



МойОфис Почта 3

Сервисно-ресурсная модель

ПОЧТОВАЯ СИСТЕМА

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«МОЙОФИС ПОЧТА 3»

СЕРВИСНО-РЕСУРСНАЯ МОДЕЛЬ

3.0

Версия 1

На 16 листах

Дата публикации: 18.06.2024

Москва

2024

МойОфис

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

1	Общие сведения	7
1.1	Введение	7
1.2	Влияния Сервисов на Подсистемы	7
2	Диаграмма CPM Почта (PSN)	8
3	Методы контроля сервисов	9
3.1	Подсистема проксирования	10
3.1.1	Методика контроля Proxy Service	10
3.2	Подсистема база учетных записей пользователей	11
3.2.1	Методика контроля LDAP Service	11
3.3	Подсистема база данных	11
3.3.1	Методика контроля EtcD Service	11
3.3.2	Методика контроля Postgresql Service	12
3.3.3	Методика контроля Pgbouncer Service	12
3.3.4	Методика контроля Redis Service	12
3.4	Подсистема почтового ядра	12
3.4.1	Методика контроля Postfix Service	13
3.4.2	Методика контроля Dovecot Service	13
3.4.3	Методика контроля Rspamd Service	13
3.5	Подсистема веб-интерфейса пользователя	13
3.5.1	Методика контроля Web Mail Service	14
3.5.2	Методика контроля Web Calendar Service	14
3.5.3	Методика контроля Web Contacts Service	14
3.5.4	Методика контроля Web Admin Service	14
3.6	Подсистема управления	15
3.6.1	Методика контроля PBM Service	15
3.6.2	Методика контроля Autoconfig Service	15
3.6.3	Методика контроля Triton Service	15
3.6.4	Методика контроля Rabbitmq Service	15
3.6.5	Методика контроля Events Backend Service	16

3.6.6 Методика контроля Avatars Service	16
---	----

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе применяют следующие сокращения с соответствующими расшифровками (см. таблицу 1):

Таблица 1 – Сокращения и расшифровки

Сокращение, термин	Расшифровка и определение
СРМ	Сервисно-ресурсная модель. Логическая модель сервиса, описывающая состав и взаимосвязи компонентов
PSN	Postal Solution (англ.). Сервис почты, календарей и контактов

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Введение

Сервисно-ресурсная модель (SRM) – это логическая модель сервиса, описывающая состав и взаимосвязи компонентов (ресурсов), которые совместно обеспечивают предоставление сервиса. SRM может быть представлена в виде иерархического графа, узлами которого являются компоненты, а ребрами – связи между ними. SRM может быть использована для решения широкого круга задач, однако в первую очередь предназначена для мониторинга параметров качества сервиса, который может выполняться в рамках процесса управления доступностью. В данном документе графическое представление SRM приведено в разделе [Диаграмма SRM](#), а методики проверки значений параметров сервиса - в соответствующих разделах.

1.2 Влияния Сервисов на Подсистемы

В данном документе используются несколько уровней влияния отдельного Сервиса на Подсистему и на Систему в целом. Уровни определены ниже, в таблице 2.

Таблица 2 - Уровни влияния сервисов на Подсистему

Уровень влияния	Описание
Critical	Отказ Сервиса приводит к полной неработоспособности Подсистемы
Major	Отказ Сервиса не приводит к неработоспособности Подсистемы/Отказ Сервиса приводит к отсутствию критичной функциональности при общей доступности Сервиса
Minor	Отказ Сервиса приводит к неработоспособности редко используемой функциональности, либо к падению производительности
Warning	Отказ Сервиса не приводит к потере функциональности (например, отказ одного из узлов кластера)

2 ДИАГРАММА СРМ ПОЧТА (PSN)

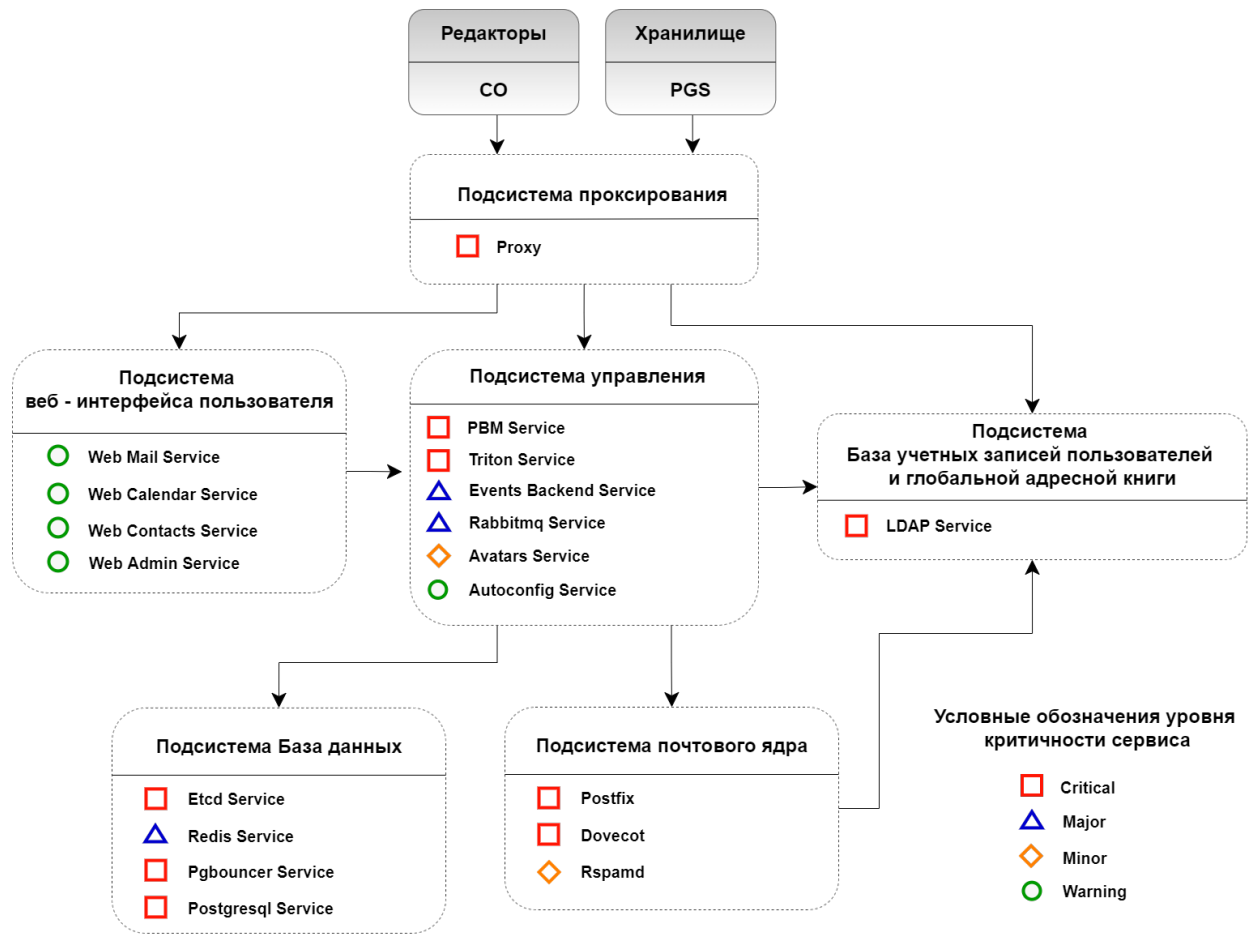


Рисунок 1 - Сервисно-ресурсная модель Почта (PSN)

3 МЕТОДЫ КОНТРОЛЯ СЕРВИСОВ

В описании методов контроля приведены команды, которые необходимо выполнить на всех серверах, где запущен проверяемый сервис. Таблица 3 устанавливает соответствие между описанием сервисов Почта (PSN) и ролями хостов, описанными в главе «Описание ролей» документа «Руководство по установке почтового сервера».

Таблица 3 - Соответствие сервисов и ролей

Сервис	Сервисно-ресурсная модель	Роль (группа хостов МойОфис Почта)
Proxy service	Подсистема проксирования	frontend(proxy)
LDAP Service	Подсистема база учетных записей пользователей и глобальной адресной книги	db(ldap)
EtcD Service	Подсистема базы данных	db(etcd)
Redis Service	Подсистема базы данных	db(redis)
Pgbouncer Service	Подсистема базы данных	db(postgrtes)
Postgresql Service	Подсистема базы данных	db(postgrtes)
Postfix Service	Подсистема почтового ядра	mail(mail)
Dovecot Service	Подсистема почтового ядра	mail(mail)
Rspamd Service	Подсистема почтового ядра	mail(mail)
Web Mail Service	Подсистема веб-интерфейса пользователя	frontend(frontend)
Web Calendar Service	Подсистема веб-интерфейса пользователя	frontend(frontend)
Web Contacts Service	Подсистема веб-интерфейса пользователя	frontend(frontend)
Web Admin Service	Подсистема веб-интерфейса пользователя	frontend(frontend)
Autoconfig Service	Подсистема управления	backend(backend)
PBM Service	Подсистема управления	backend(backend)
Triton Service	Подсистема управления	backend(backend)
Rabbitmq Service	Подсистема управления очередями	backend(backend)
Events Backend Service	Подсистема управления	backend(backend)

Сервис	Сервисно-ресурсная модель	Роль (группа хостов МойОфис Почта)
Avatars Service	Подсистема управления	backend(backend)

3.1 Подсистема проксирования

Таблица 4 - Параметры подсистемы проксирования

Контролируемый параметр	Корректное значение	Критичность
Proxy Service	running	Critical

3.1.1 Методика контроля Proxy Service

Для проверки статуса Proxy service на серверах группы frontend выполнить следующие команды:

```
docker ps -qf name=psn-nginx-proxy_nginx --format "{{.State}}"
```

Пример вывода команды:

```
running
```

Дополнительно рекомендуется выполнить команду для проверки прослушивающих tcp-портов (должны быть открыты 25, 80, 143, 443, 465, 589, 636, 993):

```
ss -nt4l
```

Пример вывода:

```
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
LISTEN 0      65535      *:*         *:*
```

Запросить версии основных компонентов:

```
curl -k https://localhost/version
```

Пример вывода:

```
{"release":"2.0", "web_mail":"2.2.7-418", "web_calendar":"2.0.29-331",  
"web_contacts":"2.0.17-436", "triton":"1.0.63-936", "pbm":"2.3.25-266",  
"autoconfig":"1.0.0-33"}
```

3.2 Подсистема база учетных записей пользователей

Таблица 5 - Параметры подсистемы базы учетных записей пользователей

Контролируемый параметр	Корректное значение	Критичность
LDAP Service	dn: cn=manager...	Critical

3.2.1 Методика контроля LDAP Service

Для проверки статуса LDAP service на серверах группы ldap выполнить команду:

```
docker exec $(docker ps -qf name=psn-ldap_ldap) bash -c 'ldapwhoami -xD  
$LDAP_ROOT_DN -w $LDAP_ROOT_PASSWORD'
```

Примечание: вводить команду как указано, раскрывать значения переменных \$LDAP_ не нужно.

3.3 Подсистема база данных

Таблица 6 - Параметры подсистемы базы данных

Контролируемый параметр	Корректное значение	Критичность
Etcd Service	is healthy	Critical
Redis Service	PONG	Major
Pgbouncer Service	accepting connections	Critical
Postgresql Service	accepting connections	Critical

3.3.1 Методика контроля Etcd Service

Для проверки статуса Etcd service на серверах группы etcd выполнить команду:

```
docker exec $(docker ps -qf name=etcd_etcd.[0-9]) ash -c 'etcdctl --write-out=simple --endpoints=$ENDPOINTS endpoint health'
```

3.3.2 Методика контроля Postgresql Service

Для проверки статуса Postgresql service на серверах группы postgres выполнить команду:

```
docker exec $(docker ps -qf name=psn-postgres_postgres) pg_isready
```

3.3.3 Методика контроля Pgbouncer Service

Для проверки статуса Pgbouncer на серверах группы postgres выполнить команду:

```
docker exec $(docker ps -qf name=psn-postgres_pgb) pg_isready -d postgresql://psn:postgres_db_user@pgbouncer:5432
```

Значение параметра `postgres_db_user` будет получено из переменной инвентарного файла `setup.passwords.postgres_db_user` (см. главу «Конфигурирование инвентарного файла: переменные» документа «Руководство по установке почтового сервера»).



Статус сервиса Pgbouncer зависит от статуса Postgresql Service

3.3.4 Методика контроля Redis Service

Для проверки статуса Redis service на серверах группы redis выполнить команду:

```
docker exec $(docker ps -qf name='psn-redis_redis(.[0-9]|-slave|-master)') bash -c 'redis-cli -a $REDIS_PASSWORD ping'
```

Примечание: вводить команду как указано, раскрывать значения переменной `$REDIS_` не нужно.

3.4 Подсистема почтового ядра

Таблица 7 - Параметры подсистемы почтового ядра

Контролируемый параметр	Корректное значение	Критичность
Postfix Service	is running	Critical

Контролируемый параметр	Корректное значение	Критичность
Dovecot Service	running=yes	Critical
Rspamd Service	running	Minor

3.4.1 Методика контроля Postfix Service

Для проверки статуса Postfix service на серверах группы mail выполнить команду:

```
docker exec $(docker ps -qf name=postfix) postfix status
```

3.4.2 Методика контроля Dovecot Service

Для проверки статуса Dovecot service на серверах группы mail необходимо выполнить команду:

```
docker exec $(docker ps -qf name=dovecot) doveadm -f flow instance list
```

В случае кластерной установки на любом сервере группы mail выполнить:

```
docker exec $(docker ps -qf name=director) doveadm director ring status
```

Вывод должен содержать status synced

```
docker exec $(docker ps -qf name=director) doveadm director status
```

Вывод должен содержать state up

3.4.3 Методика контроля Rspamd Service

Для проверки статуса Rspamd service на серверах группы mail выполнить команду:

```
docker ps -f name=rspamd --format "{{.State}}"
```

3.5 Подсистема веб-интерфейса пользователя

Таблица 8 - Параметры подсистемы веб - интерфейса пользователя

Контролируемый параметр	Корректное значение	Критичность
Web Mail Service	running	Warning

Контролируемый параметр	Корректное значение	Критичность
Web Calendar Service	running	Warning
Web Contacts Service	running	Warning
Web Admin Service (при установке без интеграции)	running	Warning

3.5.1 Методика контроля Web Mail Service

Для проверки статуса Web Mail service на серверах группы frontend выполнить команду:

```
docker ps -qf name=web_mail --format "{{.State}}"
```

3.5.2 Методика контроля Web Calendar Service

Для проверки статуса Web Calendar service на серверах группы frontend выполнить команду:

```
docker ps -qf name=web_calendar --format "{{.State}}"
```

3.5.3 Методика контроля Web Contacts Service

Для проверки статуса Web Contacts service на серверах группы frontend выполнить команду:

```
docker ps -qf name=web_contacts --format "{{.State}}"
```

3.5.4 Методика контроля Web Admin Service

Для проверки статуса Web Admin service на серверах группы frontend выполнить команду:

```
docker ps -qf name=web_admin --format "{{.State}}"
```

Примечание: При установке PSN в составе Частного Облака сервис Web Admin не устанавливается.

3.6 Подсистема управления

Таблица 9 - Параметры подсистемы управления

Контролируемый параметр	Корректное значение	Критичность
PBM Service	running	Critical
Triton Service	running	Critical
Events Backend Service	running	Major
Rabbitmq Service	{"status":"ok"}	Major
Avatars Service	running	Minor
Autoconfig Service	running	Warning

3.6.1 Методика контроля PBM Service

Для проверки статуса PBM service на серверах группы backend выполнить команду:

```
docker ps -qf name=psn-backend_pbm --format "{{.State}}"
```

3.6.2 Методика контроля Autoconfig Service

Для проверки статуса Autoconfig service на серверах группы backend выполнить команду:

```
docker ps -qf name=psn-backend_autoconfig --format "{{.State}}"
```

3.6.3 Методика контроля Triton Service

Для проверки статуса Triton service на серверах группы backend выполнить команду:

```
docker ps -qf name=psn-backend_triton --format "{{.State}}"
```

3.6.4 Методика контроля Rabbitmq Service

Для проверки статуса Rabbitmq service на серверах группы proxy выполнить команду:

```
docker exec $(docker ps -qf name=nginx) curl -s -u  
rabbitmq:setup.passwords.rabbitmq_user rabbitmq:15672/api/aliveness-test/%2F
```

Значение параметра `setup.passwords.rabbitmq_user` будет получено из переменной инвентарного файла `rabbitmq_user` блока `setup/passwords` (см. главу «Конфигурирование инвентарного файла: переменные» документа «Руководство по установке почтового сервера»).

3.6.5 Методика контроля Events Backend Service

Для проверки статуса `Events Backend service` на серверах группы `backend` выполнить команду:

```
docker ps -qf name=psn-backend_events_backend --format "{{.State}}"
```

3.6.6 Методика контроля Avatars Service

Для проверки статуса `Avatars service` на серверах группы `backend` выполнить команду:

```
docker ps -qf name=psn-backend_avatars --format "{{.State}}"
```