Instrukcja tworzenia oraz instalacji aplikacji służącej do zbierania i analizy danych metrycznych pochodzących z zewnętrznych api giełdowych

1. **Instalacja narzędzia Docker**Docker jest narzędziem, które pozwala umieścić program oraz jego zależności w lekkim, przenośnym, wirtualnym kontenerze, który można uruchomić na prawie każdym serwerze z systemem [GNU/Linux](https://pl.wikipedia.org/wiki/GNU/Linux).

Link do pobrania dla systemu Windows: https://docs.docker.com/docker-for-windows/install/#download-docker-for-windows

Instalacja w systemie Ubuntu przy pomocy programu Putty:

|  |
| --- |
| $ sudo apt-get install docker-ce |

1. **Instalacja narzędzi do zbierania i analizy danych metrycznych przy użyciu narzędzia Docker**
   1. **Instalacja bazy danych InfluxDB**

InfluxDB jest narzędziem open source, służącym do przechowywania danych metrycznych. Została zoptymalizowana pod kątem szybkości zapisu oraz odczytu serii danych dotyczących monitorowania operacji, analizy danych w czasie rzeczywistym czy zbierania danych z aplikacji.

Poniżej zamieszczono komendę instalującą bazę InfluxDb o nazwie ‘millionaires’ przy pomocy narzędzia Docker.

|  |
| --- |
| $ docker run -p 8086:8086 -e INFLUXDB\_DB=millionaires -e INFLUXDB\_USER=admin -e INFLUXDB\_USER\_PASSWORD=secretpassword influxdb |

* 1. **Instalacja narzędzia Grafana do wizualizacji danych**

Grafana jest narzędziem open source, które służy do analizowania i wizualizacji danych. Dzięki licznej bazie źródeł, z których Grafana może pobierać dane może służyć jako scentralizowane narzędzie do przedstawiania różnych metryk.

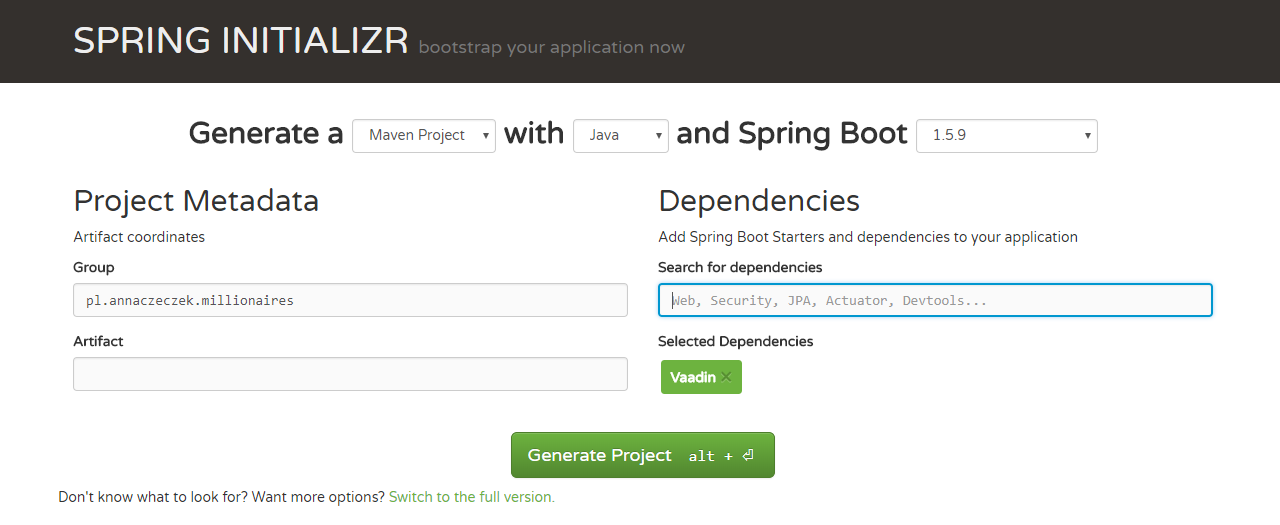
|  |
| --- |
| $ docker run -d -p 3000:3000 \  -v /var/lib/grafana:/var/lib/grafana \  -e "GF\_SECURITY\_ADMIN\_PASSWORD=secret" \  grafana/grafana |

1. **Tworzenie aplikacji służącej do zbierania danych**

Głównym zadaniem aplikacji webowej jest odczytywanie co 5 minut bieżących danych pochodzących z czterech zewnętrznych api giełdowych- dwóch Polskich (BitBay i BitMarket) oraz dwóch amerykańskich (BitFinex i GDAX). Dane pochodzące z api są zapisywane w bazie danych InfluxDB indeksowanej po czasie, a następnie wyświetlane przy użyciu narzędzia Grafana. Wybrany wykres został następnie osadzony w aplikacji webowej dostępnej pod adresem aniacz.pl.

1. **Wygenerowanie projektu przy użyciu Spring Initializr**

Narzędzie dostępne pod adresem: <https://start.spring.io/>



1. **Korzystanie z zewnętrznego api na podstawie BitBay 2.0**

Api zwraca podstawowe statystyki w postaci obiektu JSON, np.

{"max":4500,"min":1465,"last":1533,"bid":1513,"ask":1542,"vwap":1524.42,"average":1545.67,"volume":4.54042857, "bids" {...}, "asks": {...}, "transactions": {…}}

Składnia odwołań: https://bitbay.net/API/Public/[Waluta 1][Waluta 2]/[Kategoria].json

**[Waluta 1]-** obowiązkowa, skrót kryptowaluty (BTC lub LTC)

**[Waluta 2]-** opcjonalna, w przypadku niepodania jako domyślne ustawia USD

**[Kategoria]-** typ odwołania:

* + **trades**- ostatnie transakcje
  + **orderbook**- oferty z giełdy
  + **market**- oferty z giełdy i ostatnie transakcje
  + **ticker**- podstawowe statystyki
  + **all**- wszystkie powyższe informacje zebrane w całość

np. https[://bitbay.net/API/Public/BTCPLN/orderbook.json](https://bitbay.net/API/Public/BTCPLN/orderbook.json)

1. **Połączenie do api**
   1. **Cryptowatch**https://api.cryptowat.ch
   2. **BitBay 2.0**

https://bitbay.net/pl/api-publiczne

* 1. **BitMarket**

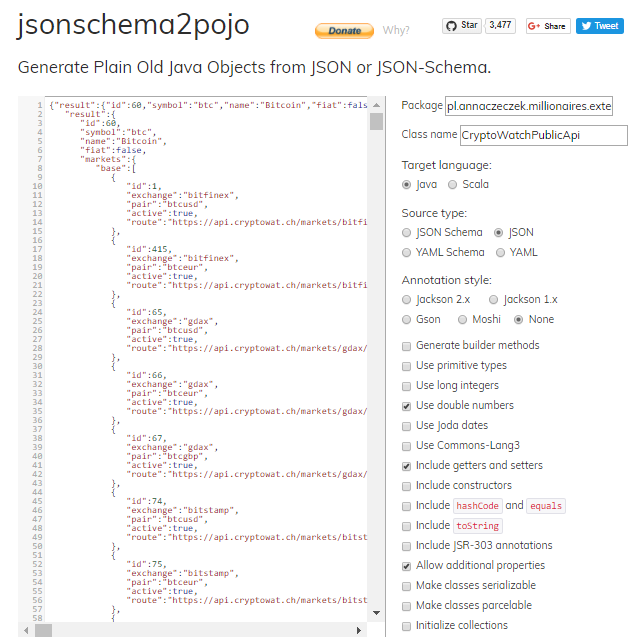
<https://www.bitmarket.pl/docs.php?file=api_public.html>

* 1. **BitFinex**

http://fixer.io/

1. **Zmapowanie obiektu JSON na klasę napisaną w języku Java przy użyciu narzędzia jsonschema2pojo**

Narzędzie dostępne pod adresem: <http://www.jsonschema2pojo.org/>



1. **Zapis danych w bazie InfluxDB, indeksowanie oraz próbkowanie danych, polityka retencji, konsolidacja danych**

InfluxDB jest metryczną bazą danych serii czasowych służącą do przechowywania danych analitycznych. Posiada ona kluczowe właściwości architektoniczne, czym znacząco różni się od tradycyjnych baz danych. Należą do nich: przechowywanie i kompresja znaczników czasu, zarzadzanie cyklem życia danych, podsumowanie danych, możliwość obsługi dużej ilości rekordów, a także tworzenie zapytań dotyczących danych pochodzących z serii czasowych. InfluxDB jest przeznaczony do monitorowania rozwoju, eksploatacji i zapewnienia jakości aplikacji (DevOps), przechowywania metryk, gromadzenia danych pochodzących z czujników oraz wykonywania analizy w czasie rzeczywistym. Baza zapewnia język zapytań posiadający funkcje wbudowane, koncentrujące się na minimalizacji czasu wykonywania zapytań dla struktury danych składającej się z pomiarów, serii i punktów. Każdy punkt składa się z kilku par klucz-wartość nazwanych tagami i jest opatrzony znacznikiem czasu. Wartości mogą być 64-bitowymi liczbami całkowitymi, zmiennoprzecinkowymi, łańcuchami i zmiennymi logicznymi. InfluxDB udostępnia cały zestaw poleceń administracyjnych służących do zarzadzania polityka utrzymywania

danych. Określa ona jak długo dane są przechowywane w bazie. Ponadto istnieje także możliwość ciągłego bądź okresowego odpytywania bazy danych przy użyciu automatycznych zapytań (ang. Continuous Queries) i zapisywania wyników pomiarów w pamięci.

Poniżej zamieszczono fragment kodu pozwalającego na zapis danych w InfluxDB przy użyciu znaczników czasu, które wraz z tagami stanowią unikalne identyfikatory rekordów w postaci klucz- wartość.

**public void save(ExchangeRate exchangeRate) {**

Point point = Point.*measurement*("exchangeRates")

.time(System.*currentTimeMillis*(), TimeUnit.*MILLISECONDS*)

.tag("exchange", exchangeRate.getExchange())

.tag("cryptoCurrency", exchangeRate.getCryptoCurrency())

.tag("currency", exchangeRate.getCurrency())

.addField("exchangeRate", exchangeRate.getExchangeRate())

.build();

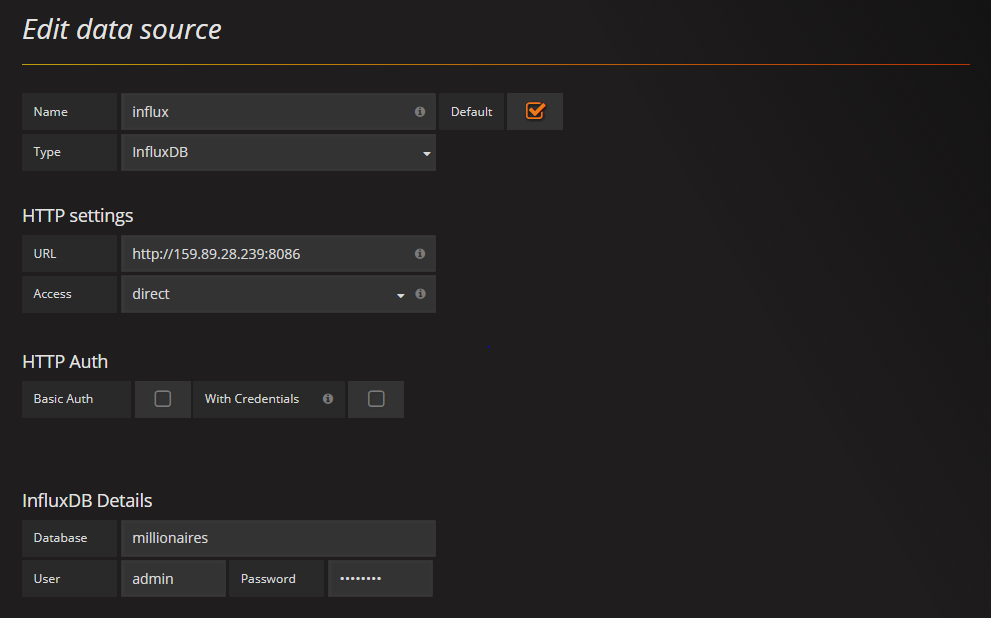
*logger*.info("Saving exchange rate : " + exchangeRate);

influxDB.write(point);

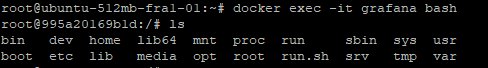
}

1. **Konfiguracja narzędzia Grafana**

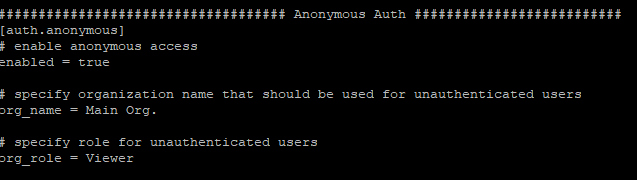
Gdy baza danych InfluxDB została stworzona, można ją podpiąć jako źródło danych w Grafanie, pozwalającej na wizualizację zapisanych danych. W tym celu należy podać nazwę bazy danych, login i hasło użytkownika, a także ustawić adres IP bazy danych.



Aby wybrany wykres, który został wygenerowany przy użycia narzędzia, a następnie osadzony w aplikacji webowej mógł być widoczny dla wszystkich użytkowników, również tych niezalogowanych, należy dokonać konfiguracji narzędzia przy pomocy programu Putty. W tym celu należy przejść do kontenera Grafany (dostępne kontenery można wyświetlić przy użyciu komendy $ docker ps).

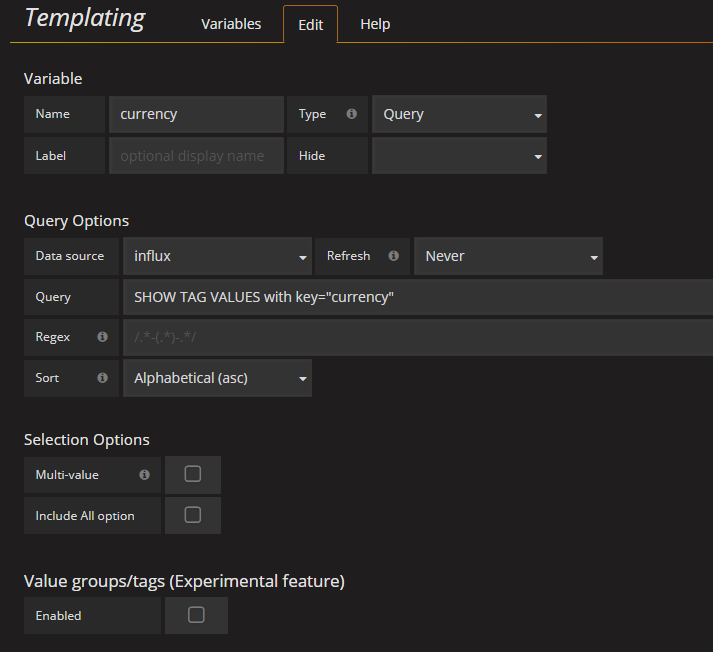


Następnie należy przejść do lokalizacji /etc/grafana i dokonać zmian w pliku konfiguracyjnym grafana.ini. W celu zezwolenia na anonimową autentykację należy zdefiniować organizację, rolę niezalogowanego użytkownika w organizacji, a także zezwolić na anonimowy dostęp.

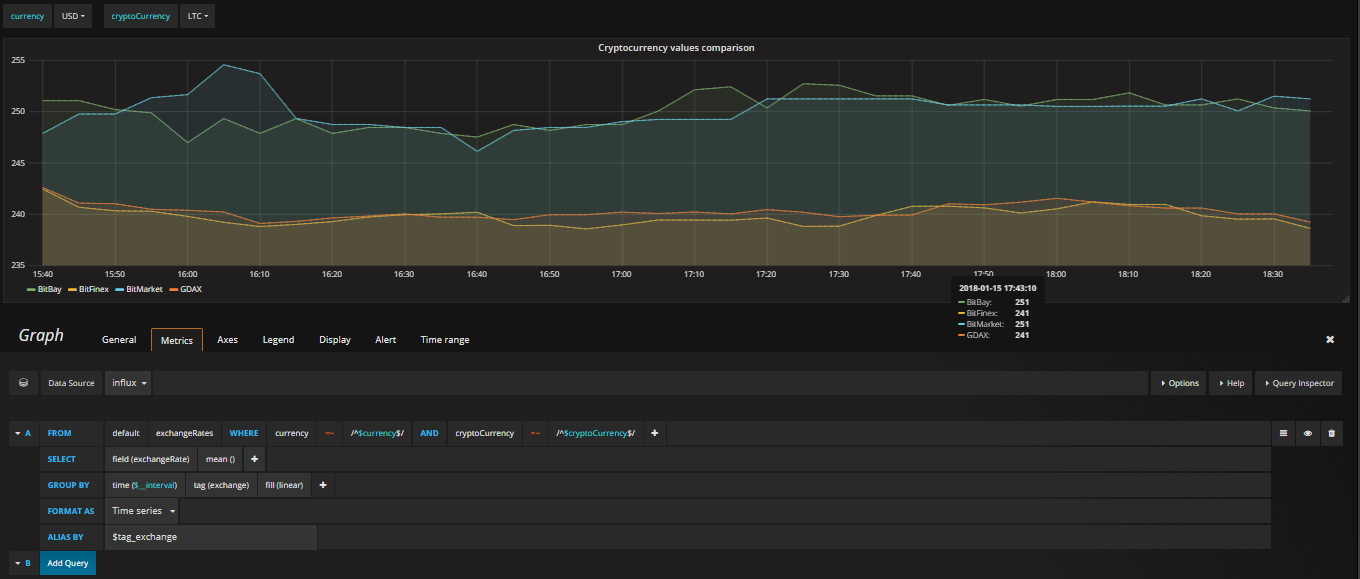


1. **Tworzenie paneli oraz wykresów wraz z użyciem mechanizmu alertowania**

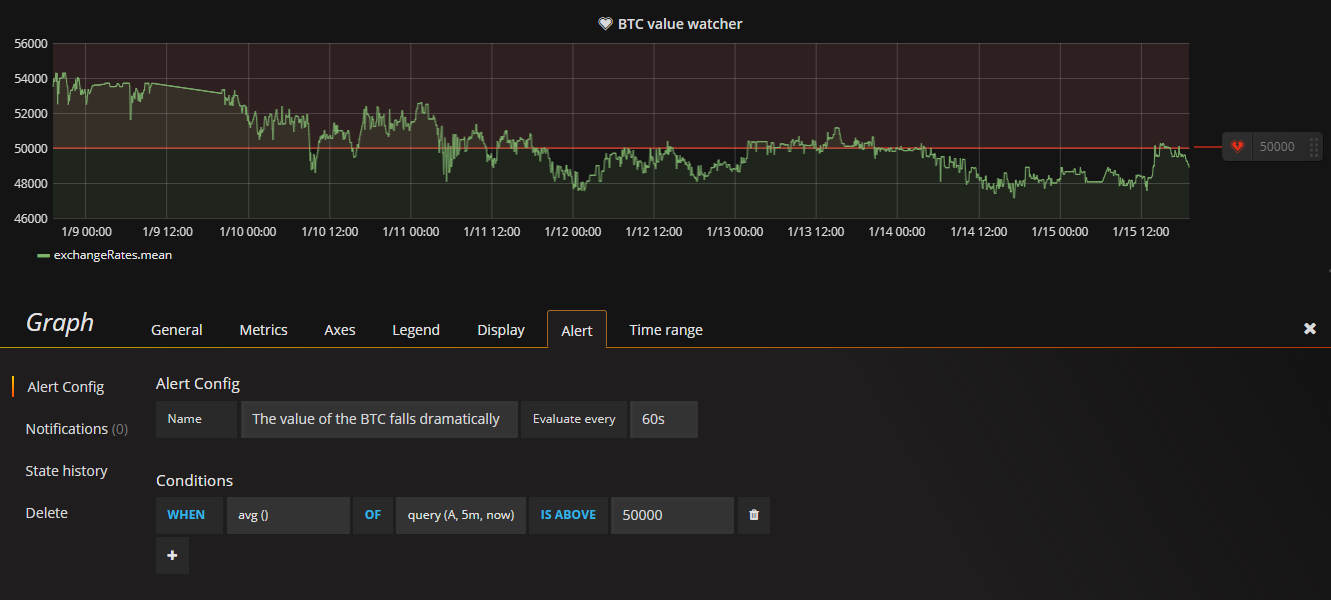
**10.1 Definiowanie zmiennych szablonowych**



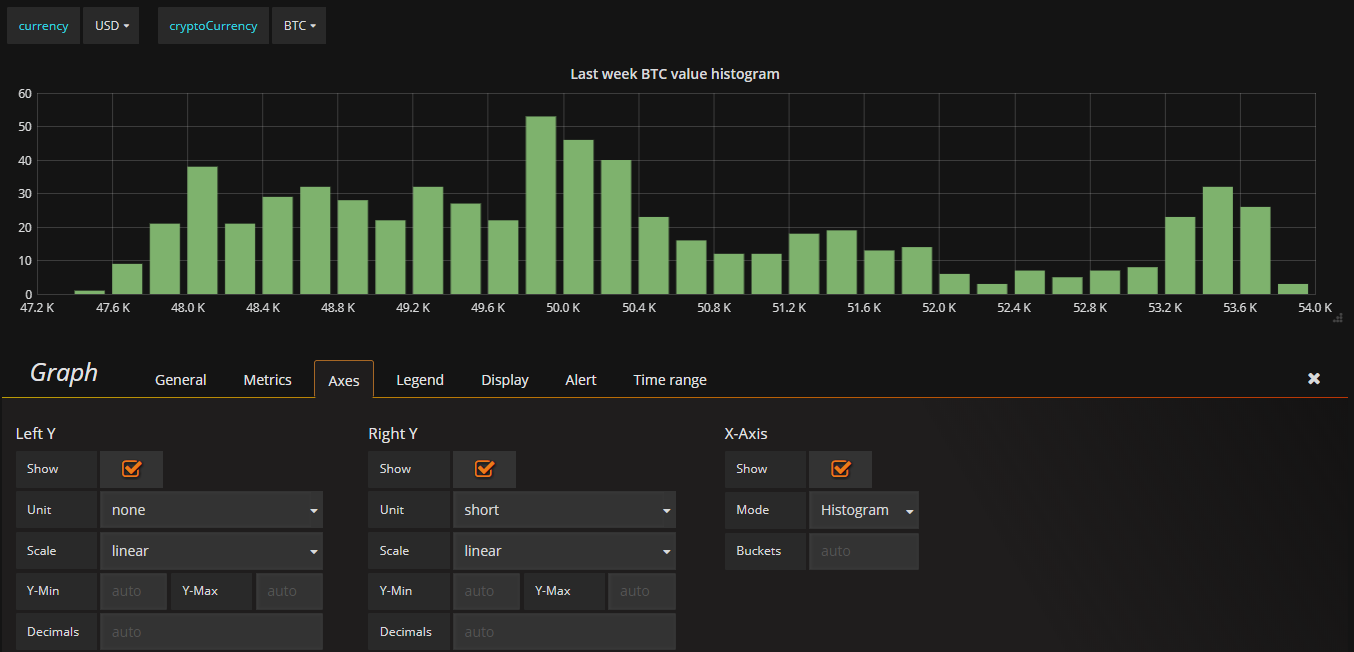
**10.2 Tworzenie wykresu przy użyciu zmiennych szablonowych**



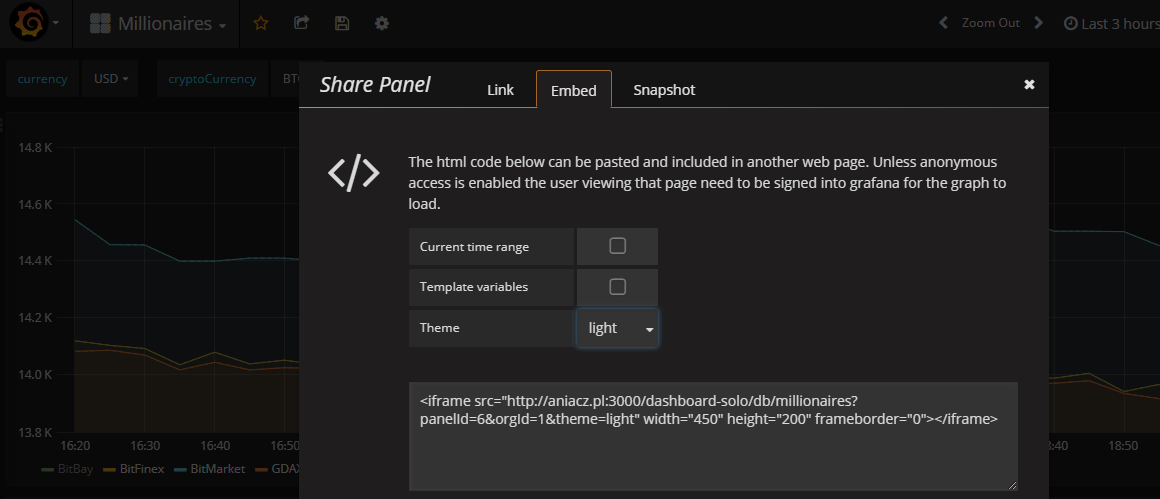
**10.3 Zastosowanie mechanizmu powiadomień**

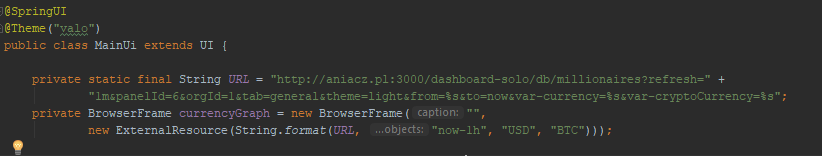


**10.4 Utworzenie histogramu**

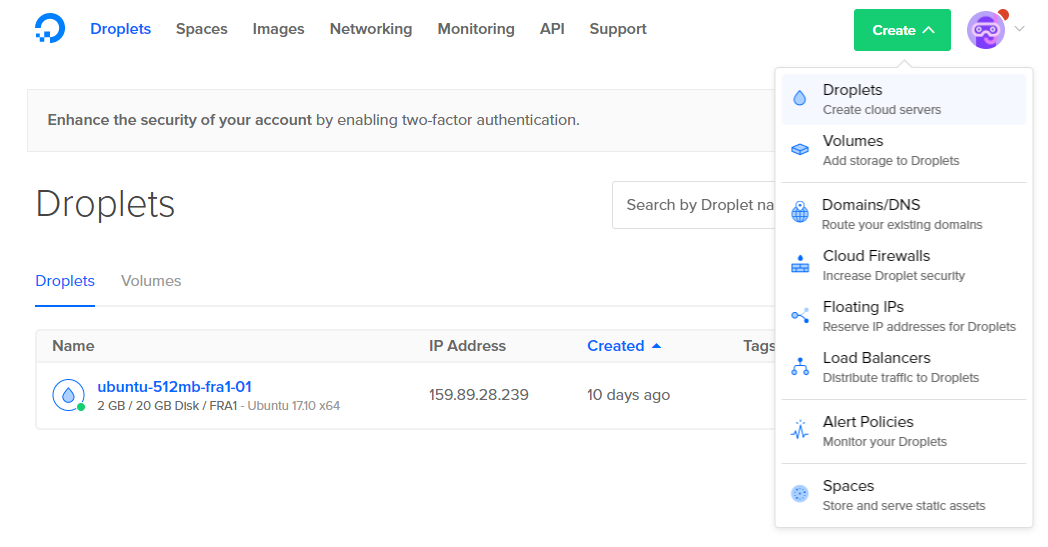


1. **Osadzenie wykresu w aplikacji webowej**

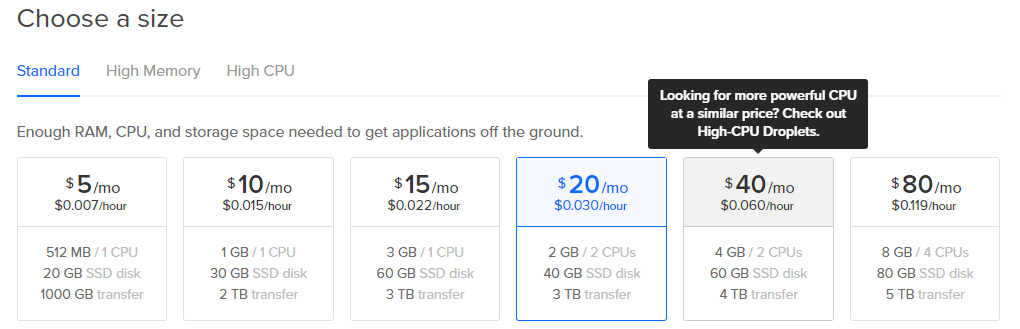




**12.Konfiguracja VPS**

****

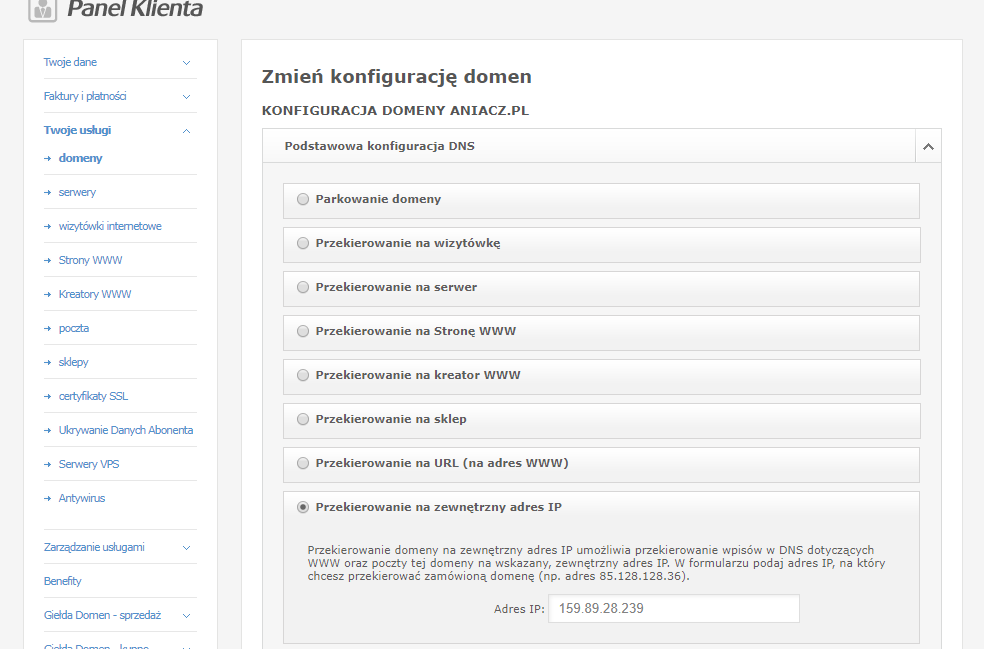
**12.1 Wybór systemu operacyjnego – Ubuntu**

**12.2 Dobranie odpowiedniej ilości pamięci RAM, procesora oraz przestrzeni dyskowej**  


**12.3 Lokalizacja serwera**

**12.4 Dodanie klucza SSH**

**13.Konfiguracja domeny aniacz.pl w serwisie nazwa.pl**



**14. Instalacja aplikacji**

W celu zbudowania aplikacji należy użyć polecenia

|  |
| --- |
| mvn clean install |

dostępnego w wybranym narzędziu budującym.

Następnie należy skopiować zbudowaną paczkę z rozszerzeniem .jar na serwer i uruchomić ją:

|  |
| --- |
| nohup java –jar millionaires.jar & |

Należy także pamiętać, że przed uruchomieniem aplikacji najpierw należy wystartować bazę InfluxDB oraz narzędzie Grafana.