

# Инфраструктура интеграционного решения

- Компоненты инфраструктуры
- Параметры виртуальных машин
- Развёртывание и настройка nginx
- Развёртывание и настройка web-сервисов Domino
- Развёртывание и настройка планировщиков Domino
- Установка обновлений
- Масштабирование решения
- Тестовый контур

# Компоненты инфраструктуры

Для интеграционного решения предполагается использовать инфраструктуру из виртуальных машин, так как она обеспечивает большую гибкость на начальном этапе. Однако при необходимости эта же схема может быть развёрнута на физических машинах.

В качестве единой точки доступа будет использован web-сервер nginx. Его задачи:

- предоставление единой точки доступа ко всем web- сервисам интеграционного решения
- поддержка протокола https, работа с SSL- сертификатами
- фильтрация входящего трафика по ip для предотвращения DDOS атак
- проксирование входящих запросов на соответствующие web- сервисы Domino
- балансировка нагрузки в том случае, когда web- сервис Domino реализован в виде пула из нескольких серверов

Web-сервис (хук) для обработки входящих уведомлений от СМ. Принимает и обрабатывает уведомления о создании, изменении состояния и отмене заказа со стороны СМ. Реализуется в виде Domino web-сервера. Для обеспечения достаточной производительности развёртывается пул из 4 однородных web-серверов с балансировкой нагрузки через nginx.

Web-сервис для сборки заказов СМ. Обрабатывает запросы от приложений сборки заказов СМ, изменяет состояние и состав заказов и отправляет уведомления (нотификации) в СМ. Реализуется в виде Domino web-сервера. Для обеспечения достаточной производительности развёртывается пул из 4 однородных web-серверов с балансировкой нагрузки через nginx.

Сервис мониторинга заказов СМ. Контролирует качество процесса обработки заказов СМ на основании метрик, и направляет через телеграмм-бот уведомления обо всех обнаруженных проблемах. Реализуется в виде процедуры планировщика, которая запускается с небольшим интервалом (один раз в минуту).

Сервис формирования реализации по заказам СМ. Формирует документы реализации по отработанным (доставленным) заказам СМ. Реализуется в виде процедуры планировщика, которая запускается каждый час.

Сервис отправки телеграмм-уведомлений. В автоматическом режиме выбирает сообщения из очереди на отправку, отправляет их в соответствующие чаты телеграмм и переносит сообщения в архив. Реализуется в виде процедуры планировщика, которая запускается с небольшим интервалом (один раз в 10 секунд).

# Параметры виртуальных машин

Для развёртывания интеграционного решения используются две виртуальные машины

1. ВМ для установки http- сервера nginx

Параметры ВМ:

- ОС Linux (CentOS 7)

...

2. ВМ для развёртывания web-сервисов и планировщиков Domino

Параметры ВМ:

- ОС MS Windows Server2019
- память - 4Гб для самого сервера + 400Мб на каждый экземпляр Domino
- число процессорных ядер - одно ядро на 4 экземпляра Domino, но не менее 2 ядер
- дисковое пространство 100Гб.

# Развёртывание и настройка nginx

После установки linux следует отключить систему безопасности SELinux.

В противном случае возникает масса проблем с настройками nginx, когда отклоняются те или иные операции. Можно также отключить SE только для домена httpd\_t

```
“ semanage permissive -a httpd_t
```

Информацию о настройках безопасности nginx можно найти  
здесь: <https://disnetern.ru/nginx-server-configure-and-protect/>

Если после настройки прокси в nginx.conf location ... { proxy\_pass домино\_web\_сервер} получаем ошибку 502 Bad Gateway, а в логах nginx видим (13: Permission denied) while connecting to upstream, то нужно разрешить в SELinux upstream http соединения:

```
“ setsebool -P httpd_can_network_connect 1
```

Для того, чтобы при включённом SELinux увеличить число файлов на рабочий процесс (worker\_rlimit\_nofile) нужно разрешить эту операцию:

```
“ setsebool -P httpd_setrlimit 1
```

В настройках /etc/nginx/nginx.conf указывается:

## Контекст main

```
“ # Увеличиваем общее число файлов для рабочих процессов. Так как мы
    работаем
    # в режиме проху, то минимум число входящих соединений x2 + резерв
    worker_rlimit_nofile 65535;

    events {
        # Увеличиваем число соединений на рабочий процесс
        worker_connections 4096;
        # Включаем возможность одновременного приема множества
        подключений
        multi_accept on;
    }
```

## Контекст http

```
“ # Включаем буферизацию логов доступа

    # В продуктиве логи доступа лучше вообще отключить
    access_log ... buffer=32k flush=10s;

    # Пул web- серверов Домино для обработки уведомлений СМ по заказам
    (хук)
    upstream SberMarketOrdersHook {
        server <ip web- сервера Домино>:8081;
        server <ip web- сервера Домино>:8082;
        server <ip web- сервера Домино>:8083;
        server <ip web- сервера Домино>:8094;
    }

    # Пул web- серверов Домино для реализации процесса сборки заказов (api)
    upstream SberMarketOrdersApi {
```

```
server <ip web- сервера Домино>:8091;  
server <ip web- сервера Домино>:8092;  
server <ip web- сервера Домино>:8093;  
server <ip web- сервера Домино>:8094;  
}
```

#### Контекст server

```
“ # Локации для перенаправления запросов  
location /sbermarket/orders/hook/  
  
    # Проксируем запросы на пул SberMarketOrdersHook по методу round-  
    robin  
    proxy_pass http://SberMarketOrdersHook;  
  
    # Ставим ограничение по ip CM (не применять на тестовых  
    конфигурациях)  
  
    allow 84.201.148.122;  
  
    allow 178.154.232.168;  
  
    allow 51.250.5.205;  
  
    deny all;  
}  
  
location /sbermarket/orders/api/  
  
    # Проксируем запросы на пул SberMarketOrdersApi по методу round-robin  
    proxy_pass http://SberMarketOrdersApi;  
}
```

Обязательно на сервере nginx следует отключить протокол http и настроить протокол

https.



# Развёртывание и настройка web-сервисов Домино

В настройках Windows Server firewall нужно разрешить входящие соединения по портам 8080-8099. Если количество web- серверов Домино превысит 20, то нужно соответственно расширить диапазон портов.

Для нормальной работы экземпляров Домино, работающих в режиме сервиса нужно резервировать примерно 400Мб оперативной памяти сервера на экземпляр.

## Структура каталогов

Корневой каталог для всех web-сервисов - /Domino/SberMarket/WebServices. Каждый сервис развёртывается изолированно в своём собственном каталоге (бины, проект). Такая схема позволит обновлять сервисы по отдельности, без перезапуска всей инфраструктуры.

“ /Domino

/SberMarket

/WebServices

/OrdersHook

/Bin

/Project

```
/Log
```

```
startService.bat
```

```
stopService.bat
```

```
restartService.bat
```

```
/OrdersApi
```

```
/Bin
```

```
/Project
```

```
/Log
```

```
startService.bat
```

```
stopService.bat
```

```
restartService.bat
```

## Параметры запуска приложения

Web-сервис (хук) для обработки входящих уведомлений от СМ. Четыре экземпляра приложения, прослушивающих порты с 8081 по 8084.

```
“ C:\Domino\SberMarket\WebServices\OrdersHook\Bin\domino8.exe
```

```
C:\Domino\SberMarket\WebServices\OrdersHook\Project\RETAIL-STORE
```

```
/SERVER
```

```
LISTEN=http://<ip web- сервера Domino>:8081/
```

```
DBSERVER=<Ссылка на tnsnames.ora>
```

```
SCHEME=<Имя схемы oracle (БД Домино)>
```

```
USERNAME=SberMarketOrdersHook
```

```
TOKEN=81
```

Web-сервис для сборки заказов СМ. Четыре экземпляра приложения, прослушивающих порты с 8091 по 8094.

```
“ C:\Domino\SberMarket\WebServices\OrdersApi\Bin\domino8.exe
```

```
C:\Domino\SberMarket\WebServices\OrdersApi\Project\RETAIL-STORE
```

```
/SERVER
```

```
LISTEN=http://<ip web- сервера Домино>:8091/
```

```
DBSERVER=<Ссылка на tnsnames.ora>
```

```
SCHEME=<Имя схемы oracle (БД Домино)>
```

```
USERNAME=SberMarketOrdersApi
```

```
TOKEN=91
```

## Автозапуск сервисов при старте сервера

Через задание с однократным запуском.

## Ручной останов и запуск сервиса

Батники start/stop/restartService.

# Развёртывание и настройка планировщиков Домино

## Структура каталогов

Корневой каталог для всех планировщиков - /Domino/SberMarket/Schedulers. Каждый сервис разворачивается изолированно в своем собственном каталоге (бины, проект). Такая схема позволит обновлять сервисы по отдельности, без перезапуска всей инфраструктуры.

“ /Domino

/SberMarket

/Schedulers

/QualityMonitor

/Bin

/Project

/Log

startScheduler.bat

stopScheduler.bat

restartScheduler.bat

/OrdersSales

/Bin

/Project

/Log

startScheduler.bat

stopScheduler.bat

restartScheduler.bat

/TelegramNotifcations

/Bin

/Project

/Log

startScheduler.bat

stopScheduler.bat

restartScheduler.bat

## Параметры запуска приложения

Авто- запуск сервисов при старте сервера

## Ручной останов и запуск сервиса

# Установка обновлений

Как устанавливать обновления при работающих сервисах. Обновление проекта. Обновление ядра.

Вариант реализации для web-сервиса: развернуть второй пул процессов Domino на отдельных бинах и проекте. На той же самой VM (другое подмножество портов), или на другой VM. Описать пул в `nginx.conf` и заменить ссылку в разделе `location`. Выполнить переключение, перезагрузив конфигурацию `nginx`.

# Масштабирование решения

Делается через запуск дополнительных экземпляров web- сервера Domino. Их можно запускать на той же самой VM, либо развернуть еще одну. Сервисы - планировщики не масштабируются, однако они спроектированы так, чтобы их производительность не была критически важна.

# Тестовый контур

Тестовый контур разворачивается полностью идентично продуктивному. Для тестирования используется копия ЦБД клиента. Адреса web- сервисов для обработки входящих уведомлений от СМ и для сборки заказов СМ должны отличаться от адресов продуктивного контура.

Пример формирования адресов:

Продуктивный контур

“ <https://bim.domino.ru/sbermarket/orders>

Тестовый контур

“ <https://test-bim.domino.ru/sbermarket/orders>