwszego tenanta.

19. Utwórz obiekty *deployment* i *service* dla pozostałych serwisów nAxiom. Następnie sprawdź, czy pod adresem https://naxdev.local jest dostępna witryna nAxiom. Jeśli cały proces przebiegł poprawnie, zostanie wyświetlona strona logowania.



Zależnie od liczby dodanych tenantów uruchomienie serwisu *api* może potrwać kilka minut. Podobnie jak w przypadku serwisu *tenant-api*, znakiem powodzenie będzie obecność wpisu oznaczonego na ilustracji w logu poda:


wto, 30 maj 2023, 10:37:01 UTC: Time of the file access "/tmp/certificate.key" was not changed, retry no: 132.
wto, 30 maj 2023, 10:37:02 UTC: Time of the file access "/tmp/certificate.key" was not changed, retry no: 133.
wto, 30 maj 2023, 10:37:03 UTC: Time of the file access "/tmp/certificate.key" was not changed, retry no: 134.
Hosting environment: Production
Content root path: /home/app
Now listening on: https://0.0.0.0:5103
Application started. Press Ctrl+C to shut down.
wto, 30 maj 2023, 10:37:04 UTC: Time of the file access "/tmp/certificate.key" changed, removing from disk.