

Руководство по установке МоиОтчеты Корпоративный сервер и МоиОтчеты Публикатор

быстрыеотчеты.рф 1 / 76

# Разворачивание кластера Kubernetes

Продукты: МоиОтчеты Корпоративный Сервер

Для установки и настройки сервера отчётов необходимо развернуть облачную экосистему Kubernetes.

Для корректной работы МоиОтчеты Корпоративный Сервер необходимо минимум три узла - один master узел и два worker узла. В процессе работ количество узлов может быть увеличено при росте нагрузки. Так же количество узлов может быть уменьшено при снижении нагрузки. Динамическое управление количеством узлов в настоящее время не реализовано.

Видео установки Корпоративного Сервера

## Подготовка и установка компонентов Kubernetes

Установка Kubernetes осуществляется последовательным выполнением следующих bash скриптов. Эти примеры подразумевают операционную систему Debian Linux. В случае использования другого дистрибутива некоторые пункты будут отличаться. Например, для разворачивания кластера в системе Alt Linux Server, использовалась документация со страницы www.altlinux.org/Kubernetes, которой будет достаточно для установки Kubernetes, описанной в этой разделе.

1. Отключение раздела swap.

```
#
# Permanently disable swap
#
sed -e '/swap/s/^/#/g' -i /etc/fstab
swapoff -a
```

2. Загрузка необходимых модулей ядра.

3. Настройка сетевых свойств ядра системы.

```
#
# Configure kernel
#
SYSCTL=/etc/sysctl.d/k8s.conf
if [ ! -f $SYSCTL ]; then
    echo "Prepare kernel options"
    cat<<EOF | tee $SYSCTL
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
net.bridge.bridge-nf-call-ip6tables = 1
EOF
fi
sysctl --system
```

быстрыеотчеты.рф 2 / 76

4. Установка дополнительного программного обеспечения, которое потребуется для установки оркестратора и сервера отчётов.

```
apt-get update

#
# Install pre-requested packages
#
prerequest=( curl sudo gnupg2 apt-transport-https ca-certificates software-properties-common )

for package in "${prerequest[@]}"

do
    echo -n "Checking $package: "
    dpkg -s $package > /dev/null 2> /dev/null
    if [ $? -ne 0 ]; then
        echo "Installing"
        apt-get install -y $package
    else
        echo "OK"
    fi
done
```

5. Установка Container Runtime Interface (CRI).

Следующий пример использует CRI containerd. Различные версии Kubernetes и Linux могут потребовать установки CRI-O. Kubernetes использует CRI для загрузки контейнеров, управления контейнерами и для запуска процессов в этих контейнерах.

```
## Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key --keyring /etc/apt/trusted.gpg.d/docker.gpg add -
## Add Docker apt repository.
add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/debian \
  $(lsb_release -cs) \
  stable"
## Install containerd
apt-get update && apt-get install -y containerd.io
# Configure containerd
if [ ! -d /etc/containerd ]; then
mkdir -p /etc/containerd
fi
# Remove default config to avoid errors
if [ -f /etc/containerd/config.toml ]; then
rm /etc/containerd/config.toml
fi
# Restart containerd
systemctl restart containerd
```

6. Установка компонентов Kubernetes.

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
cat <<EOF | tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
apt-get update
apt-get install -y kubelet kubeadm kubectl
apt-mark hold kubelet kubeadm kubectl
```

быстрыеотчеты.рф 3 / 76

#### **Старт Kubernetes**

Для старта главного узла кластера отредактируйте 3 переменные, отвечающие за соответствующие им пути к конфигурационному файлу (он будет создан автоматически), IP адресу (будет доступен извне) и маску внутренней подсети.

Затем выполните команду инициализации кластера.

```
export KUBECONFIG=/etc/kubernetes/admin.conf1
export MAIN_IF=192.168.1.191
export POD_NETWORK=10.244.0.0/16 # В случае использования flannel это значение менять нельзя!
kubeadm init --pod-network-cidr=$POD_NETWORK --apiserver-advertise-address=$MAIN_IF
```

Адрес MAIN\_IF это IP-адрес главного узла кластера (master node). Адрес может быть реальным IP-адресом или адресом подсети 192.168.

В случае успешного выполнения инициализации главного узла кластера (master node), на экран будет выведена строка, с помощью которой инициализируются рабочие узлы кластера. С помощью мыши скопируйте эту строку и сохраните её в файл. В этой строке содержится команда для инициализации рабочих узлов. Команда включает в себя секретный ключ. При попадании этого ключа злоумышленнику, он может "скомпрометировать" кластер.

Пример строки:

```
kubeadm join 192.168.1.191:6443 --token lw0lgz.d5zy9fb4jikc89yv \
--discovery-token-ca-cert-hash sha256:ba83ed75e9fd5f430007000000000000039e1d05c7915a54435faaa7fe62b77
```

Для добавления узла в кластер выполните действия, описанные в предыдущем разделе (за исключением установки helm chart), скопируйте на каждый узел эту строку и выполните её от имени администратора (root). В результате выполнения этой команды в кластер будет добавлен новый узел.

Hастройка kubectl.

```
mkdir ~/.kube
rm ~/.kube/config
cp /etc/kubernetes/admin.conf ~/.kube
```

Теперь необходимо установить сервис flannel.

```
kubectl apply -f https://raw.githubusercontent.com/flannel-io/flannel/master/Documentation/kube-flannel.yml
```

В случае успешной установки вывод команды kubectl get pods --all-namespaces должен выглядеть примерно так:

```
NAMESPACE NAME
                                    READY STATUS RESTARTS AGE
kube-system coredns-78fcd69978-brp76
                                          1/1
                                                              5m13s
                                                Running 0
kube-system coredns-78fcd69978-shdv5
                                           1/1
                                                Running 0
                                                               5m13s
                                      1/1 Running 0
kube-system etcd-altlinux-10-1
                                                         5m27s
kube-system kube-apiserver-altlinux-10-1
                                         1/1 Running 0
                                                             5m22s
kube-system kube-controller-manager-altlinux-10-1 1/1
                                                  Running 0
                                                                 5m31s
kube-system kube-flannel-ds-7nn2c
                                       1/1 Running 0
                                                            28s
kube-system kube-flannel-ds-9phqs
                                        1/1
                                             Running 0
                                                            28s
kube-system kube-flannel-ds-ddcw6
                                       1/1
                                              Running 0
                                                            28s
kube-system kube-proxy-74pll
                                      1/1
                                           Running 0
                                                          3m36s
                                                           3m15s
kube-system kube-proxy-82nld
                                      1/1
                                            Running 0
                                                           5m14s
kube-system kube-proxy-njxwv
                                       1/1 Running 0
                                                              5m32s
kube-system kube-scheduler-altlinux-10-1
                                          1/1 Running 0
```

быстрыеотчеты.рф 4 / 76



быстрыеотчеты.рф 5 / 76

# МоиОтчеты Корпоративный Сервер и импортозамещение

Продукты: МоиОтчеты Корпоративный Сервер

#### Версии компонентов Kubernetes

Изначально продукт разрабатывался для кластера, развёрнутого на основе дистрибутива Debian GNU/Linux 10 (buster). Для поддержки импортозамещения был протестирован ряд дистрибутивов Linux российских производителей. По итогам тестирования зависимости от версии Kubernetes не выявлено - минимальные номера версий, на которых производилось тестирование показаны в следующей таблице:

компонент	ВЕРСИЯ
docker	19.3.8
kubelet	1.18.1
kube-proxy	1.18.1

При тестировании проверялись также более свежие версии Kubernetes. Обновление версии Kubernetes не нарушает работу сервера отчётов. Таким образом тестирование отечественных дистрибутивов свелось к проверке возможности установки Kubernetes.

### Провайдеры CNI

Сервер отчётов был протестирован со CNI провайдерами Flannel и Calico. Оба провайдера показали удовлетворительную работу. Конфигурация с Flannel была протестирована на самостоятельно развёрнутом кластере, конфигурация с Calico была протестирована на облачном провайдере Selectel (Россия).

## Российские дистрибутивы Linux

Результаты тестирования представлены в следующей таблице:

ПРОИЗВОДИТЕЛЬ	СОВМЕСТИМОСТЬ	принарамичп
Alt Linux Server 9	Полная	https://www.altlinux.org/Kubernetes
ROSA "Хром" Linux	Заявлена	http://wiki.rosalab.com/ru/index.php/Установка_docker,_kebernates_кластера
ASTRA Linux 1.7	Deckhouse	https://deckhouse.io/ru/
RED OS	Заявлена	https://redos.red-soft.ru/base/server-configuring/container/kubernetes/
AlterOS 7	Deckhouse	https://deckhouse.io/ru/

Примечание: Deckhouse - продукт российских разработчиков для разворачивания кластера.

быстрыеотчеты.рф 6 / 76

# Установка сервиса nginx-ingress

Продукты: МоиОтчеты Корпоративный Сервер

#### Все описанные команды выполняется только на master узле!

Добавление в Helm репозитория ingress-nginx:

helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx helm repo update

### Установка nginx в Kubernetes

Внимание! Переменная MAIN\_IF должна указывать IP адрес внешнего интерфейса для доступа к серверу отчётов.

export MAIN\_IF=192.168.1.191 kubectl create namespace nginx

helm upgrade --install nginx-ingress ingress-nginx/ingress-nginx \

- --create-namespace --namespace nginx \
- --set controller.replicaCount=2 \
- --set controller.service.externalIPs[0]=\$MAIN\_IF \
- --set controller.extraArgs.v=2

Следующая команда конфигурирует nginx в кластере. nginx будет принимать все входящие запросы и перенаправлять их в gateway, который распределяет входящие запросы между компонентами сервера отчётов.

быстрыеотчеты.рф 7 / 76

```
HOST=my.server-server.ru
NAMESPACE=fr-corporate-server
cat <<EOF | kubectl apply -n $NAMESPACE -f -
kind: Ingress
apiVersion: networking.k8s.io/v1
metadata:
name: $HOST-gateway
 namespace: $NAMESPACE
 annotations:
  ingress.kubernetes.io/ssl-redirect: 'true'
  nginx.ingress.kubernetes.io/configuration-snippet: |
   add header X-Robots-Tag "noindex, nofollow, nosnippet, noarchive";
  nginx.ingress.kubernetes.io/limit-rps: '50'
  nginx.ingress.kubernetes.io/proxy-body-size: '0'
  nginx.ingress.kubernetes.io/proxy-buffering: 'off'
  nginx.ingress.kubernetes.io/proxy-request-buffering: 'off'
spec:
 ingressClassName: nginx
tls:
  - hosts:
    - $HOST
   secretName: corporate-tls-secret
 rules:
  - host: $HOST
   http:
    paths:
       pathType: ImplementationSpecific
       backend:
        service:
         name: fr-gateway
         port:
          number: 80
EOF
```

Далее понадобится SSL сертификат для настройки защищённого соединения. Обычно его можно купить/получить у регистратора доменного имени или купить у удостоверяющего центра. При использовании сервера отчётов в интранете можно создать самостоятельно подписанный сертификат с помощью следующей команды:

```
export CERT_NAME=my.server-server.ru
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout del_me_file.key -out del_me_file.cer -subj
"/CN=$CERT_NAME/O=$CERT_NAME"
```

Регистрация сертификата в Kubernetes с именем fr-corporate-tls. Далее этот сертификат используется различными компонентами отчёта, в том числе nginx.

```
kubectl create secret tls fr-corporate-tls --key del_me_file.key --cert del_me_file.cer
```

Регистрация nginx-ingress в кластере Kubernetes.

Необходимо установить переменную HOST, соответствующую доменному имени сервера отчётов.

быстрыеотчеты.рф 8 / 76

```
cat <<EOF | kubectl apply -n $NAMESPACE -f -
kind: Ingress
apiVersion: networking.k8s.io/v1
metadata:
name: $HOST-gateway
 namespace: $NAMESPACE
 annotations:
  ingress.kubernetes.io/ssl-redirect: 'true'
  nginx.ingress.kubernetes.io/configuration-snippet: |
   add_header X-Robots-Tag "noindex, nofollow, nosnippet, noarchive";
  nginx.ingress.kubernetes.io/limit-rps: '50'
  nginx.ingress.kubernetes.io/proxy-body-size: '0'
  nginx.ingress.kubernetes.io/proxy-buffering: 'off'
  nginx.ingress.kubernetes.io/proxy-request-buffering: 'off'
spec:
 ingressClassName: nginx
 tls:
  - hosts:
     - $HOST
   secretName: corporate-tls-secret
  - host: $HOST
   http:
    paths:
      - path: /
       pathType: ImplementationSpecific
       backend:
        service:
         name: fr-gateway
         port:
           number: 80
EOF
```

быстрыеотчеты.рф 9 / 76

# Лог установки helm и пакетов №2

Продукты: МоиОтчеты Корпоративный Сервер

Kubernetes-Dashboard это WEB-интерфейс для управления кластером. Он позволяет с помощью WEBбраузера управлять Kubernetes. Dashboard не является необходимым компонентом для работы сервера отчётов, однако он позволяет упростить настройку и конфигурирование сервера, а также мониторить состояние узлов и контейнеров.

#### Установка Dashboard

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-rc7/aio/deploy/recommended.yaml

#### Конфигурирование Dashboard

При конфигурации Dashboard используется переменная SERVER\_DOMAIN\_NAME . Обратите внимание, нижеприведённая конфигурация использует Dashboard на том же самом IP адресе, что и сервер отчётов. Для захода на страницу Dashboard используется правило, определяющее путь к Dashboard.

```
export SERVER DOMAIN NAME="my.server-server.ru"
cat <<EOF | kubectl apply -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: fastreport.cloud-dashboard
 namespace: kubernetes-dashboard
 annotations:
  kubernetes.io/ingress.class: "nginx"
  nginx.ingress.kubernetes.io/backend-protocol: HTTPS
  nginx.ingress.kubernetes.io/use-regex: "true"
  nginx.ingress.kubernetes.io/rewrite-target: "/\$2"
  nginx.ingress.kubernetes.io/configuration-snippet: |
   rewrite ^(/dashboard)\$ \$1/ redirect;
spec:
tls.
 - hosts:
  - $SERVER DOMAIN NAME
  secretName: fr-corporate-tls
 rules:
 - host: $SERVER_DOMAIN_NAME
  http:
   paths:
   - path: /dashboard(/|$)(.*)
     serviceName: kubernetes-dashboard
     servicePort: 443
EOF
```

Проверить созданную конфигурацию nginx сервера можно с помощью следующих команд, используя вместо nginx-ingress-controller-6674ff5868-t47xk имя созданного nginx контейнера:

```
kubectl exec -it nginx-ingress-controller-6674ff5868-t47xk -n nginx -- ls /etc/nginx/
kubectl exec -it nginx-ingress-controller-6674ff5868-t47xk -n nginx -- cat /etc/nginx/nginx.conf
```

Далее необходимо получить token, который будет использоваться для административного доступа к Dashboard. Создание административной учётной записи:

быстрыеотчеты.рф 10 / 76

cat <<EOF | kubectl apply -f -

apiVersion: v1

kind: ServiceAccount

metadata:

name: dashboard-admin-user namespace: kube-system

**EOF** 

Назначение прав доступа административной учётной записи:

cat <<EOF | kubectl apply -f -

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: dashboard-admin-user

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole name: cluster-admin

subjects:

- kind: ServiceAccount

name: dashboard-admin-user namespace: kube-system

**EOF** 

#### Создание токена:

kubectl -n kubernetes-dashboard create token dashboard-admin-user -n kube-system

Распаковка и вывод на экран токена, использующегося для авторизации на Dashboard:

kubectl -n kube-system describe secret \$(kubectl -n kube-system get secret | grep dashboard-admin-user | awk '{print \$1}')

Сохраните токен для последующего использования в недоступном для посторонних месте и используйте его для авторизации в Dashboard.

Пример URL для доступа к dashboard: https://my.server-server.ru/dashboard/

быстрыеотчеты.рф 11 / 76

# Лог установки helm и пакетов №3

Продукты: МоиОтчеты Корпоративный Сервер

Grafana это компонент аналитики и интерактивной визуализации для мониторинга состояния кластера. Это необязательный компонент для разворачивания сервера отчётов, но он может быть полезен для контроля состояния сервера.

Loki это агрегатор логов, который используется совместно с Grafana.

# Создание пространства имён для компонентов мониторинга

kubectl create namespace monitoring

# Обновление чартов для установки компонентов мониторинга

helm repo add loki https://grafana.github.io/loki/charts helm repo update

#### Установка Loki

helm upgrade \

- --install loki loki/loki \
- --namespace=monitoring \
- --set persistence.enabled=true \
- $\hbox{--set persistence.} storage Class Name = hcloud-volumes$

helm upgrade \

- --install promtail loki/promtail \
- --namespace=monitoring \
- --set loki.serviceName=loki.monitoring

#### Установка Grafana

Не забудьте установить переменную SERVER\_DOMAIN\_NAME.

export SERVER\_DOMAIN\_NAME="<доменное имя сервера>"

helm upgrade \

- --install grafana stable/grafana \
- --namespace=monitoring \
- --set persistence.enabled=true \
- $\hbox{--set persistence.storageClassName=$hcloud-volumes $$\$
- --set 'grafana\.ini'.server.serve\_from\_sub\_path=true \
- --set 'grafana\.ini'.server.root\_url=https://\$SERVER\_DOMAIN\_NAME/grafana/\
- --set 'grafana\.ini'.server.domain=\$SERVER\_DOMAIN\_NAME \
- --set readinessProbe.httpGet.path=/grafana/api/health \
- --set livenessProbe.httpGet.path=/grafana/api/health \

### Получение пароля от Grafana

быстрыеотчеты.рф 12 / 76

```
kubectl get secret --namespace monitoring grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo admin *********
```

# Добавление ingress

Этот блок конфигурации позволяет расположить Grafana на том же самом сетевом интерфейсе, что и сервер отчётов и обращаться к ней по ссылке вида: https://my.server-server.ru/grafana

```
cat <<EOF | kubectl apply -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: fastreport.cloud-grafana
 namespace: monitoring
 annotations:
  kubernetes.io/ingress.class: "nginx"
  ingress.kubernetes.io/ssl-redirect: "true"
spec:
 tls:
 - hosts:
  - $SERVER_DOMAIN_NAME
  secretName: fr-corporate-tls
 - host: $SERVER_DOMAIN_NAME
  http:
   paths:
   - path: /grafana
    backend:
      serviceName: grafana
      servicePort: 80
EOF
```

Для завершения конфигурации Grafana необходимо зайти на её страницу, авторизоваться и добавить ссылку на Loki http://loki.monitoring:3100

быстрыеотчеты.рф 13 / 76

# Установка брокера сообщений RabbitMQ

Продукты: МоиОтчеты Корпоративный Сервер

Для установки RabbitMQ необходимо определить переменную RABBITMQ\_PASSWORD, содержащую пароль. Также можно настроить размер тома для очереди брокера сообщений.

Добавление чарта и обновление helm репозитория.

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm repo update
```

Создать пространство имён для RabbitMQ.

```
kubectl create namespace rabbitmq
```

Установка RabbitMQ может варьироваться от типа установки. Ниже приведены два примера - для облачного провайдера и для установки на тестовой конфигурации.

#### Установка на облачном хостинге

Нижеприведённый скрипт использует размер очереди 10 Гб.

```
export RABBITMQ_PASSWORD="<пароль для RabbitMQ>"
cat <<EOF > ./rabbitmq.yaml
rabbitmq:
extraConfiguration: |-
  #disk_free_limit.absolute = 10GB
  management.path_prefix = /rabbit
  #management.load_definitions = /app/load_definition.json
persistence:
 enabled: true
 storageClass: hcloud-volumes
size: 10Gi
livenessProbe:
 commandOverride:
  - sh
  - -C
  - rabbitmq-api-check "http://user:\$RABBITMQ_PASSWORD@127.0.0.1:15672/rabbit/api/healthchecks/node"
'{"status":"ok"}'
readinessProbe:
commandOverride:
  - sh
  - rabbitmq-api-check "http://user:\$RABBITMQ PASSWORD@127.0.0.1:15672/rabbit/api/healthchecks/node"
'{"status":"ok"}'
EOF
helm upgrade \
--install rabbitmq bitnami/rabbitmq \
--namespace=rabbitmq \
--values ./rabbitmq.yaml
rm ./rabbitmq.yaml
```

# Пример минимальной установки RabbitMQ для разворачивания на тестовом кластере

быстрыеотчеты.рф 14 / 76

Нижеприведённая конфигурация использует размер очереди 4 Гб.

Обратите внимание - данный пример является минимальным для запуска МоиОтчеты Корпоративный Сервер и описанная конфигурация томов не является отказоустойчивой.

```
apiVersion: v1
kind: PersistentVolume
metadata:
namespace: "rabbitmq"
name: pv-for-rmq
labels:
  name: mongo-volume-1-0-0
spec:
storageClassName: manual
capacity:
 storage: 4Gi
 accessModes:
- ReadWriteOnce
hostPath:
  path: /devkube/rabbitmq
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: "rabbitmq-pvc"
namespace: "rabbitmq"
spec:
accessModes:
- ReadWriteOnce
storageClassName: manual
resources:
  requests:
   storage: 4Gi
```

kubectl apply -f <имя вышеприведённого Выполните yaml>

и собственно скрипт установки RabbitMQ.

Обратите внимание, что используется ранее созданный PersistentVolumeClaim.

```
helm upgrade \
--namespace=rabbitmq \
--install rabbitmq bitnami/rabbitmq \
--set persistence.existingClaim=rabbitmq-pvc \
--set volumePermissions.enabled=true \
--set ingress.tlsSecret="fr-corporate-tls" \
| tee RabitMQ.txt
```

## Добавление конфигурации ingress

Этот пункт необходим если вы хотите получить доступ к контрольной панели сервиса RabbitMQ извне кластера.

быстрыеотчеты.рф 15 / 76

```
export SERVER_DOMAIN_NAME="<доменное имя сервера>"
cat <<EOF | kubectl apply -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
name: fastreport.cloud-rabbitmq
 namespace: rabbitmq
 annotations:
  kubernetes.io/ingress.class: "nginx"
  nginx.ingress.kubernetes.io/rewrite-target: "/\$2"
  nginx.ingress.kubernetes.io/configuration-snippet: |
   rewrite ^(/rabbit)\$ \$1/ redirect;
spec:
tls:
 - hosts:
  - $SERVER_DOMAIN_NAME
  secretName: fr-corporate-tls
 rules:
 - host: $SERVER_DOMAIN_NAME
  http:
   paths:
   - path: /rabbit
    backend:
     serviceName: rabbitmq
     servicePort: 15672
EOF
```

Доступ по user \*\*\*\*\*\*\*.

быстрыеотчеты.рф 16 / 76

## Установка MongoDB в кластер

Продукты: МоиОтчеты Корпоративный Сервер

#### Хранилище данных с открытым исходным кодом

На данной странице рассказывается о варианте установки с использованием MongoDB. Альтернативой выступает FerretDB - аналог с открытым исходным ходом, который использует PostgreSQL как движок базы данных. Инструкцию по установке с использованием FerretDB можно прочитать по ссылке.

#### Начало установки

Добавление репозитория для helm:

helm repo add bitnami bitnami/mongodb helm repo update

Сервер отчётов использует MongoDB для хранения шаблонов отчётов, подготовленных отчётов и различных документов, экспортированных из подготовленных отчётов.

Подготовка пространства имён:

NAMESPACE=mongo MONGO\_USER=any\_name MONGO\_PASS=mongo\_password MONGO\_DB\_SIZE=10Gi MONGO\_SERVICE\_NAME=fr-mongo kubectl create namespace \$NAMESPACE

Обратите внимание на переменную MONGO\_SERVICE\_NAME - это имя в дальнейшем будет использоваться для обращения к MongoDB из сервера отчётов.

Дальнейшая настройка будет отличаться от того, используется ли хранилище облачного провайдера или локальное хранилище. Ниже приведено два примера. Используйте только один из них, или модифицируйте настройки для своего облачного провайдера.

## Настройка хранилища для облачного провайдера

helm install \$MONGO\_SERVICE\_NAME bitnami/mongodb \

- --namespace \$NAMESPACE \
- --set persistence.enabled=true \
- --set persistence.storageClass=hcloud-volumes \
- --set persistence.size=\$MONGO DB SIZE
- --set auth.username=\$MONGO\_USER \
- --set auth.password=\$MONGO PASS \

| tee MongoDB.txt

## Настройка локального хранилища

StorageClass предоставляет средства для администраторов для описания "классов" хранилищ, которые они предоставляют.

быстрыеотчеты.рф 17 / 76

```
MONGO_DATA_DIR=/devkube/mongodb
MONGO_DB=reports_db # Можно использовать любое имя
##
## Create StorageClass for MongoDB
##
cat <<EOF | kubectl apply -n $NAMESPACE -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: local-sc-mongodb
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
EOF
##
## Create Persistent volume for MongoDB
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: PersistentVolume
metadata:
name: mongo-storage
namespace: $NAMESPACE
labels:
 type: local
spec:
capacity:
 storage: $MONGO_DB_SIZE
 accessModes:
  - ReadWriteOnce
storageClassName: local-sc-mongodb
local:
 path: $MONGO DATA DIR
nodeAffinity:
 required:
   nodeSelectorTerms:
   - matchExpressions:
    - key: kubernetes.io/hostname
     operator: In
     values:
     - altlinux-10-2
EOF
```

Для установки MongoDB используйте следующий скрипт:

```
helm install $MONGO_SERVICE_NAME bitnami/mongodb \
--namespace $NAMESPACE \
--set persistence.storageClass=local-sc-mongodb \
--set auth.username=$MONGO_USER \
--set auth.password=$MONGO_PASS \
--set auth.database=$MONGO_DB | tee MongoDB.txt
```

Результатом выполнения этого скрипта, помимо установки MongoDB в кластер, будет создан файл, содержащий информацию о результате установки базы данных. Обратите внимание, что параметры, передаваемые скрипту, были описаны в самом начале этого документа.

Проверить корректность установки MongoDB проще всего при помощи Kubernetes Dashboard, установка которого была описана ранее.

быстрыеотчеты.рф 18 / 76

## Установка FerretDB в кластер

Продукты: МоиОтчеты Корпоративный Сервер

## MongoDB как хранилище данных

На данной странице рассказывается о варианте установки с использованием FerretDB - хранилище с открытым исходным ходом, которое использует PostgreSQL как движок базы данных. Альтернативно вы можете установить MongoDB. Инструкцию по установке с использованием MongoDB можно прочитать по ссылке.

#### Разница с MongoDB

Вы не почувствуете значительных изменений и сможете общаться с базой аналогично MongoDB (в том числе используя MongoDB Compass и утилиту mongosh). Единственное значительное различие - FerretDB не поддерживает роли, поэтому создание новых пользователей базы данных должно происходить следующим образом:

```
db.createUser({
  user: 'newuser',
  pwd: 'newpassword',
  roles: []
})
```

#### Начало установки

Подготовка пространства имён:

```
FERRET_USER=any_name
FERRET_PASS=ferret_password
FERRET_SERVICE_NAME=fr-ferret
NAMESPACE=ferret
kubectl create namespace $NAMESPACE
```

Обратите внимание на переменную FERRET\_SERVICE\_NAME - это имя в дальнейшем будет использоваться для обращения к FerretDB из сервера отчётов.

# Создание Deployment

```
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: apps/v1
kind: Deployment
metadata:
name: $FERRET_SERVICE_NAME
spec:
replicas: 1
selector:
  matchLabels:
   app: $FERRET SERVICE NAME
template:
 metadata:
   labels:
    app: $FERRET SERVICE NAME
  spec:
   containers:
    - name: $FERRET_SERVICE_NAME
     image: ghcr.io/ferretdb/ferretdb-eval:2
```

быстрыеотчеты.рф 19 / 76

```
ports:
       - containerPort: 27017
       - name: POSTGRES_USER
        value: $FERRET_USER
       - name: POSTGRES PASSWORD
        value: $FERRET_PASS
     volumeMounts:
     - mountPath: /var/lib/postgresql/data
       name: volume
   volumes:
    - name: volume
     persistentVolumeClaim:
       claimName: fr-ferret-pvc
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: ferret-storageclass
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
apiVersion: v1
kind: PersistentVolume
metadata:
 name: ferret-storage
 namespace: $NAMESPACE
 labels:
  type: local
spec:
 capacity:
  storage: 5Gi
 accessModes:
  - ReadWriteOnce
 storageClassName: ferret-storageclass
 local:
  path: /devkube/ferretdb
 nodeAffinity:
  required:
   nodeSelectorTerms:
   - matchExpressions:
    - key: kubernetes.io/hostname
     operator: In
     values:
     - node1
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: fr-ferret-pvc
spec:
 accessModes:
  - ReadWriteOnce
 storageClassName: ferret-storageclass
 resources:
  requests:
   storage: 2Gi
apiVersion: v1
kind: Service
metadata:
 name: $FERRET_SERVICE_NAME
 labels:
  app: $FERRET_SERVICE_NAME
spec:
 selector:
```

быстрыеотчеты.рф 20 / 76

app: \$FERRET\_SERVICE\_NAME
ports:

- port: 27017 targetPort: 27017

EOF

Проверить корректность установки FerretDB проще всего при помощи Kubernetes Dashboard, установка которого была описана ранее.

#### Внимание

Проверьте создалась ли на ноде директория /devkube/ferretdb , если нет - создайте её самостоятельно.

быстрыеотчеты.рф 21 / 76

# Установка компонентов сервера отчётов МоиОтчеты Корпоративный Сервер на узлы кластера Kubernetes

Продукты: МоиОтчеты Корпоративный Сервер

#### Получение образов Docker

Образы компонентов доступны в реестре по следующим URL. Вы можете загрузить их с помощью docker pull :

docker pull xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-app:latest

 $docker\ pull\ xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-backend: latest$ 

docker pull xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-designer:latest

 $docker\ pull\ xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-fonts:latest$ 

docker pull xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-gateway:latest

 $docker\ pull\ xn--80akiaokt 3b4b. xn--90aia 9 aifhdb 2 cxbdg. xn--p1ai/repository/docker-registry/moiotchety-corporate-server-homer: latest$ 

 $docker\ pull\ xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-scheduler: latest$ 

docker pull xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-static-preview:latest

docker pull xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-wasm-preview:latest

 $docker\ pull\ xn--80 akia okt 3b4b. xn--90 aia 9aifhdb 2cxbdg. xn--p1 ai/repository/docker-registry/moiotchety-corporate-server-workercore: latest$ 

**Примечание:** Загружать образы на каждую ноду вручную не обязательно. Достаточно указать нужные образы и теги в манифестах Kubernetes (Deployment, StatefulSet и т.д.). Kubernetes сам скачает необходимые образы при запуске подов.

#### Описание микросервисов

НАЗВАНИЕ СЕРВИСА	НАЗНАЧЕНИЕ
арр	Панель пользователя — интерфейс для работы с отчётами.
backend	Основное ядро системы — обрабатывает запросы, управляет доступом к БД.
designer	Дизайнер отчётов — инструмент для создания и редактирования отчётов.
fonts	Сервис обработки шрифтов, необходим для корректного отображения отчётов.
gateway	Шлюз — центральная точка входа для всех НТТР-запросов.
homer	Панель администрирования — управление пользователями, настройками и т.д.

быстрыеотчеты.рф 22 / 76

НАЗВАНИЕ СЕРВИСА	НАЗНАЧЕНИЕ
scheduler	Менеджер задач — управляет расписанием выполнения отчётов.
static-preview	Фронтенд для просмотра отчётов в статичном виде.
wasm-preview	Фронтенд для интерактивного просмотра отчётов (только для С# отчётов).
workercore	Ядро обработки отчётов — выполняет генерацию и построение отчётов.

## Поддерживаемые теги образов

TEF	ОПИСАНИЕ
:latest	Последняя версия на основе контейнеров Debian.
:debian-latest	Последняя версия на основе контейнеров Debian.
:redos-latest	Последняя версия на основе контейнеров RedOS (ubi).
:debian-{version}	Конкретная версия на основе Debian, например: :debian-2025.2.10.
:redos-{version}	Конкретная версия на основе RedOS, например: :redos-2025.2.10.

#### Важно:

- Используйте **конкретные версии** образов (например, idebian-2025.2.10), а не ilatest. Это обеспечит стабильность и предсказуемость при развертывании.
- Все микросервисы должны использовать **одинаковую версию** образов для обеспечения совместимости.

### Дополнительные замечания

- Обновления могут не затрагивать все микросервисы сразу. Однако для стабильной работы рекомендуется обновлять все компоненты одновременно.
- В случае проблем с совместимостью обращайтесь к технической поддержке.

быстрыеотчеты.рф 23 / 76

# Настройка сертификатов

Продукты: МоиОтчеты Корпоративный Сервер

Для работы сервера отчётов необходимо создать сертификат.

Чтобы создать самостоятельно подписанный сертификат, используйте следующие команды:

#!/bin/sh

openssl req -x509 -nodes -new -sha256 -days 1024 -newkey rsa:2048 -keyout RootCA.key -out RootCA.pem -subj
"/C=US/CN=debian-master.fast-report.com"

openssl x509 -outform pem -in RootCA.pem -out RootCA.crt

openssl pkcs12 -export -out ./certificate.pfx -inkey RootCA.key -in RootCA.crt

Чтобы создать секрет, используйте следующие команды:

NAMESPACE=fr-corporate-server
SECRET\_VOLUME\_NAME=corporate-volume-secret
kubectl create namespace \$NAMESPACE
kubectl create secret generic \$SECRET\_VOLUME\_NAME -n \$NAMESPACE --from-file=certificate.pfx

# Альтернативный способ генерации сертификата, предложенный пользователем продукта

1. Создайте и отредактируйте файл san.cnf:

```
[req]
default bits = 2048
default md = sha256
distinguished name = req distinguished name
req extensions = v3 req
[ req distinguished name ]
countryName = CN # C=
stateOrProvinceName = Shanghai # ST=
localityName = MyCity # L=
#postalCode = 200000 # L/postalcode=
#streetAddress = "My Address" # L/street=
organizationName = My Corporation # O=
organizationalUnitName = My Department # OU=
commonName = myname.mysoftware.mycorporation.com # CN=
emailAddress = myname@example.com # CN/emailAddress=
[ v3 req ]
subjectAltName = @alt_names
[ alt_names ]
DNS.1 = myname.mysoftware.mycorporation.com
#DNS.2 = other2.com
#DNS.3 = other3.coM
```

2. Сгенерируйте сертификат:

```
openssl req -x509 -nodes -days 365 -subj "/C=CN/ST=Shanghai/L=Shanghai/O=My Corporation/OU=My Department/CN=myname.mysoftware.mycorporation.com/emailAddress=myname@example.com" -keyout privateKey.pem -out public.crt -config san.cnf -extensions v3_req openssl pkcs12 -export -out ./certificate.pfx -inkey privateKey.key -in public.crt
```

3. Создайте секрет в облачном сервере:

быстрыеотчеты.рф 24 / 76

NAMESPACE=fr-corporate-server SECRET\_VOLUME\_NAME=corporate-volume-secret

kubectl create namespace \$NAMESPACE kubectl create secret generic \$SECRET\_VOLUME\_NAME -n \$NAMESPACE --from-file=certificate.pfx

быстрыеотчеты.рф 25 / 76

# Настройка параметров МоиОтчеты Корпоративный Сервер

Продукты: МоиОтчеты Корпоративный Сервер

Данный раздел описывает процесс конфигурирования сервера отчётов. Здесь описана базовая информация. Для тонкой настройки сервера обратитесь к руководству <a href="mailto:admi\_config">admi\_config</a>. Конфигурация сохраняется в защищенном хранилище Kubernetes. Эту операцию оптимально выполнить до старта контейнеров, описанных в секции fr-corporate-server.

Базовые параметры сервера отчётов определены ниже. Для удобства они показаны в отдельной секции. Возможные варианты: поместить этот фрагмент в отдельный файл и с помощью операции точка внедрить их в конфигурационный скрипт. Или же скопировать этот фрагмент непосредственно в конфигурационный скрипт. Так возможно дописать export перед каждой переменной и экспортировать в переменные среды.

Значение переменной MONGO\_HOST (или FERRET\_HOST) можно проверить в Kubernetes Dashboard - оно соответствует имени сервиса и его пространству имён, разделённых точкой. Для успешного старта сервера отчётов необходима лицензия - специальная строка, описывающая права использования сервера отчётов. Получите её у компании ООО "Быстрые отчеты".

Значения параметров MONGO\_USER и MONGO\_PASS (или FERRET\_USER и FERRET\_PASS) должны соответствовать тем же самым параметрам, что указаны при установке MongoDB (FerretDB).

Вариант для MongoDB:

быстрыеотчеты.рф 26 / 76

NAMESPACE=fr-corporate-server CLOUD\_ENV=prod

## Пароль и имя пользователя могут быть установлены любые, главное, чтобы совпадали с теми, что были заданы при установке MongoDB.

## MONGO DB должна быть admin.

## Однако, есть правила установки имени хоста - оно образуется из имени сервиса и пространства имён, в котором зарегистрирована MongoDB.

MONGO\_HOST=fr-mongo-mongodb.mongo MONGO\_USER=root MONGO\_PASS=mxzgrkGvvk MONGO\_DB=admin

## Следующие параметры лучше ЗДЕСЬ не изменять. Хост и имя пользователя устанавливаются автоматически, а ## пароль для RabbitMQ генерируется автоматически на момент установки и сохраняется в хранилище секретов Kubernetes.

RABBITMQ HOST=rabbitmq.rabbitmq.svc.cluster.local

# RABBITMQ\_HOST=rabbitmq-0.rabbitmq-headless.rabbitmq

RABBITMQ\_USER=user

RABBITMQ\_PASS=\$(kubectl get secret --namespace rabbitmq rabbitmq -o jsonpath="{.data.rabbitmq-password}" | base64 --decode)

## Разработчики утверждают, что здесь может быть любая последовательность символов. И длина не опрелена. ## Для теста были использованы следующие значения, и установка прошла успешно

## В эту переменную должна быть вписана лицензия, полученная у компании ООО "Быстрые отчеты". Увы, без лицензии сервер отчётов работать не будет. ## Не используйте кавычки.

#### LICENSE=

## Строка подключения к MongoDB формируется автоматически на основе ранее определённых переменных. Тут лучше ничего не менять.

MONGO\_ACCESS="mongodb://\$MONGO\_USER:\$MONGO\_PASS@\$MONGO\_HOST:27017/"

CONNECTION\_STRING="\$MONGO\_ACCESS?

authSource=\$MONGO\_DB&readPreference=primary&appname=MongoDB%20Compass&ssl=false&maxPoolSize=100&wait
QueueMultiple=100"

Вариант для FerretDB:

быстрыеотчеты.рф 27 / 76

```
NAMESPACE=fr-corporate-server
CLOUD_ENV=prod
## Пароль и имя пользователя могут быть установлены любые, главное, чтобы совпадали с теми, что были заданы
при установке FerretDB.
## FERRET DB должна быть admin.
## Однако, есть правила установки имени хоста - оно образуется из имени сервиса и пространства имён, в котором
зарегистрирована FerretDB.
FERRET_HOST=fr-ferret.ferret
FERRET_USER=any_name
FERRET_PASS=ferret_password
FERRET DB=admin
## Следующие параметры лучше ЗДЕСЬ не изменять. Хост и имя пользователя устанавливаются автоматически, а
## пароль для RabbitMQ генерируется автоматически на момент установки и сохраняется в хранилище секретов
Kubernetes.
RABBITMQ HOST=rabbitmq.rabbitmq.svc.cluster.local
# RABBITMQ_HOST=rabbitmq-0.rabbitmq-headless.rabbitmq
RABBITMQ USER=user
RABBITMQ_PASS=$(kubectl get secret --namespace rabbitmq rabbitmq -o jsonpath="{.data.rabbitmq-password}" | base64 -
-decode)
## Разработчики утверждают, что здесь может быть любая последовательность символов. И длина не определена.
## Для теста были использованы следующие значения, и установка прошла успешно
DESIGNER SECRET=yXgQDpNzohhN3dAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHrXiepieQtOaLwdqfRijMaZwy5p27tho7XK4C
WORKER_SECRET=9bYuYu4cvt73oWD3AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA5Z6i6pRzkR8mKGL1pTRnns3P0clxzTuVu
GATEWAY SECRET=R0jEINg6OA4TssyAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAY46ZtsYowI7gB9yteZNMSvPoL6sGgiWC4
SCHEDULER SECRET=fcfFZNWC3zyBmAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA5gD2MeexV3svLdy87qC6bX3bM2W6m
ZjRC
## В эту переменную должна быть вписана лицензия, полученная у компании ООО "Быстрые отчеты". Увы, без
лицензии сервер отчётов работать не будет.
## Не используйте кавычки.
LICENSE=
## Строка подключения к FerretDB формируется автоматически на основе ранее определённых переменных. Тут
лучше ничего не менять.
FERRET ACCESS="mongodb://$FERRET USER:$FERRET PASS@$FERRET HOST:27017/"
CONNECTION STRING="$FERRET ACCESS?
auth Source = \$FERRET\_DB\& read Preference = primary \& appname = MongoDB\%20 Compass \& ssl = false \& maxPoolSize = 100 \& wait = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoDB\%20 Compass & ssl = false \& maxPoolSize = 100 MongoD
```

Регистрация пространства имён сервера отчётов в кластере:

kubectl create namespace \$NAMESPACE

QueueMultiple=100"

Создание минимальной конфигурации сервера отчётов:

```
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: ConfigMap
metadata:
name: fast-report-config
data:
appsettings.Production.json: |
{
    "Auth": {
```

быстрыеотчеты.рф 28 / 76

```
"UseOpenId": false.
 "UseLocal": true
},
"MainConfig": {
 "InternalHeaders": {
 },
"License": "$LICENSE",
 "Server": {
 "Title": "<Заголовок в приложении>",
 "Copyright": "<Краткое описание авторского права>",
 "LogoLink": "<путь к логотипу, отображаемому в панели пользователя>",
 "LastSlaUpdate": "2022-11-28",
 "CorporateServerMode": false,
 "SlaLink": "<Ссылка на пользовательское соглашение>",
 "FirstStepsVideoLink": "https://rutube.ru/play/embed/f64e80d2fa8e2d73b30c47483fa2b59e",
 "AboutLink": "https://быстрыеотчеты.pф/products/corporate-server",
 "HomePageLink": "быстрыеотчеты.рф",
 "AuthServerName": "ID Быстрые отчеты",
 "UsersPerWorkSpace": 0
},
 "Rabbit": {
  "Host": "$RABBITMQ_HOST",
  "Port": 5672,
  "UserName": "$RABBITMQ USER",
  "Password": "$RABBITMQ PASS",
  "QueueName": "ReportProcessQueue",
  "DirectExchangeName": "DirectEx",
  "AlternateExchangeName": "AExchange",
  "UnroutedQueueName": "Default",
  "Capacity": 1
 },
 "Database": {
  "ConnectionString": "$CONNECTION_STRING",
  "DatabaseName": "$MONGO DB"
 }
},
"Gateway": {
 "BackendUrl": "http://fr-backend.$NAMESPACE:80",
 "InternalKey": "$GATEWAY SECRET",
 "SignInPagePath": "/account/signin?r={0}",
 "MaxConcurrentRequests": 200,
 "RequestQueueLimit": 5000
},
"Serilog": {
"MinimumLevel": {
  "Default": "Debug"
}
},
"Services": {
 "Items": {
  "OnlineDesigner": {
   "Namespace": "$NAMESPACE"
 },
  "Backend": {
   "Namespace": "$NAMESPACE"
  "FrontendApp": {
  "Namespace": "$NAMESPACE"
  },
  "Fonts": {
  "Namespace": "$NAMESPACE"
```

быстрыеотчеты.рф 29 / 76

```
},
      "StaticPreviewApp": {
       "K8sServiceName": "fr-s-preview",
       "Namespace": "$NAMESPACE",
       "HostType": "WebApp",
       "PathBase": "/staticpreview",
       "Type": "K8s",
       "PingPath": "/staticpreview/",
       "IsSignInRequired": false,
       "Priority": 21
     "Default": {
      "Namespace": "$NAMESPACE"
     "HomerApp": {
       "Namespace": "$NAMESPACE",
       "WhiteListClaims": {
       "cloud_service_access": "super_user"
      }
     }
    }
   },
   "Designer": {
    "BackendUrl": "http://fr-backend.$NAMESPACE:80",
    "InternalKey": "$DESIGNER_SECRET"
   },
   "WorkerCore": {
    "BackendUrl": "http://fr-backend.$NAMESPACE:80",
    "InternalKey": "$WORKER_SECRET"
   },
   "Scheduler": {
    "BackendUrl": "http://fr-backend.$NAMESPACE:80",
    "InternalKey": "$SCHEDULER_SECRET"
   }
  }
EOF
```

#### Используя FerretDB замените

```
"Database": {
    "ConnectionString": "$CONNECTION_STRING",
    "DatabaseName": "$MONGO_DB"
}
```

на

```
"Database": {

"ConnectionString": "$CONNECTION_STRING",

"DatabaseName": "$FERRET_DB"
}
```

Обратите внимание, вышеприведённый фрагмент shell-скрипта содержит базовые настройки, проверенные разработчиками. При первоначальном разворачивании сервера отчётов рекомендуется использовать их, чтобы убедиться в работоспособности сервера. Перед изменением настроек сохраните эту версию, таким образом вы сэкономите значительно время при тонкой настройке сервера, используя параметры, описанные в секции admin\_config.

быстрыеотчеты.рф 30 / 76

# Установка МоиОтчеты Корпоративный Сервер

Продукты: МоиОтчеты Корпоративный Сервер

Минимальная конфигурация корпоративного сервера отчётов состоит из девяти компонентов, описанных ниже. Перед установкой необходимо определить переменные, например, в файле config.sh. Во избежание конфликтов имён лучше иметь глобальный файл конфигурации для всех скриптов отчёта. Ниже показан пример инициализации переменных, используемых скриптами установки сервера отчётов:

IMAGE\_STORAGE=xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry
IMAGE\_TAG=debian-2025.1.10
NAMESPACE=fr-corporate-server
SECRET\_VOLUME\_NAME=corporate-volume-secret
IMAGE\_REGISTRY\_SECRET\_NAME=storage\_secret
SECRET\_VOLUME\_NAME=corporate-volume-secret
PULL\_POLICY=Never

#### Описание переменных:

IMAGE\_STORAGE - адрес хранилища образов сервисов сервера отчётов, Корпоративный Сервер использует локальное хранилище, наполнение образами которого описано в разделе Подготовка образов.

IMAGE\_TAG - версия сервера отчётов.

NAMESPACE - пространство имён Kubernetes, в котором будет выполняться сервер отчётов. SECRET\_VOLUME\_NAME - имя тома, на котором будет храниться конфигурация и информация для авторизации.

IMAGE\_REGISTRY\_SECRET\_NAME - имя записи, сохраняющей информацию для авторизации на сервере, хранящем образы сервисов сервере отчётов. PULL\_POLICY - определяет как Kubernetes будет скачивать образы сервисов. Возможные значения:

- Always всегда скачивать образы при старте сервиса;
- IfNotPresent скачивать образ только если он не присутствует локально;
- Never использовать заранее импортированные образы, никогда не скачивать их с внешнего сервера.

Обратите внимание: описанная конфигурация не скачивает образы из внешнего хранилища (PULL\_POLICY=Never), эти переменные должны быть заданы во избежание ошибок при применении конфигурации. Перед установкой значения PULL\_POLICY обратитесь к секции fr-images для ознакомления с настройками.

Настройка параметров сервера отчётов описана в документе Конфигурация сервера отчётов. В этом документе описаны базовые параметры, определяющие режимы работы сервера отчётов.

## Gateway - сервис обработки входящих запросов

Gateway это сервис, обрабатывающий все соединения к серверу отчётов. Он анализирует запросы и распределяет их между другими сервисами. Если сервис Gateway не работает или работает неправильно, то ни один из компонентов сервера отчётов работать не будет. Ingress конфигурация по умолчанию отправляет все внешние запросы именно этому сервису. В случае каких-либо проблем с сервисом начните анализ проблемы с просмотров логов сервиса fr-gateway.

#!/bin/sh

HTTPS\_NODE\_PORT=32222

IMAGE="\${IMAGE\_STORAGE}/moiotchety-corporate-server-gateway:\${IMAGE\_TAG}"

быстрыеотчеты.рф 31 / 76

```
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 name: fr-gateway
 namespace: $NAMESPACE
 labels:
  app.kubernetes.io/name: fr-gateway
- apiGroups: [""]
  resources:
   - services
   - endpoints
   - pods
  verbs: ["get", "list", "watch"]
apiVersion: v1
kind: ServiceAccount
metadata:
 name: fr-gateway
 namespace: $NAMESPACE
 labels:
  app.kubernetes.io/name: fr-gateway
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
name: fr-gateway
 namespace: $NAMESPACE
  app.kubernetes.io/name: fr-gateway
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: fr-gateway
subjects:
- kind: ServiceAccount
 name: fr-gateway
 namespace: $NAMESPACE
apiVersion: v1
kind: Service
metadata:
name: fr-gateway
 namespace: $NAMESPACE
spec:
 type: NodePort
 selector:
  app: fr-gateway
 ports:
  - name: http
   protocol: TCP
   port: 80
   nodePort: 32223
  - name: https
   protocol: TCP
   port: 5005
   nodePort: $HTTPS_NODE_PORT
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
```

быстрыеотчеты.рф 32 / 76

```
metadata:
name: fr-gateway
 namespace: $NAMESPACE
spec:
replicas: 1
selector:
  matchLabels:
   app: fr-gateway
 strategy:
  type: Recreate
 template:
  metadata:
   namespace: $NAMESPACE
   labels:
    app: fr-gateway
  spec:
   serviceAccountName: fr-gateway
   containers:
   - image: $IMAGE
    imagePullPolicy: $PULL_POLICY
    name: fr-gateway
    #env:
    # # Use secret in real usage
    #- name: MYSQL_ROOT_PASSWORD
    # value: password
    ports:
    - containerPort: 80
     name: fr-gateway
    volumeMounts:
    - name: config-volume
     mountPath: /app/appsettings.Production.json
     subPath: appsettings.Production.json
    - name: secret-volume
     mountPath: /etc/cert
   imagePullSecrets:
    - name: $IMAGE_REGISTRY_SECRET_NAME
   volumes:
   - name: config-volume
    configMap:
     name: fast-report-config
     - key: appsettings.Production.json
       path: appsettings.Production.json
   - name: secret-volume
    secret:
     secretName: $SECRET_VOLUME_NAME
EOF
```

## Установка планировщика сервера отчётов

быстрыеотчеты.рф 33 / 76

```
IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-scheduler:${IMAGE_TAG}"
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: Service
metadata:
name: fr-scheduler
namespace: $NAMESPACE
spec:
type: ClusterIP
ports:
- port: 80
selector:
  app: fr-scheduler
EOF
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: apps/v1
kind: Deployment
metadata:
name: fr-scheduler
namespace: $NAMESPACE
spec:
replicas: 1
 selector:
  matchLabels:
   app: fr-scheduler
 strategy:
  type: Recreate
 template:
  metadata:
   namespace: $NAMESPACE
   labels:
    app: fr-scheduler
  spec:
   containers:
   - image: $IMAGE
    imagePullPolicy: $PULL_POLICY
    name: fr-scheduler
    #env:
    # # Use secret in real usage
    #- name: MYSQL_ROOT_PASSWORD
    # value: password
    ports:
    - containerPort: 80
     name: fr-scheduler
    volumeMounts:
    - name: config-volume
     mountPath: /app/appsettings.Production.json
    - name: secret-volume
     mountPath: /etc/cert
   imagePullSecrets:
    - name: $IMAGE_REGISTRY_SECRET_NAME
   volumes:
   - name: config-volume
    configMap:
     name: fast-report-config
   - name: secret-volume
    secret:
     secretName: $SECRET_VOLUME_NAME
EOF
```

### Static Preview - сервис просмотра отчётов

быстрыеотчеты.рф 34 / 76

```
#!/bin/sh
IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-static-preview:${IMAGE_TAG}"
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: Service
metadata:
 name: fr-s-preview
 namespace: $NAMESPACE
spec:
 type: ClusterIP
 ports:
 - port: 80
 selector:
  app: fr-s-preview
apiVersion: apps/v1
kind: Deployment
metadata:
name: fr-s-preview
 namespace: $NAMESPACE
spec:
 replicas: 1
 selector:
  matchLabels:
   app: fr-s-preview
 strategy:
  type: Recreate
 template:
  metadata:
   namespace: $NAMESPACE
   labels:
    app: fr-s-preview
  spec:
   containers:
   - image: $IMAGE
    resources:
     limits:
      memory: 200Mi
     requests:
       memory: 200Mi
    imagePullPolicy: $PULL_POLICY
    name: fr-s-preview
    #env:
    # # Use secret in real usage
    #- name: MYSQL_ROOT_PASSWORD
    # value: password
    ports:
    - containerPort: 80
     name: fr-s-preview
   imagePullSecrets:
    - name: $IMAGE_REGISTRY_SECRET_NAME
EOF
```

# **Сервис Homer**

быстрыеотчеты.рф 35 / 76

```
#!/bin/sh
IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-homer:${IMAGE_TAG}"
SECRET_VOLUME_NAME=corporate-volume-secret
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: Service
metadata:
name: fr-homer
namespace: $NAMESPACE
spec:
type: ClusterIP
ports:
- port: 80
selector:
 app: fr-homer
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
name: fr-homer
namespace: $NAMESPACE
spec:
replicas: 1
selector:
 matchLabels:
   app: fr-homer
strategy:
  type: Recreate
template:
  metadata:
   namespace: $NAMESPACE
   labels:
    app: fr-homer
  spec:
   containers:
   - image: $IMAGE
    imagePullPolicy: $PULL_POLICY
    name: fr-homer
    #env:
    # # Use secret in real usage
    #- name: MYSQL_ROOT_PASSWORD
    # value: password
    - containerPort: 80
     name: fr-homer
   imagePullSecrets:
    name: $IMAGE_REGISTRY_SECRET_NAME
```

## Сервис Backend

```
#!/bin/sh

IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-backend:${IMAGE_TAG}"

cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: Service
metadata:
name: fr-backend
```

быстрыеотчеты.рф 36 / 76

```
namespace: $NAMESPACE
spec:
type: ClusterIP
ports:
- port: 80
selector:
  app: fr-backend
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
name: fr-backend
namespace: $NAMESPACE
spec:
 replicas: 3
selector:
  matchLabels:
   app: fr-backend
strategy:
  type: Recreate
 template:
  metadata:
   namespace: $NAMESPACE
   labels:
    app: fr-backend
  spec:
   containers:
   - image: $IMAGE
    resources:
     limits:
      memory: 400Mi
     requests:
      memory: 400Mi
    imagePullPolicy: $PULL_POLICY
    name: fr-backend
    env:
    - name: Serilog Using 0
     value: FastReport.Cloud.Base.Mvc
    ports:
    - containerPort: 80
     name: fr-backend
    volumeMounts:
    - name: config-volume
     mountPath: /app/appsettings.Production.json
     subPath: appsettings.Production.json
   imagePullSecrets:
    - name: $IMAGE_REGISTRY_SECRET_NAME
   volumes:
   - name: config-volume
    configMap:
     name: fast-report-config
     - key: appsettings.Production.json
       path: appsettings.Production.json
EOF
```

## Cepsuc Designer - онлайн дизайнер отчётов

```
#!/bin/sh

IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-designer:${IMAGE_TAG}"

cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1</pre>
```

быстрыеотчеты.рф 37 / 76

```
kind: Service
metadata:
 name: fr-designer
 namespace: $NAMESPACE
spec:
 type: ClusterIP
 ports:
- port: 80
 selector:
  app: fr-designer
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
name: fr-designer
 namespace: $NAMESPACE
spec:
 replicas: 1
 selector:
  matchLabels:
   app: fr-designer
 strategy:
  type: Recreate
 template:
  metadata:
   labels:
    app: fr-designer
   namespace: $NAMESPACE
  spec:
   containers:
   - image: $IMAGE
    imagePullPolicy: $PULL_POLICY
    name: fr-designer
    #env:
    # # Use secret in real usage
    #- name: MYSQL ROOT PASSWORD
    # value: password
    ports:
    - containerPort: 80
     name: fr-designer
    volumeMounts:
    - name: config-volume
     mountPath: /app/appsettings.Production.json
     subPath: appsettings.Production.json
    - name: secret-volume
     mountPath: /etc/cert
   imagePullSecrets:
    - name: $IMAGE_REGISTRY_SECRET_NAME
   volumes:
   - name: config-volume
    configMap:
     name: fast-report-config
     - key: appsettings.Production.json
       path: appsettings.Production.json
   - name: secret-volume
    secret:
     secretName: $SECRET_VOLUME_NAME
EOF
```

## Сервис шрифтов

#!/bin/sh

быстрыеотчеты.рф 38 / 76

```
IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-fonts:${IMAGE_TAG}"
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: Service
metadata:
 name: fr-fonts
 namespace: $NAMESPACE
spec:
 type: ClusterIP
 ports:
 - port: 80
 selector:
  app: fr-fonts
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: fr-fonts
 namespace: $NAMESPACE
spec:
 replicas: 1
 selector:
  matchLabels:
   app: fr-fonts
 strategy:
  type: Recreate
 template:
  metadata:
   labels:
    app: fr-fonts
   namespace: $NAMESPACE
  spec:
   containers:
   - image: $IMAGE
    resources:
     limits:
      memory: 200Mi
     requests:
       memory: 200Mi
    imagePullPolicy: $PULL_POLICY
    name: fr-fonts
    - name: Serilog__Using__0
     value: FastReport.Cloud.MvcExtentions
    ports:
    - containerPort: 80
     name: fr-fonts
    volumeMounts:
    - name: config-volume
     mountPath: /app/appsettings.Production.json
     subPath: appsettings.Production.json
   imagePullSecrets:
    - name: $IMAGE_REGISTRY_SECRET_NAME
   volumes:
   - name: config-volume
    configMap:
     name: fast-report-config
     items:
     - key: appsettings.Production.json
       path: appsettings.Production.json
EOF
```

быстрыеотчеты.рф 39 / 76

## Сервис облачного приложения

```
#!/bin/sh
IMAGE="${IMAGE_STORAGE}/moiotchety-corporate-server-app:${IMAGE_TAG}"
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: v1
kind: Service
metadata:
name: fr-app
namespace: $NAMESPACE
spec:
type: ClusterIP
ports:
- port: 80
selector:
  app: fr-app
apiVersion: apps/v1
kind: Deployment
metadata:
name: fr-app
namespace: $NAMESPACE
spec:
replicas: 1
selector:
  matchLabels:
   app: fr-app
strategy:
  type: Recreate
 template:
  metadata:
   namespace: $NAMESPACE
   labels:
    app: fr-app
  spec:
   containers:
   - image: $IMAGE
    imagePullPolicy: $PULL POLICY
    name: fr-app
    #env:
    # # Use secret in real usage
    #- name: MYSQL_ROOT_PASSWORD
    # value: password
    ports:
    - containerPort: 80
     name: fr-app
   imagePullSecrets:
    - name: $IMAGE REGISTRY SECRET NAME
EOF
```

# Сервис построения отчётов

быстрыеотчеты.рф 40 / 76

```
#!/bin/sh
IMAGE="${IMAGE STORAGE}/moiotchety-corporate-server-workercore:${IMAGE TAG}"
cat <<EOF | kubectl apply -n $NAMESPACE -f -
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: fr-workercore
 namespace: $NAMESPACE
spec:
replicas: 4
selector:
  matchLabels:
   app: fr-workercore
 strategy:
  type: Recreate
 template:
  metadata:
   labels:
    app: fr-workercore
   namespace: $NAMESPACE
  spec:
   containers:
   - image: $IMAGE
    imagePullPolicy: $PULL_POLICY
    name: fr-workercore
    volumeMounts:
    - name: config-volume
     mountPath: /app/appsettings.Production.json
     subPath: appsettings.Production.json
    - name: secret-volume
     mountPath: /etc/cert
   imagePullSecrets:
    - name: $IMAGE REGISTRY SECRET NAME
   volumes:
   - name: config-volume
    configMap:
     name: fast-report-config
     items:
     - key: appsettings.Production.json
       path: appsettings.Production.json
   - name: secret-volume
    secret:
     secretName: $SECRET_VOLUME_NAME
EOF
```

После выполнения всех вышеописанных скриптов установка сервера отчётов закончена. Для того чтобы проверить успешность установки и начать работу, зайдите с помощью браузера на сервер отчётов, используя доменное имя, указанное в переменной HOST при настройке nginx-ingress и в появившемся окне введите следующие данные:

Имя пользователя: admin@example.com

Пароль: admin

Приятной работы!

быстрыеотчеты.рф 41 / 76

## Конфигурация

Продукты: МоиОтчеты Корпоративный Сервер

Конфигурация МоиОтчеты Корпоративный Сервер осуществляется через файлы appsettings.json, которые лежат в директории приложения. Эти файлы по умолчанию уже имеют ряд свойств, которые можно переопределить одним из трех способов:

1. Через appsettings.{Environment}.json, по умолчанию переменная Environment имеет значение Production, поэтому достаточно изменить содержание файла appsettings.Production.json. Регистр имеет значение!

Для смены значения используйте переменные среды ASPNETCORE\_ENVIRONMENT и DOTNET ENVIRONMENT.

```
export ASPNETCORE_ENVIRONMENT=Production
export DOTNET_ENVIRONMENT=Production
```

2. Через переменные среды. Подробности описаны ниже.

Приоритезация загрузки конфига appsettings.json -> appsettings.{Environment}.json -> переменные среды. Это означает, что конфигурация будет загружена слева направо, иначе говоря, те файлы что находятся правее перезаписывают конфиг предыдущих.

## Описание формата appsettings.json

В каждом из разделов будет приведён пример конфигурации и описательная часть: ключ -> значение.

```
{
    "Logging":[
        {
            "Name":"ToEmail"
        }
        ]
}
```

Этот ключ можно использовать как переменную среды:

```
export Logging_0_Name=ToEmail
```

Здесь запись Logging\_0\_Name означает обращение к разделу Logging , элементу под индексом 0 и свойство Name.

## Раздел Kestrel

Позволяет конфигурировать http сервер для прослушивания определённого порта или использования сертификата. Пример конфига:

быстрыеотчеты.рф 42 / 76

```
{
 "Kestrel":{
   "Endpoints":{
     "Http":{
       "Url": "http://localhost:5000"
     },
     "Https":{
       "Url": "https://localhost:5001",
       "Certificate":{
         "Path":"<путь к файлу .pfx>",
         "Password":"<пароль к сертификату>"
       }
     }
   }
 }
}
```

ключ	тип	ОПИСАНИЕ
Kestrel_Endpoints_Http_Url	string (uri)	Точка доступа для прослушивания http.
Kestrel_Endpoints_Https_Url	string (uri)	Точка доступа для прослушивания https.
Kestrel_Endpoints_Https_Certificate_Path	string (local path)	Путь к файлу .pfx сертификата для https трафика.
KestrelEndpointsHttps_Certificate_Password	string	Пароль для доступа к .pfx файлу.

### Раздел Auth

Позволяет конфигурировать процесс аутентификации и авторизации.

```
{
 "Auth":{
   "ClientId":"<клиент OpenID>",
   "Scopes": "OpenID и другие области видимости",
   "Authority": "https://id.быстрыеотчеты.рф",
   "Audience":"https://облако.моиотчеты.рф",
   "Secret":"<секрет для клиента OpenID>",
   "UseApiKeys":true,
   "UseOpenId":true,
   "UseLocal":false,
   "AllowLocalSignUp":false,
   "RsaXml":"<XML-кодированный объект>",
   "UserInfoEndpoint": "https://example.com/userinfo",
   "TokenEndpoint": "https://example.com/token",
   "MetadataEndpoint": "https://example.com/.well-known/openid-configuration"
 }
}
```

ключ	тип	ОПИСАНИЕ
AuthClientId	string	Уникальный идентификатор на сервере аутентификации по протоколу openid .
AuthScopes	string	Список областей клиента по протоколу openid.

быстрыеотчеты.рф 43 / 76

ключ	тип	ОПИСАНИЕ
AuthAuthority	string (uri)	Точка доступа Authority у которой будет запрашиваться токен доступа по протоколу openid.
Auth_Audience	string (uri)	Точка доступа Audience для которой будет запрашиваться токен доступа по протоколу openid.
Auth_Secret	string	Секретный токен для подтверждения авторизации по протоколу openid.
Auth_UseApiKeys	boolean	Включает или отключает возможность авторизации через ключи доступа.
Auth_UseOpenId	boolean	Включает или отключает возможность авторизации по протоколу openid, обратите внимание, клиент должен быть настроен для авторизации через code flow.
AuthUseLocal	boolean	Включает или отключает возможность авторизации по почте и паролю. Данные пользователей в этом случае хранятся локально в базе сервера.
AuthAllowLocalSignUp	boolean	Включает или отключает регистрацию со стороны пользователя. Если отключено, то регистрация доступна только администраторам.
AuthRsaXml	string (xml)	RSA сериализованный в XML. Например, <rsakeyvalue><modulus>{base64}</modulus><exponent>{base64}  </exponent><p>{base64}</p><q>{base64}</q><dp>{base64}</dp> <dq>{base64}</dq><inverseq>{base64}</inverseq><d>{base64}</d> </rsakeyvalue>
Auth_UserInfoEndpoint	string (uri)	Точка доступа, через которую будут запрашиваться данные пользователя по протоколу openid.
Auth_TokenEndpoint	string (uri)	Точка доступа, через которую будет запрашиваться токен пользователя по протоколу openid.
AuthMetadataEndpoint	string (uri)	Точка доступа well-known с метаданными для openid.

### Раздел MainConfig

Позволяет конфигурировать основную конфигурацию МоиОтчеты Корпоративный Сервер.

```
{
 "MainConfig": {
   "Database":{
     "ConnectionString":"<строка подключения>",
     "DatabaseName": "ReportstoreDb",
     "ExportsCollectionName": "Exports",
     "ReportsCollectionName": "Reports",
     "TemplatesCollectionName": "Templates",
     "TemplateFoldersCollectionName": "TemplateFolders",\\
     "ReportFoldersCollectionName": "ReportFolders",
     "ExportFoldersCollectionName": "ExportFolders",\\
     "UsersCollectionName": "Users",
     "SubscriptionPlansCollectionName": "SubscriptionPlans",
     "SubscriptionsCollectionName": "Subscriptions",
     "SubscriptionInvitesCollectionName": "SubscriptionInvites",
     "GroupsCollectionName":"Groups",
     "DataSourceCollectionName"·"DataSources"
```

быстрыеотчеты.рф 44 / 76

```
DataSourceconcetionivanie . DataSour
 "GridFSFilesCollectionName": "fs.files",
 "MigrationsCollectionName": "Migrations",
 "CurrentTasksCollectionName": "CurrentTasks",
 "TasksCollectionName": "Tasks",
 "AuditCollectionName": "Audit",
 "ContactsCollectionName": "Contacts",
 "ContactGroupsCollectionName": "ContactGroups",
 "FontsCollectionName": "Fonts",
 "UserFontsCollectionName": "UserFonts"
},
"Server": {
 "Title": "<Заголовок в приложении>",
 "Copyright": "<Краткое описание авторского права>",
 "LogoLink" : "<путь к логотипу, отображаемому в панели пользователя>",
 "FaviconLink": "<путь к иконке в формате ico>",
 "LastSlaUpdate": "2022-11-28",
 "CorporateServerMode": true,
 "SlaLink": "<Ссылка на пользовательское соглашение>",
 "FirstStepsVideoLink": "https://rutube.ru/play/embed/f64e80d2fa8e2d73b30c47483fa2b59e",
 "AboutLink": "https://быстрыеотчеты.pф/products/corporate-server",
 "HomePageLink": "быстрыеотчеты.рф",
 "AuthServerName": "ID Быстрые отчеты",
 "UsersPerWorkSpace": null,
 "DataSourcesPerWorkSpace": null,
 "GroupsPerWorkSpace": null
},
"Pagination":{
 "MaxEntries":120
"Rabbit":{
 "Host": "my-rabbit-server.com",
 "Port":5672,
 "UserName":"<имя пользователя>",
 "Password":"<пароль пользователя>",
 "QueueName": "ReportProcessQueue",
 "DirectExchangeName": "DirectEx",
 "AlternateExchangeName": "AExchange",
 "UnroutedQueueName": "Default",
 "Capacity":1
"SecurityAdvisor":{
 "RestrictUnsafe":true,
 "RestrictUnmanaged":true,
 "RestrictExtern":true,
 "RestrictAsync":true,
 "RestrictTypeOf":true,
 "Whitelist":[
   "^\\w+:\\s+FastReport.*$",
   ^{^{^{^{^{^{^{^{^{^{^{^{^}}}}}}}}}}}
   "^\\w+\\:.*System\\.DateTime.*$",
   "^\\w+\\:.*System\\.Environment.*$"
 ],
 "Blacklist":[
   "^Method\\:.*\\.GetType\\(\\)$",
   "^\\w+\\:.*System\\.IO.*$",
   "^\\w+\\:.*FastReport\\.Utils\\.Config.*$",
   ^{\n} "^\\w+\\:.*System\\.Environment.*$",
   ^{^{^{^{^{^{^{^{^{^{^{^{^}}}}}}}}}}}
   ^{^{^{^{^{^{^{^{^{^{^{^{^}}}}}}}}}}}
   "^\\w+\\:.*System\\.Net.*$",
   ]
},
"License": "",
"SmtpServer": {
```

быстрыеотчеты.рф 45 / 76

```
"EnableSsl": true,
     "Server": "smtp.s.com",
     "Port": 587,
     "Username": "--",
     "From": "--",
     "Password": "--"
   },
   "Tasks": {
     "Attempts": 3
   },
   "Frontend": {
     "Mixins": {
      "Head": "",
       "Body": ""
     }
   },
   "Rfc2898CryptorSettings": {
     "Salt": "",
     "Password": "",
     "IV": ""
   },
   "DataSourcesConfig": {
     "XmlConfig": { "CommandTimeout": 30 },
     "JsonConfig": { "CommandTimeout": 30 },
     "CsvConfig": { "CommandTimeout": 30 },
     "MySqlConfig": { "CommandTimeout": 30 },
     "PostgreSqlConfig": { "CommandTimeout": 30 },
     "MsSqlConfig": { "CommandTimeout": 30 },
     "OracleDbConfig": { "CommandTimeout": 30 },
     "FirebirdConfig": { "CommandTimeout": 30 },
     "MongoDbConfig": { "CommandTimeout": 30 },
     "ClickHouseConfig": { "CommandTimeout": 30 }
   "FontServerAddress": "http://fr-fonts.fr-cloud:80"
 }
}
```

ключ	тип	ОПИСАНИЕ
MainConfigDatabaseConnectionString	string	Строка для подключения к базе данных MongoDB (FerretDB).
MainConfigDatabaseDatabaseName	string	Название базы данных сервера MongoDB (FerretDB).
MainConfig_Database_ExportsCollectionName	string	Название коллекции для хранения списка экспортов.
MainConfigDatabaseReportsCollectionName	string	Название коллекции для хранения списка отчётов.
MainConfigDatabaseTemplatesCollectionName	string	Название коллекции для хранения списка шаблонов.
MainConfigDatabaseTemplateFoldersCollectionName	string	Название коллекции для хранения древовидной структуры шаблонов.

быстрыеотчеты.рф 46 / 76

ключ	тип	ОПИСАНИЕ
MainConfigDatabaseReportFoldersCollectionName	string	Название коллекции для хранения древовидной структуры отчётов.
MainConfigDatabaseExportFoldersCollectionName	string	Название коллекции для хранения древовидной структуры экспортов.
MainConfigDatabaseUsersCollectionName	string	Название коллекции для хранения списка пользователей.
MainConfigDatabaseSubscriptionPlansCollectionName	string	Название коллекции для хранения списка планов подписки.
MainConfigDatabaseSubscriptionsCollectionName	string	Название коллекции для хранения списка подписок.
MainConfigDatabaseSubscriptionInvitesCollectionName	string	Название коллекции для хранения списка приглашений.
MainConfigDatabaseGroupsCollectionName	string	Название коллекции для хранения списка групп.
MainConfigDatabaseDataSourceCollectionName	string	Название коллекции для хранения списка источников данных.
MainConfigDatabaseGridFSFilesCollectionName	string	Название коллекции для хранения файлов.
MainConfigDatabaseMigrationsCollectionName	string	Название коллекции для хранения списка применённых миграций.
MainConfigDatabaseCurrentTasksCollectionName	string	Название коллекции для хранения задач. Вспомогательная коллекция для текущих задач.
MainConfigDatabaseTasksCollectionName	string	Название коллекции для хранения задач.
MainConfigDatabaseAuditCollectionName	string	Название коллекции для хранения аудитов.
MainConfigDatabaseContactsCollectionName	string	Название коллекции для хранения контактов.
MainConfigDatabaseContactGroupsCollectionName	string	Название коллекции для хранения групп контактов.

быстрыеотчеты.рф 47 / 76

ключ	тип	ОПИСАНИЕ
MainConfigDatabaseFontsCollectionName	string	Название коллекции для хранения шрифтов.
MainConfigDatabaseUserFontsCollectionName	string	Название коллекции для хранения пользовательских шрифтов.
MainConfig_Server_Title	string	Имя для сервера, которое будет использовать пользовательское приложение в заголовке страницы.
MainConfig_Server_Copyright	string	Текст для информации об авторском праве (или любой другой информации, которая будет отображена внизу страницы пользовательского приложения).
MainConfig_Server_LogoLink	string	Ссылка на файл с логотипом.
MainConfig_Server_FaviconLink	string	Ссылка на файл c favicon.
MainConfig_Server_LastSLAUpdate	string	Дата последнего изменения пользовательского соглашения, если изменить эту дату на более новую, при открытии пользовательского приложения откроется диалог, в котором будет ссылка на обновлённые условия (из поля SlaLink).
MainConfig_Server_CorporateServerMode	boolean	Это поле изменяет поведение части API, например, если указано false, то в админской панели нельзя будет удалять рабочие пространства.
MainConfig_Server_FirstStepsVideoLink	string	Ссылка на видео руководство к корпоративному серверу, которое появится при первом входе в пользовательское приложение или при нажатии на ? внизу страницы.
MainConfig_Server_AboutLink	string	Ссылка на информацию о продукте.
MainConfig_Server_HomePageLink	string	Ссылка на домашнюю страницу сайта продукта.
MainConfig_Server_AuthServerName	string	Имя для сервера авторизации, указанного в разделе AuthConfig, используемое в пользовательском приложении.

быстрыеотчеты.рф 48 / 76

ключ	тип	ОПИСАНИЕ
MainConfig_Server_UsersPerWorkSpace	integer?	Максимальное количество пользователей в рабочем пространстве, задаваемое администратором.
MainConfig_Server_DataSourcesPerWorkSpace	integer?	Максимальное количество источников данных в рабочем пространстве, задаваемое администратором.
MainConfig_Server_GroupsPerWorkSpace	integer?	Максимальное количество групп в рабочем пространстве, задаваемое администратором.
MainConfigRabbitHost	string	Адрес хоста для доступа к RabbitMQ.
MainConfigRabbitPort	string	Порт для доступа к RabbitMQ.
MainConfigRabbitUserName	string	Имя пользователя для доступа к RabbitMQ.
MainConfigRabbitPassword	string	Пароль для доступа к RabbitMQ.
MainConfigRabbitQueueName	string	Название очереди, которая используется для базового имени конкретных очередей подписок пользователей.
MainConfigRabbitDirectExchangeName	string	Название обменника RabbitMQ, который используется для направления сообщений на очереди подписок пользователей.
MainConfigRabbitAlternateExchangeName	string	Название обменника RabbitMQ, который будет использоваться по умолчанию, когда для пользователя ещё не создана подписка.
MainConfigRabbitUnroutedQueueName	string	Название очереди RabbitMQ, которая будет обрабатываться по умолчанию, когда для пользователя ещё не создана подписка.
MainConfigRabbitCapacity	1	Не меняйте это значение! Задаёт максимальное количество параллельных построений отчётов на одном рабочем построителе.
MainConfig_SecurityAdvisor_RestrictUnsafe	string	Включает или отключает ключевое слово unsafe в скрипте.

быстрыеотчеты.рф 49 / 76

ключ	тип	ОПИСАНИЕ
MainConfig_SecurityAdvisor_RestrictUnmanaged	string	Включает или отключает ключевое слово unmanaged в скрипте.
MainConfig_SecurityAdvisor_RestrictExtern	string (uri)	Включает или отключает ключевое слово extern в скрипте.
MainConfig_SecurityAdvisor_RestrictAsync	string (uri)	Включает или отключает ключевое слово async в скрипте.
MainConfig_SecurityAdvisor_RestrictTypeOf	string	Включает или отключает ключевое слово typeof в скрипте.
MainConfig_SecurityAdvisor_Whitelist_0	boolean	Задаёт список АРІ, которые можно использовать без предупреждений в скрипте отчёта. Число указывает порядковый номер в списке.
MainConfig_SecurityAdvisor_Blacklist_0	boolean	Задаёт список АРІ, которые нельзя использовать в скрипте отчёта. Число указывает порядковый номер в списке.
MainConfig_License	string	Лицензионный ключ. Предоставляется вместе с продуктом.
MainConfigSmtpServerServer	string	Адрес почтового сервера
MainConfig_SmtpServer_Port	integer	Порт SMTP сервера
MainConfigSmtpServerUsername	string	Имя пользователя SMTP сервера
MainConfig_SmtpServer_Password	string	Пароль пользователя SMTP сервера
MainConfig_SmtpServer_From	string	Почтовый адрес отправителя (отображается в письме)
MainConfigTasksAttempts	integer	Количество попыток запуска задачи воркером
MainConfigFrontendMixinsHead	string	Примесь, встраиваемая в header пользовательской панели (например, код аналитики)

быстрыеотчеты.рф 50 / 76

ключ	тип	ОПИСАНИЕ
MainConfigFrontendMixinsBody	string	Примесь, встраиваемая в body пользовательской панели (например, код аналитики)
MainConfigRfc2898CryptorSettingsSalt	string	Соль криптографического алгоритма (используется для шифрования паролей)
MainConfigRfc2898CryptorSettingsPassword	string	Пароль криптографического алгоритма
MainConfigRfc2898CryptorSettingsIV	string	Вектор криптографического алгоритма
MainConfigInvariantLocale	string	Постоянная локализация. Работает независимо от языка браузера
MainConfigDataSourcesConfigXmlConfigCommandTimeout	integer	Время ожидания ответа от источника данных XML в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigJsonConfigCommandTimeout	integer	Время ожидания ответа от источника данных JSON в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigCsvConfigCommandTimeout	integer	Время ожидания ответа от источника данных CSV в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigMySqlConfigCommandTimeout	integer	Время ожидания ответа от источника данных MySQL в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigPostgreSqlConfigCommandTimeout	integer	Время ожидания ответа от источника данных PostgreSQL в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.

быстрыеотчеты.рф 51 / 76

ключ	тип	ОПИСАНИЕ
MainConfigDataSourcesConfigMsSqlConfigCommandTimeout	integer	Время ожидания ответа от источника данных MS SQL в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigOracleDbConfigCommandTimeout	integer	Время ожидания ответа от источника данных Oracle DB в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigFirebirdConfigCommandTimeout	integer	Время ожидания ответа от источника данных Firebird в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigMongoDbConfigCommandTimeout	integer	Время ожидания ответа от источника данных Mongo DB в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
MainConfigDataSourcesConfigClickHouseConfigCommandTimeout	integer	Время ожидания ответа от источника данных ClickHouse в секундах. Если не указать значение, будет использоваться стандартный таймаут для этого источника данных.
FontServerAddress	string (uri)	Абсолютная ссылка для доступа к серверу шрифтов.

## Раздел Gateway

Позволяет конфигурировать шлюз доступа МоиОтчеты Корпоративный Сервер.

быстрыеотчеты.рф 52 / 76

```
{
  "Gateway":{
    "WhiteListForDisabled":{
        "<any_claim_name>":"<claim_value>"
     },
     "IsDisabled":false,
     "ExcludePaths":[
        "/account",
        "/disabled"
],
     "IsSignInRequired":false
}
```

ключ	тип	ОПИСАНИЕ
GatewayWhiteListForDisabled <any_claim_name></any_claim_name>	string	Список утверждений пользователя в токене для доступа к выключенному МоиОтчеты Облако.
GatewayIsDisabled	boolean	Включает или отключает доступ к МоиОтчеты Облако.
GatewayExcludePaths0	string	Список путей, к которым можно получить доступ даже при выключенном доступе к МоиОтчеты Облако. Число указывает порядковый номер в списке.
GatewaylsSignInRequired	boolean	Включает или отключает необходимость входа для пользователя для доступа к МоиОтчеты Облако.

# Раздел Services

Позволяет конфигурировать список сервисов для маршрутизации шлюза.

быстрыеотчеты.рф 53 / 76

```
{
 "Services":{
   "HealthCheckInterval":30,
   "Items":{
     "<name>":{
       "Urls":[
        "http://localhost:5555"
       ],
       "Scheme":"http",
       "Port":5555,
       "K8sServiceName":"fr-rp",
       "HostType":"WebApp",
       "PathBase":"/api/rp/swagger",
       "Namespace":"fr-cloud",
       "Type":"K8s",
       "PingPath":"/api/rp/v1/healthcheck",
       "IsSignInRequired":true,
       "Priority":10,
       "PingResponseCode":200,
       "LoadBalanceMode":"Random",
       "HealthCheckAttemptsNumber":3,
       "WhiteListClaims":{
        "<claim_name>":"<claim_value>"
       }
     }
   }
 }
}
```

ключ	тип	ОПИСАНИЕ
ServicesHealthCheckInterval	integer	Интервал для проверки работоспособности сервисов, задаётся в секундах.
Services_Items_ <name>_Urls_0</name>	string (url)	Список url адресов для доступа к статичным сервисам. Число указывает порядковый номер в списке.
ServicesItems <name>Scheme</name>	string	Схема доступа к сервису: http или https.
Services_Items <name>Port</name>	integer	Порт доступа к сервису.
Services_Items <name>K8sServiceName</name>	string	Название сервиса в Kubernetes.
Services_Items_ <name>_HostType</name>	string	Тип сервиса для обработки перенаправления шлюзом, может принимать значения: WebApp, API, External, Websocket.
Services_Items <name>PathBase</name>	string	Базовый путь для сервиса.
Services_Items <name>Namespace</name>	string	Пространство имён сервиса в Kubernetes.
Services_Items_ <name>Type</name>	string	Тип сервиса для обработки доступа шлюзом; может принимать значения: Static, K8s.

быстрыеотчеты.рф 54 / 76

ключ	тип	ОПИСАНИЕ
ServicesItems <name>PingPath</name>	string	Часть строки запроса для получения информации о состоянии сервиса.
ServicesItems <name>IsSignInRequired</name>	string	Указывает необходимость авторизации перед получением доступа к сервису пользователям.
ServicesItems <name>Priority</name>	number	Приоритет сервиса перед другими; значение меньше значит сильнее.
Services_Items_ <name>PingResponseCode</name>	integer	Статус код ответа, который будет ожидаться от healthcheck.
Services_Items_ <name>_LoadBalanceMode</name>	string	Указывает тип балансировки нагрузки для этого сервиса; может принимать одно из значений: Random, AverageMetric
Services_Items_ <name>_HealthCheckAttemptsNumber</name>	integer	Количество попыток проверить состояния сервиса.
Services_Items_ <name>_WhiteListClaims_<claim_name></claim_name></name>	string	Указывает утверждение у пользователя для получения доступа к сервису.

Bместо <name> следует использовать имя сервиса; список сервисов можно посмотреть в файле appsettings.json.

## Раздел Constants

Позволяет устанавливать ограничения на размер тела запроса в приложении.

```
{
    "Constants": {
        "LimitsMaxRequestBodySize": 2097152000
    }
}
```

ключ	тип	ОПИСАНИЕ
ConstantsLimitsMaxRequestBodySize	long	Максимальный размер тела запроса. При превышении данного ограничения запрос будет отклонен.

## Дополнительный раздел Migrations для MainConfig

Позволяет настраивать конфигурацию применения миграций.

быстрыеотчеты.рф 55 / 76

```
{
"MainConfig": {
   "Migrations": {
    "DataBaseCheckInterval": 1,
    "LockDoubleCheckInterval": 5,
    "MigrationTimeOut": 60
   }
}
```

ключ	тип	ОПИСАНИЕ
MainConfigMigrationsDataBaseCheckInterval	double	Интервал (в секундах) проверки наличия доступа к базе данных перед запуском миграций. По умолчанию: 1.
MainConfigMigrationsLockDoubleCheckInterval	double	Интервал (в секундах) для повторной проверки блокировки миграции. Может потребоваться увеличить, если используется репликасет с задержкой. По умолчанию: 5.
MainConfig_Migrations_MigrationTimeOut	double	Время ожидания выполнения миграции (в секундах), после которого происходит таймаут. По умолчанию: 60.

## Дополнительный раздел InternalHeaders для MainConfig

Раздел InternalHeaders используется для настройки внутренней авторизации между сервисами системы. Каждый сервис должен использовать уникальный ключ для идентификации при внутреннем взаимодействии. Эти ключи должны быть сгенерированы пользователем самостоятельно — использование значений из примера недопустимо в целях безопасности.

быстрыеотчеты.рф 56 / 76

```
{
"Designer": {
 "BackendUrl": "http://fr-backend.fr-cloud:80",
 "InternalKey": "fc51a5d6-95c3-4679-8c03-c9ac6536bf5d"
 },
 "Fonts": {
 "BackendUrl": "http://fr-backend.fr-cloud:80",
 "InternalKey": "22788120-f7cd-43f2-938b-327d6c3ceed4"
 "Scheduler": {
 "BackendUrl": "http://fr-backend.fr-cloud:80",
 "InternalKey": "9da6cb50-8796-42cc-b3a7-53c1761f0ac9"
 "WorkerCore": {
 "BackendUrl": "http://fr-backend.fr-cloud:80",
 "InternalKey": "ee444483-e0f1-4ec2-9b4e-5cbacdfaa07f"
 },
 "Gateway": {
 "BackendUrl": "http://fr-backend.fr-cloud:80",
 "InternalKey": "4632ab6a-3e20-4430-9f13-2fe1851809db"
},
"MainConfig": {
 "InternalHeaders": {
  "ee444483-e0f1-4ec2-9b4e-5cbacdfaa07f": "00000000000000000000001",
  "9da6cb50-8796-42cc-b3a7-53c1761f0ac9": "00000000000000000000004",
  }
}
}
```

ключ	тип	ОПИСАНИЕ
Designer_BackendUrl	string	URL бэкенда сервиса Designer.
DesignerInternalKey	string	Уникальный ключ для внутренней авторизации Designer.
FontsBackendUrl	string	URL бэкенда сервиса Fonts.
FontsInternalKey	string	Уникальный ключ для внутренней авторизации Fonts.
Scheduler_BackendUrl	string	URL бэкенда сервиса Scheduler.
SchedulerInternalKey	string	Уникальный ключ для внутренней авторизации Scheduler.
WorkerCoreBackendUrl	string	URL бэкенда сервиса WorkerCore.
WorkerCoreInternalKey	string	Уникальный ключ для внутренней авторизации WorkerCore.
GatewayBackendUrl	string	URL бэкенда сервиса Gateway.
GatewayInternalKey	string	Уникальный ключ для внутренней авторизации Gateway.
MainConfigInternalHeaders<ключ>	string	Сопоставление InternalKey — внутреннего идентификатора сервиса.

быстрыеотчеты.рф 57 / 76

**Важно**: ключи InternalKey и соответствующие им значения в InternalHeaders должны быть уникальными и сгенерированы пользователем. Использование значений из примера недопустимо.

#### Рекомендации по генерации ключей

- Ключи могут быть представлены в формате GUID или произвольной сложной строкой.
- Рекомендуется использовать криптографически безопасные генераторы случайных строк.
- Ключи должны быть достаточно длинными и уникальными, чтобы исключить возможность коллизий или подбора.

быстрыеотчеты.рф 58 / 76

# Удаление МоиОтчеты Корпоративный Сервер из кластера

Продукты: МоиОтчеты Корпоративный Сервер

1. Удаление компонентов сервера отчётов из кластера:

kubectl delete -n fr-corporate-server deployment fr-designer kubectl delete -n fr-corporate-server deployment fr-scheduler kubectl delete -n fr-corporate-server deployment fr-workercore kubectl delete -n fr-corporate-server deployment fr-gateway kubectl delete -n fr-corporate-server deployment fr-app kubectl delete -n fr-corporate-server deployment fr-backend kubectl delete -n fr-corporate-server deployment fr-s-preview kubectl delete -n fr-corporate-server deployment fr-fonts kubectl delete -n fr-corporate-server deployment fr-homer kubectl delete -n fr-corporate-server secret corporate-volume-secret

2. Удаление MongoDB или FerretDB.

Удаление MongoDB (FerretDB) приведёт к удалению пользователей, групп, шаблонов отчётов, подготовленных отчётов и экспортированных отчётов.

kubectl delete -n mongo deployment fr-mongo-mongodb

или

kubectl delete -n ferret deployment fr-ferret

Обратите внимание - в зависимости от настроек, вышеприведённая команда может не удалять дисковое пространство, арендованное у провайдеров облачных сервисов. Если вы арендуете кластер у облачного провайдера, воспользуйтесь web-интерфейсом провайдера для удаления Persistent Volumes сервера отчётов - mongo-storage.

- 3. Удаление остальных компонентов кластера необязательно, а в случае, если сервер был использован в рабочем кластере совместно со сторонними сервисами, то удаление этих компонентов может повредить инфраструктуру. Например, RabbitMQ может быть использован сторонними сервисами кластера.
- 4. Для полного удаления кластера и его компонентов используйте команду:

kubeadm reset

быстрыеотчеты.рф 59 / 76

# Разворачивание Корпоративного Сервера или Публикатора без среды Kubernetes

Продукты: МоиОтчеты Корпоративный Сервер, МоиОтчеты Публикатор

Помимо разворачивания в Kubernetes Корпоративный Сервер или Публикатор могут быть запущены с помощью docker-compose.

Для этого необходимо выполнить следующие действия.

### Хранилище данных с открытым исходным кодом

На данной странице рассказывается о варианте установки с использованием MongoDB. Альтернативой выступает FerretDB - аналог с открытым исходным ходом, который использует PostgreSQL как движок базы данных. Инструкцию по установке с использованием FerretDB можно прочитать по ссылке.

## Указываем лицензию в файл appsettings. Production. json

Создаём файл с названием appsettings. Production. json и указываем в нём лицензионный ключ.

```
{
           "Auth": {
             "UseOpenId": false,
              "UseLocal": true
          "MainConfig": {
             "InternalHeaders": {
                 "S56nHMSzjQzXYx5KJJsU3cjU": "000000000000000000000001",
                 "QNpq2nyzvDNSCBLtVhMJ9e8m": "00000000000000000000003",
                 },
              "Frontend": {
                 "Mixins": {
                    "Head": ""
                    "Bodv": ""
                 "InvariantLocale": ""
              },
              "License": "",
              "Server": {
                 "Title": "МоиОтчеты Сервисные решения",
                 "CorporateServerMode": true
               "Rabbit": {
                 "Host": "rabbitmg",
                 "Port": 5672,
                 "UserName": "fastreports",
                 "Password": "Qwerty!23456",
                 "QueueName": "ReportProcessQueue",
                 "DirectExchangeName": "DirectEx",
                 "AlternateExchangeName": "AExchange",
                 "UnroutedQueueName": "Default",
                 "Capacity": 1
                 "ConnectionString": "mongodb://fastreport:Qwerty!23456@mongo:27017/?
auth Source = Report Store \& read Preference = primary \& appname = MongoDB \% 20 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 \& wait Quality = 100 Compass \& ssl = false \& maxPool Size = 100 Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool Size = 100 E Compass \& ssl = false \& maxPool S
ueueMultiple=100",
```

"DatabacoNamo": "BoportStoro"

```
Databaseivaille . Nepultatule
 }
},
"Gateway": {
 "BackendUrl": "http://fr-backend:80",
 "InternalKey": "QNpq2nyzvDNSCBLtVhMJ9e8m",
 "SignInPagePath": "/account/signin?r={0}",
 "MaxConcurrentRequests": 200,
 "RequestQueueLimit": 5000
},
"Serilog": {
 "MinimumLevel": {
  "Default": "Debug"
 }
},
"Services": {
 "Items": {
  "OnlineDesigner": {
   "Type": "Static",
   "Urls": [
    "http://fr-onlinedesigner:80"
   ]
  },
  "Backend": {
   "Type": "Static",
   "Urls": [
    "http://fr-backend:80"
  },
  "FrontendApp": {
   "Type": "Static",
   "Urls": [
    "http://fr-app:80"
   ]
  },
  "Fonts": {
   "Type": "Static",
   "Urls": [
    "http://fr-fonts:80"
   ]
  },
  "StaticPreviewApp": {
   "Type": "Static",
   "Urls": [
     "http://fr-staticpreview:80"
   ]
  },
  "HomerApp": {
   "Type": "Static",
   "Urls": [
    "http://fr-homer:80"
   ]
  }
 }
},
"Designer": {
 "BackendUrl": "http://fr-backend:80",
 "InternalKey": "x2aHtuSsxFeYqE8xPTaxAnbH"
},
"WorkerCore": {
 "BackendUrl": "http://fr-backend:80",
 "InternalKey": "S56nHMSzjQzXYx5KJJsU3cjU"
},
"Scheduler": {
 "BackendUrl": "http://fr-backend:80",
 "InternalKey": "9MXgeFwLNjeUrJvw7N2aQv9F"
```

быстрыеотчеты.рф 61 / 76

```
}
}
```

Находим поле License и указываем ключ:

```
...
"License": "{ключ из файла лицензии}"
...
```

## Настраиваем базу данных

Создаём файл mongo-init.js со следующим содержимым:

```
db = db.getSiblingDB('admin')

db.auth('admin', 'Qwerty!23456')

db = db.getSiblingDB('ReportStore')

db.createUser({
  user: "fastreport",
  pwd: "Qwerty!23456",
  roles: [ { role: "readWrite", db: "ReportStore" } ]
});
```

## Файл docker compose

Создаём файл с именем docker-compose.yml со следующим содержимым:

Для МоиОтчеты Корпоративный Сервер:

```
version: "3.9"
services:
mongo:
  image: mongo:5.0
  volumes:
   - mongo:/data/db
   - ./mongo-init.js:/docker-entrypoint-initdb.d/mongo-init.js:ro
  restart: always
  environment:
   - MONGO_INITDB_ROOT_USERNAME=admin
   - MONGO INITDB ROOT PASSWORD=Qwerty!23456
   - MONGO_INITDB_DATABASE=ReportStore
  networks:
   - fr-cs
 rabbitmq:
  image: bitnami/rabbitmq:3.9
  volumes:
   - rabbitmq:/bitnami
  restart: always
  networks:
   - fr-cs
  environment:
   - RABBITMQ_USERNAME=fastreports
   - RABBITMQ_PASSWORD=Qwerty!23456
 fr-backend:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
backend:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
```

быстрыеотчеты.рф 62 / 76

```
networks:
   - fr-cs
 fr-gateway:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
gateway:debian-2025.2.10
  ports:
   - 8080:80
  restart: always
   environment:
#
    - SomeEnviromentSettings=value
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-fonts:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
fonts:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-app:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
app:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-staticpreview:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
static-preview:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-onlinedesigner:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
designer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-homer:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdq.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
homer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-workercore:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
workercore:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
  deploy:
   mode: replicated
   replicas: 2
```

быстрыеотчеты.рф

#### Для МоиОтчеты Публикатор:

```
version: "3.9"
services:
mongo:
  image: mongo:5.0
  volumes:
   - mongo:/data/db
   - ./mongo-init.js:/docker-entrypoint-initdb.d/mongo-init.js:ro
  restart: always
  environment:
   - MONGO INITDB ROOT USERNAME=admin
   - MONGO_INITDB_ROOT_PASSWORD=Qwerty!23456
   - MONGO_INITDB_DATABASE=ReportStore
  networks:
   - fr-cs
 rabbitmq:
  image: bitnami/rabbitmq:3.9
  volumes:
   - rabbitmq:/bitnami
  restart: always
  networks:
   - fr-cs
  environment:
   - RABBITMQ_USERNAME=fastreports
   - RABBITMQ_PASSWORD=Qwerty!23456
fr-backend:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
backend:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-gateway:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
gateway:debian-2025.2.10
  ports:
   - 8080:80
  restart: always
   environment:
    - SomeEnviromentSettings=value
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-fonts:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-fonts:debian-
```

быстрыеотчеты.рф 64 / 76

```
ZUZD.Z.1U
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-app:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-app:debian-
2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-staticpreview:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-static-
preview:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-onlinedesigner:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
designer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-homer:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdq.xn--p1ai/repository/docker-registry/moiotchety-publisher-
homer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-workercore:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
workercore:debian-2025.2.10
  restart: always
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
  deploy:
   mode: replicated
   replicas: 2
   endpoint mode: vip
fr-scheduler:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
scheduler:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
networks:
fr-cs:
  driver: bridge
volumes:
 mongo:
 rabbitmq:
```

быстрыеотчеты.рф 65 / 76

## Запускаем сервер

По итогу в докере должны быть загружены скачанные образы, а в папке лежать файлы appsettings. Production. json, docker-compose. yml и mongo-init. js.

Запускаем сервер:

docker-compose up --build -d

## Вход в систему

После выполнения всех вышеописанных скриптов установка сервера отчётов закончена. Для того чтобы проверить успешность установки и начать работу, зайдите с помощью браузера на сервер отчётов по адресу localhost:8080 и в появившемся окне введите следующие данные:

Имя пользователя: admin@example.com

Пароль: admin

Чтобы попасть в панель администратора перейдите по ссылке localhost:8080/homer

## Выключаем сервер

docker-compose down

быстрыеотчеты.рф 66 / 76

# Разворачивание Корпоративного Сервера или Публикатора без среды Kubernetes

Продукты: МоиОтчеты Корпоративный Сервер, МоиОтчеты Публикатор

Помимо разворачивания в Kubernetes Корпоративный Сервер или Публикатор могут быть запущены с помощью docker-compose.

Для этого необходимо выполнить следующие действия.

## MongoDB как хранилище данных

На данной странице рассказывается о варианте установки с использованием FerretDB - хранилище с открытым исходным ходом, которое использует PostgreSQL как движок базы данных. Альтернативно вы можете установить MongoDB. Инструкцию по установке с использованием MongoDB можно прочитать по ссылке.

## Разница с MongoDB

Вы не почуствуете значительных изменений и сможете общаться с базой аналогично MongoDB (в том числе используя MongoDB Compass и утилиту mongosh). Единственное значительное различие - FerretDB не поддерживает роли, поэтому создание новых пользователей базы данных должно происходить следующим образом:

```
db.createUser({
  user: 'newuser',
  pwd: 'newpassword',
  roles: []
})
```

## Указываем лицензию в файл appsettings.Production.json

Создаём файл с названием appsettings. Production. json и указываем в нём лицензионный ключ.

```
"Auth": {
 "UseOpenId": false,
 "UseLocal": true
},
"MainConfig": {
 "InternalHeaders": {
 "S56nHMSzjQzXYx5KJJsU3cjU": "000000000000000000000001",
 "QNpq2nyzvDNSCBLtVhMJ9e8m": "00000000000000000000003",
 },
 "Frontend": {
 "Mixins": {
  "Head": "",
  "Body": ""
 },
 "InvariantLocale": ""
 "License": "",
 "Server": {
 "Title": "МоиОтчеты Сервисные решения",
 "CorporateServerMode": true
```

быстрыеотчеты.рф 67 / 76

```
"Rabbit": {
               "Host": "rabbitmq",
               "Port": 5672,
               "UserName": "fastreports",
               "Password": "Qwerty!23456",
                "QueueName": "ReportProcessQueue",
               "DirectExchangeName": "DirectEx",
                "AlternateExchangeName": "AExchange",
               "UnroutedQueueName": "Default",
               "Capacity": 1
            },
             "Database": {
                "ConnectionString": "mongodb://fastreport:Qwerty!23456@ferret:27017/?
auth Source = Report Store \& read Preference = primary \& appname = MongoDB \% 20 Compass \& ssl = false \& maxPool Size = 100 \& waitQuestion = 100 Label MaxPool Size = 100 
ueueMultiple=100",
               "DatabaseName": "ReportStore"
            }
         },
         "Gateway": {
            "BackendUrl": "http://fr-backend:80",
            "InternalKey": "QNpq2nyzvDNSCBLtVhMJ9e8m",
            "SignInPagePath": "/account/signin?r={0}",
            "MaxConcurrentRequests": 200,
            "RequestQueueLimit": 5000
         },
         "Serilog": {
            "MinimumLevel": {
               "Default": "Debug"
            }
         },
          "Services": {
            "Items": {
               "OnlineDesigner": {
                  "Type": "Static",
                   "Urls": [
                     "http://fr-onlinedesigner:80"
                  ]
               },
                "Backend": {
                   "Type": "Static",
                   "Urls": [
                     "http://fr-backend:80"
                  ]
                },
                "FrontendApp": {
                   "Type": "Static",
                   "Urls": [
                     "http://fr-app:80"
                  1
                },
                "Fonts": {
                   "Type": "Static",
                   "Urls": [
                      "http://fr-fonts:80"
                },
                "StaticPreviewApp": {
                   "Type": "Static",
                   "Urls": [
                     "http://fr-staticpreview:80"
                  ]
                "HomerApp": {
                   "Type": "Static",
                   "Hrle" · [
```

быстрыеотчеты.рф 68 / 76

```
U113 . [
      "http://fr-homer:80"
   }
  }
 },
 "Designer": {
  "BackendUrl": "http://fr-backend:80",
  "InternalKey": "x2aHtuSsxFeYqE8xPTaxAnbH"
 },
 "WorkerCore": {
  "BackendUrl": "http://fr-backend:80",
  "InternalKey": "S56nHMSzjQzXYx5KJJsU3cjU"
 },
 "Scheduler": {
  "BackendUrl": "http://fr-backend:80",
  "InternalKey": "9MXgeFwLNjeUrJvw7N2aQv9F"
 }
}
```

Находим поле License и указываем ключ:

```
...
"License": "{ключ из файла лицензии}"
...
```

## Файл docker compose

Создаём файл с именем docker-compose.yml со следующим содержимым:

Для МоиОтчеты Корпоративный Сервер:

```
version: "3.9"
services:
 ferret:
  image: ghcr.io/ferretdb/ferretdb-eval:2
  volumes:
   - ferret:/var/lib/postgresql/data
  restart: always
  environment:
   - POSTGRES_USER=fastreport
   - POSTGRES_PASSWORD=Qwerty!23456
  networks:
   - fr-cs
 rabbitmq:
  image: bitnami/rabbitmq:3.9
  volumes:
   - rabbitmq:/bitnami
  restart: always
  networks:
   - fr-cs
  environment:
   - RABBITMQ USERNAME=fastreports
   - RABBITMQ PASSWORD=Qwerty!23456
 fr-backend:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
backend:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-gateway:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
```

быстрыеотчеты.рф 69 / 76

```
gateway:debian-2025.2.10
  ports:
   - 8080:80
  restart: always
  environment:
    - SomeEnviromentSettings=value
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-fonts:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
fonts:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-app:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
app:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-staticpreview:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
static-preview:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-onlinedesigner:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
designer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-homer:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
homer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-workercore:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdq.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
workercore:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
  deploy:
   mode: replicated
   replicas: 2
   endpoint_mode: vip
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-corporate-server-
scheduler:debian-2025.2.10
  roctart, always
```

быстрыеотчеты.рф 70 / 76

```
volumes:
- ./appsettings.Production.json:/app/appsettings.Production.json:ro
networks:
- fr-cs
networks:
fr-cs:
driver: bridge
volumes:
ferret:
rabbitmq:
```

#### Для МоиОтчеты Публикатор:

```
version: "3.9"
services:
ferret:
  image: ghcr.io/ferretdb/ferretdb-eval:2
  volumes:
   - ferret:/var/lib/postgresql/data
  restart: always
  environment:
   - POSTGRES_USER=fastreport
   - POSTGRES_PASSWORD=Qwerty!23456
  networks:
   - fr-cs
 rabbitmq:
  image: bitnami/rabbitmq:3.9
  volumes:
   - rabbitmq:/bitnami
  restart: always
  networks:
   - fr-cs
  environment:
   - RABBITMQ_USERNAME=fastreports
   - RABBITMQ PASSWORD=Qwerty!23456
 fr-backend:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
backend:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-gateway:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
gateway:debian-2025.2.10
  ports:
   - 8080:80
  restart: always
   environment:
#
    - SomeEnviromentSettings=value
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
fr-fonts:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-fonts:debian-
2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
```

быстрыеотчеты.рф 71 / 76

```
image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-app:debian-
2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
   - fr-cs
fr-staticpreview:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-static-
preview:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-onlinedesigner:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
designer:debian-2025.2.10
  restart: always
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
 fr-homer:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
homer:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
   - fr-cs
 fr-workercore:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
workercore:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
   - fr-cs
  deploy:
   mode: replicated
   replicas: 2
   endpoint mode: vip
fr-scheduler:
  image: xn--80akiaokt3b4b.xn--90aia9aifhdb2cxbdg.xn--p1ai/repository/docker-registry/moiotchety-publisher-
scheduler:debian-2025.2.10
  restart: always
  volumes:
   - ./appsettings.Production.json:/app/appsettings.Production.json:ro
  networks:
   - fr-cs
networks:
 fr-cs:
  driver: bridge
volumes:
 ferret:
 rabbitmq:
```

## Запускаем сервер

По итогу в докере должны быть загружены скачанные образы, а в папке лежать файлы appsettings. Production. json и docker-compose.yml.

быстрыеотчеты.рф 72 / 76

Запускаем сервер:

docker-compose up --build -d

## Вход в систему

После выполнения всех вышеописанных скриптов установка сервера отчётов закончена. Для того чтобы проверить успешность установки и начать работу, зайдите с помощью браузера на сервер отчётов по адресу localhost:8080 и в появившемся окне введите следующие данные:

Имя пользователя: admin@example.com

Пароль: admin

Чтобы попасть в панель администратора перейдите по ссылке localhost:8080/homer

## Выключаем сервер

docker-compose down

быстрыеотчеты.рф 73 / 76

# Диагностика и устранение неполадок

Продукты: МоиОтчеты Корпоративный Сервер

Если при установке или запуске приложения возникли проблемы, данный раздел поможет вам в их устранении.

## Миграции

Корпоративный сервер предусматривает автоматическое применение миграций при запуске приложения. Ручное управление миграциями, как правило, не предусмотрено для конечного пользователя.

Хотя сервер имеет встроенный механизм восстановления после сбоев, определённые проблемы могут быть вызваны внешними факторами — например, отключением питания или потерей соединения с базой данных.

Если во время выполнения миграций произойдёт разрыв соединения с базой данных, система активирует режим блокировки и предотвратит запуск бэкенда для защиты целостности базы данных. В таком случае потребуется ручное восстановление состояния миграций.

**Важно:** Рекомендуется всегда создавать резервную копию базы данных перед обновлением версии корпоративного сервера. Это позволит оперативно восстановить работоспособность системы в случае возникновения ошибок.

Если резервная копия отсутствует, можно воспользоваться инструментами разработчика. Следует учитывать, что использование этих инструментов может привести к потере данных, однако они позволяют восстановить целостность базы данных и обеспечить стабильную работу сервера.

Для ознакомления с инструментами разработчика перейдите в соответствующий раздел.

## Инструменты разработчика

Стандартная поставка корпоративного сервера поддерживает развертывание с использованием docker-compose и Kubernetes. В данной документации рассматриваются только эти два способа развертывания.

Если у вас возникли сложности или требуется дополнительная помощь, вы можете обратиться в нашу службу технической поддержки.

#### **Docker**

Чтобы использовать инструменты разработчика, необходимо временно остановить работающий экземпляр бэкенда. Выполните следующие команды:

1. Проверка состояния контейнеров

Получите список запущенных контейнеров:

docker-compose -f /путь/к/папке/docker-compose.yml ps

Обратите внимание: для выполнения всех команд необходим файл docker-compose.yml. Убедитесь, что текущая директория содержит этот файл, либо указывайте путь к нему с помощью параметра -f

2. Остановка контейнера бэкенда

быстрыеотчеты.рф 74 / 76

Остановите нужный контейнер (например, fr-backend):

docker-compose -f /путь/к/папке/docker-compose.yml stop fr-backend

3. Запуск контейнера с инструментами разработчика

Для работы с инструментами разработчика создайте новый контейнер и подключите его к той же сети, которая используется в docker-compose.yml.

Сначала проверьте доступные сети:

docker network Is

Затем выполните команду запуска контейнера. Не забудьте заменить debian-2025.2.0 на актуальный тег, который указан в вашем файле docker-compose.yml:

docker run -it --network corporate\_fr-cs -v /путь/к/папке/appsettings.Production.json:/app/appsettings.Production.json fastreport-corporate-server-backend:debian-2025.2.0 /bin/bash

После входа в контейнер запустите утилиту:

dotnet FastReport.Cloud.Backend.dll -- --corporate-dev-tools

Следуйте инструкциям, отображаемым в терминале. После завершения работы выйдите из контейнера:

exit

4. Перезапуск основного сервиса

После использования инструментов разработчика перезапустите бэкенд:

docker-compose -f /путь/к/папке/docker-compose.yml start fr-backend

#### **Kubernetes**

Для использования инструментов разработчика необходимо временно остановить работающий экземпляр бэкенд-сервиса. Выполните следующие шаги:

1. Проверка состояния deployment

Получите список запущенных deployment-объектов. Обратите внимание на количество активных реплик — это значение потребуется на следующем этапе:

kubectl get deployments --namespace fr-corporate

Важно: укажите корректное пространство имён. В данном примере используется fr-corporate.

2. Остановка подов путём масштабирования до нуля реплик

Установите количество реплик равным нулю, чтобы остановить соответствующие поды:

kubectl scale --replicas=0 deployment/fr-backend --namespace fr-corporate

3. Запуск контейнера с инструментами разработчика

Создайте файл my-debug-pod.yml со следующим содержимым:

быстрыеотчеты.рф 75 / 76

apiVersion: v1 kind: Pod metadata:

name: my-debug-pod namespace: fr-corporate

spec: containers: - name: app

image: fastreport-corporate-server-backend:debian-2025.2.0

command: ["/bin/bash"]

args: ["-c", "while true; do sleep 10; done"]

tty: true volumeMounts:

- mountPath: /app/appsettings.Production.json

name: config-volume

subPath: appsettings.Production.json

volumes:

- name: config-volume

configMap:

name: fast-report-config

items:

 key: appsettings.Production.json path: appsettings.Production.json

defaultMode: 420

#### Примените манифест:

kubectl apply -f my-debug-pod.yml --namespace fr-corporate

4. Подключение к поду

Подключитесь к созданному поду:

kubectl exec -it my-debug-pod --namespace fr-corporate -- /bin/bash

Запустите утилиту разработчика:

dotnet FastReport.Cloud.Backend.dll -- --corporate-dev-tools

Следуйте инструкциям в терминале. По завершении работы выйдите из контейнера:

exit

5. Удаление отладочного пода

После выполнения всех операций удалите временный под:

kubectl delete pod my-debug-pod --namespace fr-corporate

6. Восстановление количества реплик

Верните deployment к исходному количеству реплик (например, 1):

 ${\it kubectl\ scale\ -- replicas = 1\ deployment/fr-backend\ -- namespace\ fr-corporate}$ 

быстрыеотчеты.рф 76 / 76