

Руководство пользователя по установке программного комплекса «Оператор электронной торговли»

Для использования WireGuard необходимо ПК, требования минимальны, так как это легковесный и производительный VPN-протокол. Вот основные требования:

Минимальные системные требования:

1. Операционная система: Windows 7, 8, 8.1, 10 или 11 (32-bit и 64-bit).
2. Процессор: Любой современный x86 или x64 процессор (поддерживаются и ARM-процессоры для специфичных устройств).
3. Оперативная память: Минимум 512 МБ (рекомендуется 1 ГБ и выше для стабильной работы при высоких нагрузках).
4. Свободное место на диске: Всего около 20-30 МБ для самого приложения и конфигураций.
5. Сетевое подключение: Требуется активное интернет-соединение для работы VPN.

WireGuard оптимизирован для работы на различных платформах и не требует мощного оборудования.

1. Загрузка приложения

Убедитесь, что на вашем сервере или в облачной инфраструктуре установлены следующие компоненты:

- Kubernetes (k8s) версии 1.18 или выше.
- Apache Kafka версии 2.8.0 или выше.
- PostgreSQL версии 13.0 или выше.
- MongoDB версии 4.4 или выше.

Установите Helm для управления чартами Kubernetes:

```
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
```

Загрузите необходимые Docker образы для каждого микросервиса из репозитория:

```
docker pull yourregistry/your-microservice:latest
```

Минимальные требования:

- 3 сервера с характеристиками:
 - 40 ГБ оперативной памяти на каждую ноду.
 - 4 ядра ЦПУ на каждую ноду.

- Рекомендуемое дисковое пространство: 500 ГБ для каждой ноды (в зависимости от объема данных и нагрузки).
- Операционная система: Linux
- Docker

2. Запуск приложения

Запуск PostgreSQL и MongoDB:

Разверните PostgreSQL и MongoDB через Helm чарт:
helm repo add bitnami https://charts.bitnami.com/bitnami
helm install my-postgres bitnami/postgresql
helm install my-mongodb bitnami/mongodb

Настройка Kafka:

Запустите Kafka с помощью Helm:
helm repo add bitnami https://charts.bitnami.com/bitnami
helm install my-kafka bitnami/kafka

Запуск микросервисов:

Разверните каждый микросервис в Kubernetes:
kubectl apply -f your-microservice-deployment.yaml

Настройка сетевых взаимодействий и секретов:

Создайте ConfigMap и Secret для хранения переменных окружения и конфиденциальных данных:
kubectl apply -f your-configmap.yaml
kubectl apply -f your-secret.yaml

После выполнения указанных выше шагов, вам необходимо обратиться в тех.поддержку и мы вам установим актуальную версию конфигурации (дистрибутива), которая разворачивает всю необходимую инфраструктуру автоматически.

Процесс выглядит следующим образом: вы получите дистрибутив, который включает в себя всё необходимое для развёртывания микросервисной архитектуры. Этот дистрибутив упакован в виде конфигураций для Kubernetes. Вам нужно будет лишь выполнить команду **kubectl apply -f .** и система сама позаботится о развёртывании всех компонентов.

3. Выполнение приложения

После подключения маркетплейсов и партнеров, которые обслуживают процесс доставки, будет проходить процесс обмена и управления информацией через дашборд: (панель управления)

4. Остановка приложения

Остановите микросервисы:

```
kubectl delete -f your-microservice-deployment.yaml
```

Остановите Kafka, PostgreSQL, и MongoDB:

```
helm uninstall my-kafka
```

```
helm uninstall my-postgres
```

```
helm uninstall my-mongodb
```

5. Удаление приложения

Удалите все связанные ресурсы в Kubernetes:

```
kubectl delete namespace your-app-namespace
```

Очистите данные из базы данных (при необходимости):

```
kubectl exec -it my-postgres-pod -- psql -U postgres -c "DROP DATABASE your_database;"
```

```
kubectl exec -it my-mongodb-pod -- mongo --eval "db.dropDatabase();"
```

6. Решение типовых проблем

Проблемы с подключением к Kafka:

Убедитесь, что Kafka запущена и доступна: `kubectl get pods | grep kafka`

Проверьте сетевые настройки и переменные окружения микросервисов, которые используют Kafka.

Не удается подключиться к PostgreSQL:

Проверьте запущен ли PostgreSQL и доступны ли креденшлы:

```
kubectl get pods | grep postgres
```

```
kubectl exec -it my-postgres-pod -- psql -U postgres -c "\l"
```

MongoDB отказывается принимать соединения:

Убедитесь, что база данных запущена и правильно настроены переменные окружения:

```
kubectl logs my-mongodb-pod
```

Проблемы с балансировщиком Kubernetes:

Проверьте статус сервисов и их доступность:

```
kubectl get services
```

7. Способы обращения в техподдержку

График работы техподдержки:
Понедельник - Пятница, с 9:00 до 18:00 (по мск).

Способы связи:

Электронная почта: support@unifintech.ru

Контактные данные:

Чугринов Станислав:

grin@unitrade.su.

Дурнов Кирилл

durnov@unitrade.su

Это руководство должно помочь вам правильно развернуть и управлять программным комплексом «Оператор электронной торговли».