МойОфис
Частное Облако 2

Сервисно-ресурсная модель

2.1

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «МОЙОФИС ЧАСТНОЕ ОБЛАКО 2» СЕРВИСНО-РЕСУРСНАЯ МОДЕЛЬ

2.1

На 35 листах

Москва

2022

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

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

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

Таблица 1 – Сокращения и расшифровки

Сокращение, термин	Расшифровка и определение
ОС	Операционная система.
СРМ	Сервисно-ресурсная модель. Логическая модель сервиса, описывающая состав и взаимосвязи компонентов.
API	Application programming interface (англ.), программный интерфейс приложения — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
CDN	Content Delivery Network или Content Distribution Network (англ.), сеть доставки (и дистрибуции) содержимого — распределённая сетевая инфраструктура, позволяющая оптимизировать доставку и дистрибуцию содержимого конечным пользователям в сети Интернет.
CO	Cloud Office (англ.), Облачный Офис, общее название продукта.
CVM	Conversion Manager (англ.), Сервис управления конвертированием файлов.
CU	Conversion Unit (англ.), экземпляр процесса конвертора различных форматов файлов.
DCM	Document Collaboration Manager (англ.), Сервис управления редактированием документов.
DU	Document Unit (англ.), экземпляр процесса редактирования и коллаборации документов.
FM	File Manager (англ.). Сервис файлового менеджера.
HA	High Availability (англ.), высокая доступность — характеристика технической системы, разработанной во избежание невыполненного обслуживания путём уменьшения или управления сбоями и минимизацией времени плановых простоев.
JOD	Java OpenDocument Converter (англ.). Сервис конвертирования документов

Сокращение, термин	Расшифровка и определение
PGS	Pythagoras Storage (англ.). Компонент сервиса МойОфис Частное Облако, представляет собой объектное хранилище.
WFE	Web Frontend (англ.). Общее название web приложений МойОфис Частное Облако.
FM	File Manager (англ.). Сервис файлового менеджера.
HA	High Availability (англ.), высокая доступность — характеристика технической системы, разработанной во избежание невыполненного обслуживания путём уменьшения или управления сбоями и минимизацией времени плановых простоев.
JOD	Java OpenDocument Converter (англ.). Сервис конвертирования документов стандарта OpenDocument.
NM	Notification Manager (англ.). Сервис управления оповещениями.
SSO	Single Sign-On (англ.), технология единого входа — технология, при использовании которой пользователь переходит из одной системы в другую, не связанную с первой системой, без повторной аутентификации.
WFE	Web Frontend (англ.). Общее название web приложений МойОфис Частное Облако.

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Введение

Сервисно-ресурсная модель (СРМ) – это логическая модель сервиса, описывающая состав и взаимосвязи компонентов (ресурсов), которые совместно обеспечивают предоставление сервиса. Как правило, СРМ имеет графическое представление в виде иерархического графа, узлами которого являются компоненты, а ребрами – связи между ними. СРМ 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 СРМ приведено в разделе Диаграммы, а методики проверки значений параметров сервиса - в соответствующих разделах.

1.2 Градации и определение степени 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 потере функциональности (например, отказ одного из узлов кластера).

2 ДИАГРАММЫ

2.1 Сервис «МойОфис Частное Облако»

Рисунок 1 – Сервисно-ресурсная модель МойОфис Частное Облако

2.2 Сервисно-ресурсная модель Редакторы (CO)

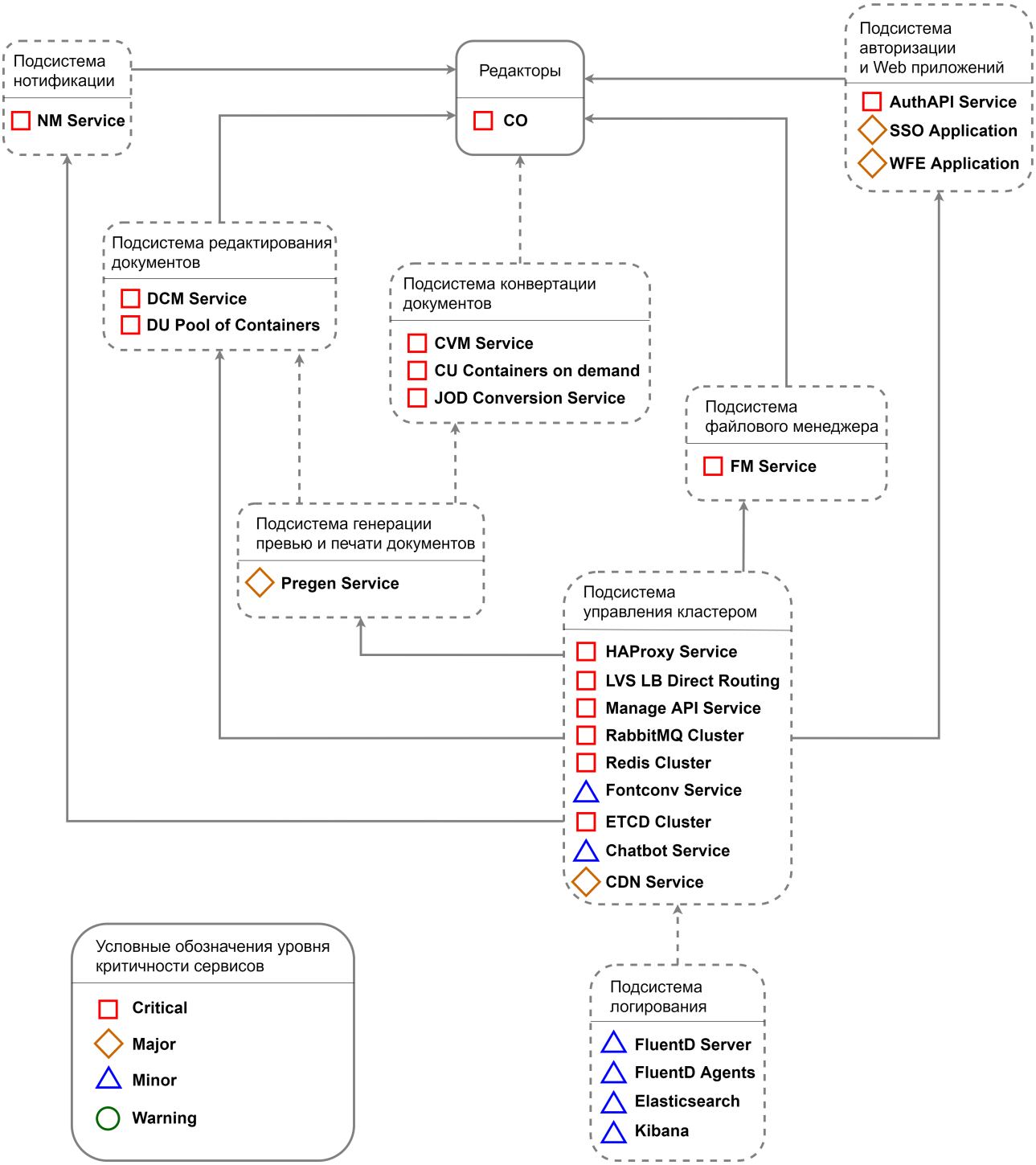

Рисунок 2 – Сервисно-ресурсная модель Редакторы(CO)

2.3 Сервисно-ресурсная модель Хранилище (PGS)

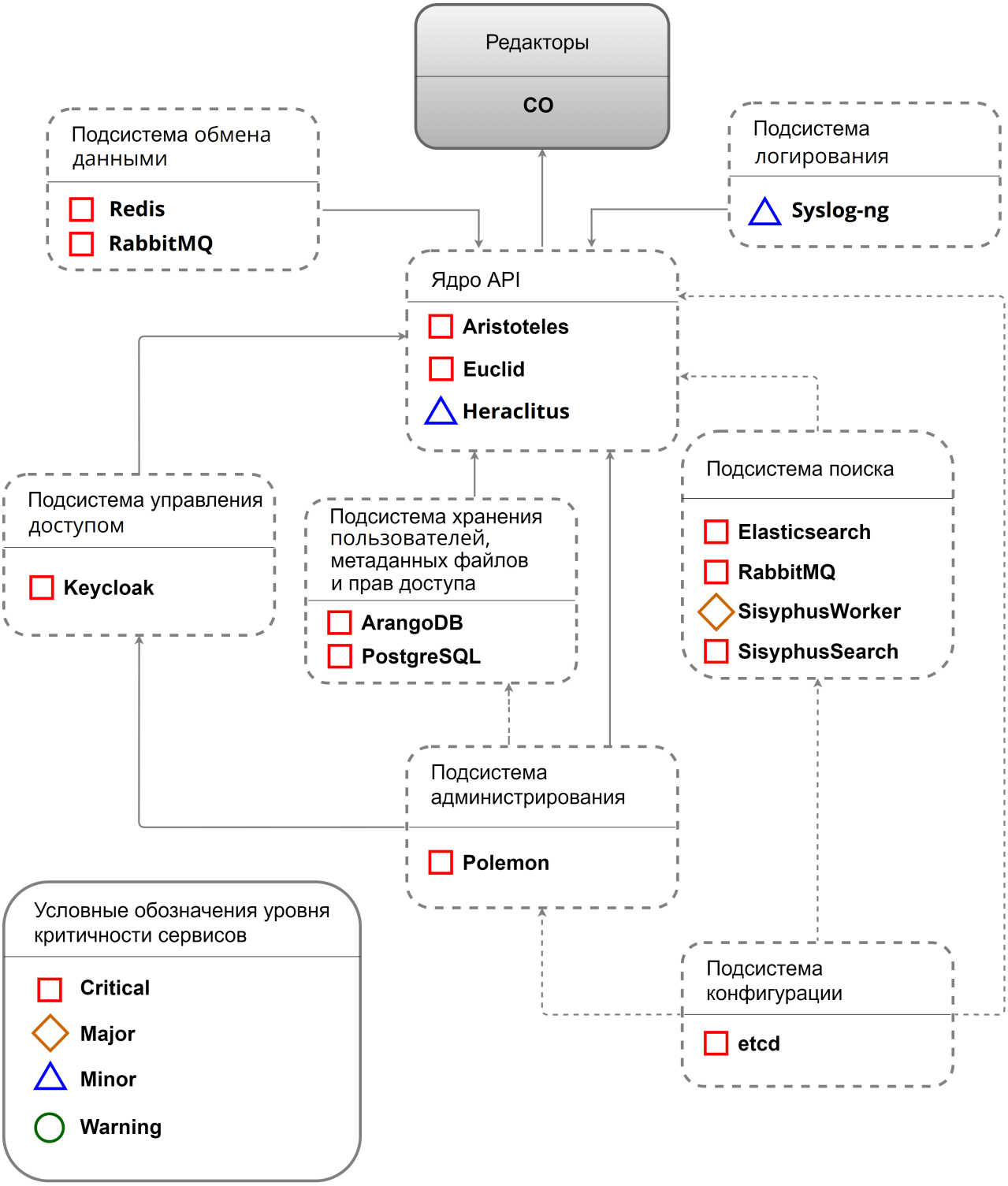

Рисунок 5 – Сервисно-ресурсная модель Хранилище (PGS)

3 РЕДАКТОРЫ (CO)

В описании методов контроля используются переменные вида **$PRIVATE_IPV4**, **$CVM_PORT** или **$DCM_CHECK_PATH**, которые определяют параметры конкретного окружения. В качестве значения переменной **$PRIVATE_IPV4** используйте внутренний IP адрес хоста той роли, проверка которой производится. Остальные переменные описаны в методах контроля.

В дальнейшем везде, где описывается файл с путем **/opt/co**, стоит учитывать, что расположение файла может быть иным, если при развертывании было переопределено значение переменной **CO_DIR**.

Контроль функционирования сервисов выполняется запросами из командной строки, вызовами утилиты **curl**.

3.1 Подсистема авторизации и Web приложений

Таблица 3 - Параметры подсистемы авторизации и Web приложений

Контролируемый параметр	Корректное значение	Критичность
Auth API	OK	Critical
TLS GOST Proxy	running	Critical
SSO Application	running	Major
WFE Application	running	Major
WTE Application	running	Major

3.1.1 Методика контроля AuthAPI

Проверка статуса работы подсистемы AuthAPI осуществляется командной строке:

```
curl -s
http://$PRIVATE_IPV4:$OPENRESTY_LB_CORE_AUTH_MNG_PORT/api/manage/core/st
atus | jq '.all'
```

Значение параметра OPENRESTY_LB_CORE_AUTH_MNG_PORT приведено в Приложении 1 Руководства по установке.

Пример ответа:

```
"OK"
```

3.1.2 Методика контроля TLS GOST Proxy

Внимание! Данная проверка распространяется только на версии приложения, содержащие криптографические преобразования согласно ГОСТ РФ.

Для проверки статуса TLS GOST требуется убедиться в том, что контейнер с приложением запущен, для этого в терминале от пользователя **root** выполнить команду:

```
curl -s --unix-socket /var/run/docker.sock http://v1.38/containers/json | jq '.[]
| select(.Names == ["/nginx-gost"]) | .State'
```

Пример ответа:

```
"running"
```

3.1.3 Методика контроля параметров SSO Application

Для проверки статуса SSO Application на всех хостах с ролью **lb-core-auth** убедиться в том, что контейнер с приложением запущен, для этого в терминале от пользователя **root** выполнить команду:

```
curl -s --unix-socket /var/run/docker.sock http:/v1.38/containers/json |
jq '.[] | select(.Names == ["/web-sso"]) | .State'
```

Пример ответа:

```
"running"
```

3.1.4 Методика контроля параметров WFE Application

Для проверки статуса WFE Application на всех хостах с ролью **lb-core-auth** убедиться в том, что контейнер с приложением запущен, для этого в терминале от пользователя **root** выполнить команду:

```
curl -s --unix-socket /var/run/docker.sock http:/v1.38/containers/json | jq '.[] | select(.Names == ["/web-wfe"]) | .State'
```

Пример ответа:

```
"running"
```

3.1.5 Методика контроля параметров WTE Application

Для проверки статуса WTE Application на всех хостах с ролью **lb-core-auth** убедиться в том, что контейнер с приложением запущен, для этого в терминале от пользователя **root** выполнить команду:

```
curl -s --unix-socket /var/run/docker.sock http:/v1.38/containers/json | jq '.[] | select(.Names == ["/web-wte"]) | .State'
```

Пример ответа:

```
"running"
```

3.2 Подсистема файлового менеджера

Таблица 4 - Параметры подсистемы файлового менеджера

Контролируемый параметр	Корректное значение	Критичность
FM Service	UP	Critical

3.2.1 Методика контроля FM Service

```
curl -s $FM_SCHEME://$FM_HOST:$FM_PORT/$FM_CHECK_PATH | jq '.status'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/fm-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $FM_SCHEME

-h $FM_HOST

-p $FM_PORT

-u $FM_CHECK_PATH

Пример ответа:

```
"UP"
```

3.3 Подсистема конвертации документов

Таблица 5 - Параметры подсистемы конвертации документов

Контролируемый параметр	Корректное значение	Критичность
CVM Service	UP	Critical
CU Pool of Containers	true	Critical
JOD Conversion Service	UP	Critical

3.3.1 Методика контроля CVM Service

Проверка статуса работы подсистемы CVM Service осуществляется в командной строке:

```
curl -s $CVM_SCHEME://$CVM_HOST:$CVM_PORT/$CVM_CHECK_PATH | jq '.status'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/cvm-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $CVM_SCHEME

-h $CVM_HOST

-p $CVM_PORT

-u $CVM_CHECK_PATH

Пример ответа:

```
"UP"
```

3.3.2 Методика контроля параметров CU Pool of Containers

Проверка статуса работы подсистемы CU Containers on demand осуществляется в командной строке:

```
curl -s $CVM_SCHEME://$CVM_HOST:$CVM_PORT/$CVM_CHECK_PATH | jq
'.components.CVM.details.npsPool > 0'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/cvm-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $CVM_SCHEME

-h $CVM_HOST

-p $CVM_PORT

-u $CVM_CHECK_PATH

Пример ответа:

```
"true"
```

3.3.3 Методика контроля параметров JOD Conversion Service

```
curl -s $JOD_SCHEME://$PRIVATE_IPV4:$JOD_PORT/$JOD_CHECK_PATH | jq
'.status'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/jod-watchdog.service.

В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $JOD_SCHEME

-p $JOD_PORT

-u $JOD_CHECK_PATH

Пример ответа:

```
"UP"
```

3.4 Подсистема генерации превью и печати документов

Таблица 6 - Параметры подсистемы генерации превью и печати документов

Контролируемый параметр	Корректное значение	Критичность
Pregen Service	OK	Major

3.4.1 Методика контроля Pregen Service

Команда контроля **Pregen**:

```
curl -sX HEAD -I
$PREGEN_SCHEME://$PREGEN_HOST:$PREGEN_PORT$PREGEN_CHECK_PATH | grep HTTP
| awk '{print $3 }'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/pregen-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $PREGEN_SCHEME

-h $PREGEN_HOST

-p $PREGEN_PORT

-u $PREGEN_CHECK_PATH

Пример ответа:

```
"OK"
```

3.5 Подсистема редактирования документов

Таблица 7 - Параметры подсистемы редактирования документов

Контролируемый параметр	Корректное значение	Критичность
DCM Service	UP	Critical
DU Pool of Containers	ok	Critical

3.5.1 Методика контроля DCM Service

```
curl -s $DCM_SCHEME://$DCM_HOST:$DCM_PORT/$DCM_CHECK_PATH | jq '.status'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/dcm-watchdog.service.

В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $DCM_SCHEME

-h $DCM_HOST

-p $DCM_PORT

-u $DCM_CHECK_PATH

Пример ответа:

```
"UP"
```

3.5.2 Методика контроля параметров DU Pool of Containers

Количество доступных для работы контейнеров DU определяется командой:

```
curl -s $DCM_SCHEME://$DCM_HOST:$DCM_PORT/manage/documents | jq length
```

Полученное значение должно соответствовать значению **Du.PoolSize** в файле **/opt/co/etcd/config/nps-du.properties**

3.6 Подсистема нотификации

Таблица 8 - Параметры подсистемы нотификации

Контролируемый параметр	Корректное значение	Критичность
NM Service	UP	Critical

3.6.1 Методика контроля параметров NM Service

```
curl -s $NM_SCHEME://$PRIVATE_IPV4:$NM_PORT/$NM_CHECK_PATH | jq
'.status'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/nm-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $NM_SCHEME

-p $NM_PORT

-u $NM_CHECK_PATH

Пример ответа:

```
"UP"
```

3.7 Подсистема управления кластером

Таблица 9 - Параметры подсистемы управления кластером

Контролируемый параметр	Корректное значение	Критичность
Redis Cluster	PONG	Critical
RabbitMQ Cluster	ok	Critical
ETCD Cluster	true	Critical
HA Proxy Service	200 OK Service ready	Critical
LVS LB Direct Routing	true	Critical
Manage API Service	OK	Critical
Fontconv Service	OK	Minor
CDN Service	running	Major
Chatbot Service	OK	Minor

3.7.1 Методика контроля Redis Cluster

Для проверки сервиса Redis Cluster на хосте, с которого будет производиться проверка, требуется предварительно установить утилиту **redis-cli**.

Для rpm-based дистрибутивов это можно сделать командой:

```
yum install redis
```


Для deb-based дистрибутивов это можно сделать командой:

```
sudo apt install redis-tools
```

После чего выполнить проверку корректности работы сервиса Redis Cluster

```
redis-cli -h $PRIVATE_IPV4 -p $REDIS_SENTINEL_PORT ping
```

Значение параметра REDIS_SENTINEL_PORT приведено в Приложении 1 Руководства по установке, где он описан под именем REDIS_SEN_PORT.

Пример ответа:

```
"PONG"
```

3.7.2 Методика контроля RabbitMQ Cluster

```
curl -s
$RABBITMQ_SCHEME://$RABBITMQ_USERNAME:$RABBITMQ_PASSWORD@$RABBITMQ_NODE:
$RABBITMQ_MNG_PORT$RABBITMQ_CHECK_PATH | jq '.status'
```

Значения параметров RABBITMQ_USERNAME, RABBITMQ_PASSWORD, RABBITMQ_MNG_PORT приведены в Приложении 1 Руководства по установке.

Значения остальных параметров вызова приведены в файле /opt/co/systemd/units/mq-ms-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $RABBITMQ_SCHEME

-h $RABBITMQ_NODE

-u $RABBITMQ_CHECK_PATH

Пример ответа:

```
"OK"
```

3.7.3 Методика контроля ETCD Cluster

```
curl -ks
https://$ETCD_BROWSER_USERNAME:$ETCD_BROWSER_PASSWORD@$PRIVATE_IPV4:$ETC
D_CLIENT_PORT/health | jq '.health'
```

Значения параметров ETCD_BROWSER_USERNAME, ETCD_BROWSER_PASSWORD, ETCD_CLIENT_PORT приведены в Приложении 1 Руководства по установке.

Пример ответа:

```
"true"
```

3.7.4 Методика контроля HA Proxy Service

```
curl -s
$HAPROXY_SCHEME://$HAPROXY_USERNAME:$HAPROXY_PASSWORD@$PRIVATE_IPV4:$HAP
ROXY_PORT_STATS_HTTP$HAPROXY_CHECK_PATH |  sed -e 's/<[^>]*>//g'
```

Значения параметров HAPROXY_USERNAME, HAPROXY_PASSWORD, приведены в Приложении 1 Руководства по установке.

Значения остальных параметров вызова приведены в файле /opt/co/systemd/units/haproxy-http-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $HAPROXY_SCHEME

-p $HAPROXY_PORT_STATS_HTTP

-u $ HAPROXY CHECK_PATH

Пример ответа:

```
200 OK
```

```
Service ready.
```

3.7.5 Методика контроля Manage API Service

На хостах с ролью **lb_core_auth** в консоли выполнить команду:

```
curl -s
http://$PRIVATE_IPV4:$OPENRESTY_LB_CORE_AUTH_MNG_PORT/api/manage/config/
version
```

Значение параметра OPENRESTY_LB_CORE_AUTH_MNG_PORT приведено в Приложении 1 Руководства по установке.

Ответ сервера с кодом 200 OK может считаться признаком штатной работы сервиса.

3.7.6 Методика контроля параметровFontconv Service

```
curl -sX HEAD -I
$FONTCONV_SCHEME://$FONTCONV_HOST:$FONTCONV_PORT$FONTCONV_CHECK_PATH |
grep HTTP | awk '{print $3 }'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/fontconv-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $FONTCONV_SCHEME

-h $FONTCONV_HOST

-p $FONTCONV_PORT

-u $FONTCONV_CHECK_PATH

Пример отображения результата выполнения команды:

```
OK
```

3.7.7 Методика контроля параметров CDN Service

На хостах с ролью **lb_core_auth** в консоли выполнить команду

```
systemctl is-active lsyncd
```

Корректным ответом будет **active.**

3.7.8 Методика контроля параметров Chatbot Service

```
curl -sX HEAD -I
$CHATBOT_SCHEME://$CHATBOT_HOST:$CHATBOT_PORT$CHATBOT_CHECK_PATH | grep
HTTP | awk '{print $3 }'
```

Значения параметров вызова приведены в файле /opt/co/systemd/units/chatbot-watchdog.service. В разделе [Service] описаны следующие аргументы и значения переменной ExecStart:

-s $CHATBOT_SCHEME

-h $CHATBOT _HOST

-p $CHATBOT _PORT

-u $CHATBOT_CHECK_PATH

Пример ответа:

```
OK
```

3.8 Подсистема логирования

Таблица 10 - Параметры подсистемы логирования

Контролируемый параметр	Корректное значение	Критичность
Fluent Server	OK	Minor
Fluent Agents	OK	Minor
Elasticsearch	green	Minor
Kibana	OK	Minor

3.8.1 Методика контроля параметров Fluent Server

Проверка проводится в командной строке на хостах с ролью **log**

```
docker exec -t fluentd-server curl -svo /dev/null
http://127.0.0.1:$FLUENTD_SERVER_PORT_MONITOR/api/plugins.json | grep '<
HTTP' | awk '{print $4 }'
```

Значение параметра FLUENTD_SERVER_PORT_MONITOR приведено в Приложении 1 Руководства по установке.

Пример корректного ответа:

```
OK
```

3.8.2 Методика контроля параметров Fluent Agents

Проверка проводится в командной строке на всех хостах

```
docker exec -t fluentd-agent curl -svo /dev/null
http://127.0.0.1:$FLUENTD_AGENT_PORT_MONITOR/api/plugins.json | grep '<
HTTP' | awk '{print $4 }'
```

Значение параметра FLUENTD_AGENT_PORT_MONITOR приведено в Приложении 1 Руководства по установке.

Пример корректного ответа:

```
OK
```

3.8.3 Методика контроля параметров Elasticsearch

Проверка проводится в командной строке на хостах с ролью **log**

```
docker exec -t fluentd-server curl -s "localhost:9200/_cluster/health" | jq '.status'
```

Ответ **green** будет признаком корректного функционирования Elasticsearch.

3.8.4 Методика контроля параметров Kibana

Проверка проводится в командной строке на хостах с ролью **log**

```
curl –v -s --user "$KIBANA_USERNAME:$KIBANA_PASSWORD" "http://$PRIVATE_IPV4:$KIBANA_PORT/app/kibana/" | grep '< HTTP'
```

Значение параметров KIBANA_USERNAME, KIBANA_PASSWORD приведены в Приложении 1 Руководства по установке.

Значение переменной $KIBANA_PORT соответствует 8000 для standalone инсталяций и 80 – для кластерных инсталяций.

Ответ **HTTP 200 OK** будет признаком корректного функционирования Kibana.

4 ХРАНИЛИЩЕ PGS

4.1 Описание сервисов для отказоустойчивой установки

Посмотреть все сервисы и их статус можно командой:

```
docker service ls --format 'table {{.Name}}\t{{.Replicas }}'
```

По умолчанию поддерживается инсталляция с количеством реплик = 1, соответственно вывод должен быть следующим:

```
NAME                                REPLICAS
pgs-arangodb_arangodb               1/1
pgs-elasticsearch_elasticsearch     1/1
pgs-etcd_etcd                       1/1
pgs-keycloak_keycloak               1/1
pgs-keycloak_postgres               1/1
pgs-rabbitmq_rabbitmq               1/1
pgs-redis_redis                     1/1
pgs_aristoteles                     1/1
pgs_euclid                          1/1
pgs_polemon                         1/1
pgs_sisyphussearch                  1/1
pgs_sisyphusworker                  1/1
```

4.2 Подсистема Ядро API

Таблица 11 - Параметры подсистемы Ядро API

Контролируемый параметр	Корректное значение	Критичность
Aristoteles	true	Critical
Euclid	true	Critical
Heraclitus	Running	Minor

4.2.1 Методика контроля Aristoteles

Проверка статуса работы Aristoteles Service сервиса в командной строке:

```
curl -k -XPOST  https://$PGS_URL/pgsapi/\?cmd\=api_version | jq
.response.success
```

Пример ответа:

```
"true"
```

4.2.2 Методика контроля Euclid

Проверка статуса работы Aristoteles Service сервиса в командной строке:

```
curl -k -XPOST  https://$PGS_URL/adminapi/\?cmd\=api_version | jq
.response.success
```

Пример ответа:

```
"true"
```

4.2.3 Методика контроля Heraclitus

Проверка статуса работы Redis Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}'
pgs_heraclitus
```

Пример ответа:

```
NAME                 DESIRED STATE
pgs_heraclitus.1     Running
```

4.3 Подсистема обмена данными

Таблица 12 - Параметры подсистемы обмена данными

Контролируемый параметр	Корректное значение	Критичность
Redis	PONG	Critical
RabbitMQ	Running	Critical

4.3.1 Методика контроля Redis

Проверка статуса работы сервиса Redis в командной строке:

```
docker exec -it $(docker ps -qf name=redis_redis-master) redis-cli --
pass RjirbyfKfgf ping
```

Пример ответа:

```
PONG
```

4.3.2 Методика контроля RabbitMQ

Проверка статуса работы сервиса RabbitMQ в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}' pgs-
rabbitmq_rabbitmq
```

Пример ответа:

```
NAME                      DESIRED STATE
pgs-rabbitmq_rabbitmq.1   Running
```

4.4 Подсистема управления доступом

Таблица 13 - Параметры подсистемы управления

Контролируемый параметр	Корректное значение	Критичность
Keycloak	Running	Critical

4.4.1 Методика контроля Keycloak Service

Проверка статуса работы Keycloak Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}' pgs-
keycloak_keycloak
```

Пример ответа:

```
NAME                      DESIRED STATE
pgs-keycloak_keycloak.1   Running
```

4.5 Подсистема поиска

Таблица 14 - Параметры подсистемы поиска

Контролируемый параметр	Корректное значение	Критичность
Elasticsearch	green	Critical
RabbitMQ	Running	Critical
SisyphusSearch	Running	Critical
SisyphusWorker	Running	Major

4.5.1 Методика контроля Elasticsearch Service

Проверка статуса работы Elasticsearch Service сервиса в командной строке:

```
docker exec -it $(docker ps -qf name=elasticsearch_elasticsearch) curl localhost:9200/_cat/health | awk '{ print $4 }'
```

Пример ответа:

```
green
```

4.5.2 Методика контроля RabbitMQ Service

Проверка статуса работы RabbitMQ Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}' pgs-rabbitmq_rabbitmq
```

Пример ответа:

```
NAME                     DESIRED STATE
pgs-rabbitmq_rabbitmq.1  Running
```

4.5.3 Методика контроля SisyphusSearch Service

Проверка статуса работы SisyphusSearch Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}' pgs_sisyphussearch
```

Пример ответа:

```
NAME                    DESIRED STATE
pgs_sisyphussearch.1    Running
```

4.5.4 Методика контроля SisyphusWorker Service

Проверка статуса работы SisyphusWorker Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}'
pgs_sisyphusworker
```

Пример ответа:

```
NAME                     DESIRED STATE
pgs_sisyphusworker.1     Running
```

4.6 Подсистема хранения пользователей, метаданных файлов и прав доступа

Таблица 15 - Параметры подсистемы хранения метаданных файлов и прав доступа

Контролируемый параметр	Корректное значение	Критичность
ArangoDB	Running	Critical
PostgreSQL	accepting connections	Critical

4.6.1 Методика контроля ArangoDB Service

Проверка статуса работы ArangoDB Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}' pgs-
arangodb_arangodb
```

Пример ответа:

```
NAME                        DESIRED STATE
pgs-arangodb_arangodb.1     Running
```

4.6.2 Методика контроля Postgres Service

Проверка статуса работы Postgres Service сервиса в командной строке:

```
docker exec -it $(docker ps -qf name=postgres) pg_isready
```

Пример ответа:

```
/var/run/postgresql:5432 - accepting connections
```

4.7 Подсистема администрирования

Таблица 16 - Параметры подсистемы администрирования

Контролируемый параметр	Корректное значение	Критичность
Euclid	Running	Critical
Polemon	Running	Major

4.7.1 Методика контроля Euclid Service

Проверка статуса работы Euclid Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}'
pgs_euclid
```

Пример ответа:

```
NAME           DESIRED STATE
pgs_euclid.1   Running
```

4.7.2 Методика контроля Polemon Service

Проверка статуса работы Polemon Service сервиса в командной строке:

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}'
pgs_polemon
```

Пример ответа:

```
NAME            DESIRED STATE
pgs_polemon.1   Running
```

4.8 Подсистема конфигурирования

Таблица 17 - Параметры подсистемы конфигурирования

Контролируемый параметр	Корректное значение	Критичность
etcd	cluster is healthy	Critical

4.8.1 Методика контроля ETCD Service

Проверка статуса работы ETCD Service сервиса в командной строке:

```
docker exec -it $(docker ps -qf name=pgs-etcd_etcd) etcdctl cluster-health
```

Пример ответа:

```
member ade526d28b1f92f7 is healthy: got healthy result from http://etcd1:2379
member bd388e7810915853 is healthy: got healthy result from http://etcd3:2379
member d282ac2ce600c1ce is healthy: got healthy result from http://etcd2:2379
cluster is healthy
```

4.9 Подсистема хранения файлов

Таблица 18 - Параметры подсистемы хранения файла

Контролируемый параметр	Корректное значение	Критичность
MinIO	Running	Critical
GlusterFS	Active (running)	Critical

4.9.1 Методика контроля MinIO

Для контроля необходимы параметры, заданные при установке Хранилища (PGS):

- ENV – окружение установки (при наличии);
- DEFAULT_DOMAIN – домен установки;
- S3.secret_key;
- S3.access_key.

Данные переменные описаны в разделе 4.3.2 «Конфигурирование инвентарного файла: переменные» документа «МойОфис Хранилище (PGS). Руководство по установке».

4.9.2 Методика контроля старта сервиса

Проверка старта сервиса в командной строке (где $i – номер инстанса, например: 1,2,3…):

```
docker service ps --format 'table {{.Name}}\t{{.DesiredState}}' pgs-minio_minio$i
```

Пример ответа:

```
NAME                    DESIRED STATE
pgs-minio_minio2.1      Running
```

4.9.3 Методика контроля работы сервиса

Для проверки статуса сервиса необходимо заранее установить Minio Client на клиентскую машину, откуда будет производиться проверка.

В случае наличия сети Интернет скачать и запустить этот клиент можно командой:

```
docker run -it --entrypoint=/bin/sh minio/mc
```

Также необходимо создать алиас подключения:

```
mc alias set minio http://pgs-{{ ENV }}.{{ DEFAULT_DOMAIN}}:9000 {{ S3.access_key  }}  {{ S3.secret_key }}
```

Когда алиас будет создан, можно произвести проверку статуса сервиса командой:

```
mc admin info minio
```

Пример ответа:

```
     minio1:9000
     Uptime: 2 days
     Version: 2021-02-01T22:56:52Z
     Network: 3/3 OK
     Drives: 2/2 OK

     minio3:9000
     Uptime: 2 days
     Version: 2021-02-01T22:56:52Z
     Network: 3/3 OK
     Drives: 2/2 OK

     minio2:9000
     Uptime: 2 days
     Version: 2021-02-01T22:56:52Z
     Network: 3/3 OK
     Drives: 2/2 OK
```

4.9.4 Методика контроля GlusterFS

4.9.4.1 Методика контроля работы сервиса

Проверка статуса работы сервиса в командной строке:

```
Systemctl status glusterd
```

Пример корректного ответа:

```
glusterd.service -GlusterFS, a clustered file-system server
     Loaded: loaded (/usr/lib/systemd/system/glusterd.service; enabled;
vendor preset: disabled)
     Active: active (running) since Tue 2021-01-26 22:33:19 MSK; 4 weeks
```

```
0 days ago
     Docs: man:glusterd(8)
     Main PID: 23706 (glusterd)
     Tasks: 55
     Memory: 2.5G
     CGroup: /system.slice/glusterd.service
```

4.9.4.2 Методика контроля работы Gluster Volume

Проверка статуса в командной строке:

```
gluster volume status pgs-files-volume
```

Пример ответа:

```
Status of volume: pgs-files-volume
```

```
Gluster process                         TCP Port RDMA Port  Online    Pid
------------------------------------------------------------------------------
Brick 10.160.100.170:/data/glusterfs/ pgs-fil
t                                       49152     0          Y       10072
Brick 10.160.100.171:/data/glusterfs/ pgs-fil
t                                       49152     0          Y       23820
Self-heal Daemon on localhost           N/A       N/A        Y       23853
Self-heal Daemon on ldap.domain.ru      N/A       N/A        Y       10093
Task Status of Volume pgs-files-volume
```

Колонка **Online** во всех строках должна иметь значение **Y**. Количество **Brick(s)** приведено для примера и может отличаться (зависит от параметров установки).