

GIMS AUTOMATION

РАБОЧАЯ ДОКУМЕНТАЦИЯ

РУКОВОДСТВО ПО УСТАНОВКЕ СИСТЕМЫ

Москва 2021



СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ	3
1.1	ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	3
1.2	СОСТАВ ПРОГРАММНОГО РЕШЕНИЯ	3
2	УСТАНОВКА ПРОГРАММЫ.....	4
2.1	ТРЕБОВАНИЯ К УЧЕТНОЙ ЗАПИСИ.....	4
2.2	ПРЕДВАРИТЕЛЬНЫЕ ДЕЙСТВИЯ.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
2.2.1	Настройка сервера	Ошибка! Закладка не определена.
2.2.2	Скачивание дистрибутива установки	4
2.3	ПРОЦЕСС РАЗВЕРТЫВАНИЯ СИСТЕМЫ.....	5
2.4	ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ	5
2.5	НАСТРОЙКА ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА AUTOMATION	5
2.6	ОТКАТ К ПРЕДЫДУЩЕЙ ВЕРСИИ.....	7
3	ТРЕБОВАНИЯ К ОБЕСПЕЧЕНИЮ МОНИТОРИНГА	8
3.1	ТРЕБОВАНИЯ К МОНИТОРИНГУ ПРОИЗВОДИТЕЛЬНОСТИ.....	8
3.2	ТРЕБОВАНИЯ К МОНИТОРИНГУ КОМПОНЕНТ СИСТЕМЫ	8
4	ТРЕБОВАНИЯ К БЕКАПИРОВАНИЮ СИСТЕМЫ.....	9
5	ПРИЛОЖЕНИЕ.....	9



1 ОБЩИЕ СВЕДЕНИЯ

1.1 ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Минимальные технические характеристики сервера, на котором будет установлена Система, приведены в таблице ниже.

№	ЭЛЕМЕНТ	ТРЕБОВАНИЯ
1	Процессор	Не менее 4 Core CPU
2	Оперативная память	Не менее 8 GB RAM
3	Жесткий диск	Не менее 20 GB HDD

Сервер, используемый для установки Системы, должен удовлетворять следующим требованиям по программному обеспечению:

№	ЭЛЕМЕНТ	ТРЕБОВАНИЯ
1	Операционная система	CentOS 7/RHEL 7

Операционная система должна быть установлена без дополнительных компонент и без внесения изменений в конфигурацию после установки. Все необходимые дополнительные компоненты устанавливаются из дистрибутива.

1.2 СОСТАВ ПРОГРАММНОГО РЕШЕНИЯ

Технологический стек программного решения состоит из:

НАИМЕНОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	ЛИЦЕНЗИЯ	НАЗНАЧЕНИЕ
PostgreSQL	PostgreSQL License	Хранение информации о настройках компонент системы, инвентарных данных и тд.
Celery	BSD	Брокер отвечающий за формирование очереди заданий выполняемых компонентами.
RabbitMQ	Mozilla public license	Приложение, обеспечивающее обмен данными между компонентами, процессами и сервисами.



НАИМЕНОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	Лицензия	НАЗНАЧЕНИЕ
OpenResty (Nginx)	BSD license	Веб-сервер использующейся как реверсивный прокси для запросов к приложению портала
Gunicorn	MIT license	Веб-сервер отвечающий за запуск WSGI Python
Django	BSD license	Framework
React	MIT license	Framework

2 УСТАНОВКА ПРОГРАММЫ

2.1 ТРЕБОВАНИЯ К УЧЕТНОЙ ЗАПИСИ

Развертывание Системы осуществляется под пользователем с правами root на выделенном сервере, на котором будет осуществляться развертывание Системы.

2.2 СКАЧИВАНИЕ ДИСТРИБУТИВА УСТАНОВКИ

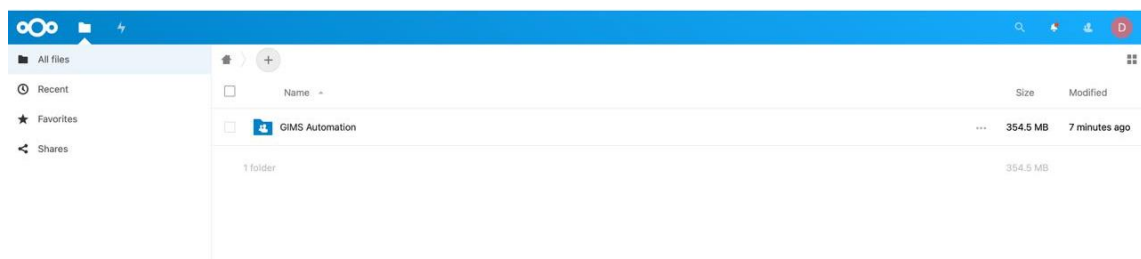
Дистрибутив системы может быть скачен по ссылке: <https://gelarm.ru/nextcloud>

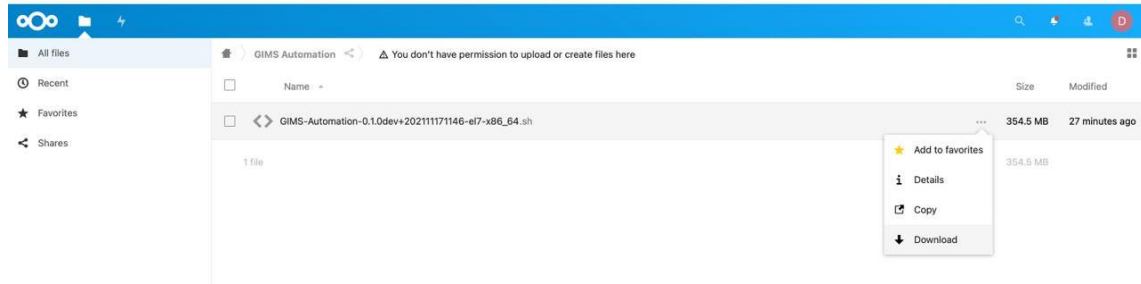
Для этого необходимо выполнить следующие действия.

Шаг 1. – Перейдите по ссылке <https://gelarm.ru/nextcloud>

Шаг 2. – В открывшемся окне введите логин и пароль. (Логин и Пароль передается отдельно.)

Шаг 3. В открывшейся форме кликаем по названию папки GIMS Automation. Внутри папки выполняем скачивание всех содержащихся в ней файлов. Для этого наведите курсор на кнопку ... и в открывшемся выпадающем списке выберите пункт **Download**.





2.3 ПРОЦЕСС РАЗВЕРТЫВАНИЯ СИСТЕМЫ

Процесс развертывания Системы

Шаг 1. Скопируйте дистрибутив, полученный в разделе 2.2 на сервер, где будет производиться установка.

Шаг 2. Запустите консоль и перейдите в папку куда был скопирован дистрибутив.

Шаг 3. Запустите в консоли команду установки вида: `bash ./Название_файла.sh`

Пример команды:

```
bash ./GIMS-Automation-0.1.0dev+202111161720-el7-x86_64.sh
```

2.4 ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Шаг 1. Запустите веб-браузер и введите в адресную строку URL `http://адрес_сервера`.

Шаг 2. На вкладке веб-браузера откроется страница авторизации в Системе.

Шаг 3. На открывшейся странице в форме авторизации введите Логин и Пароль. **(По умолчанию логин `admin` и пароль `123QWErty`).**

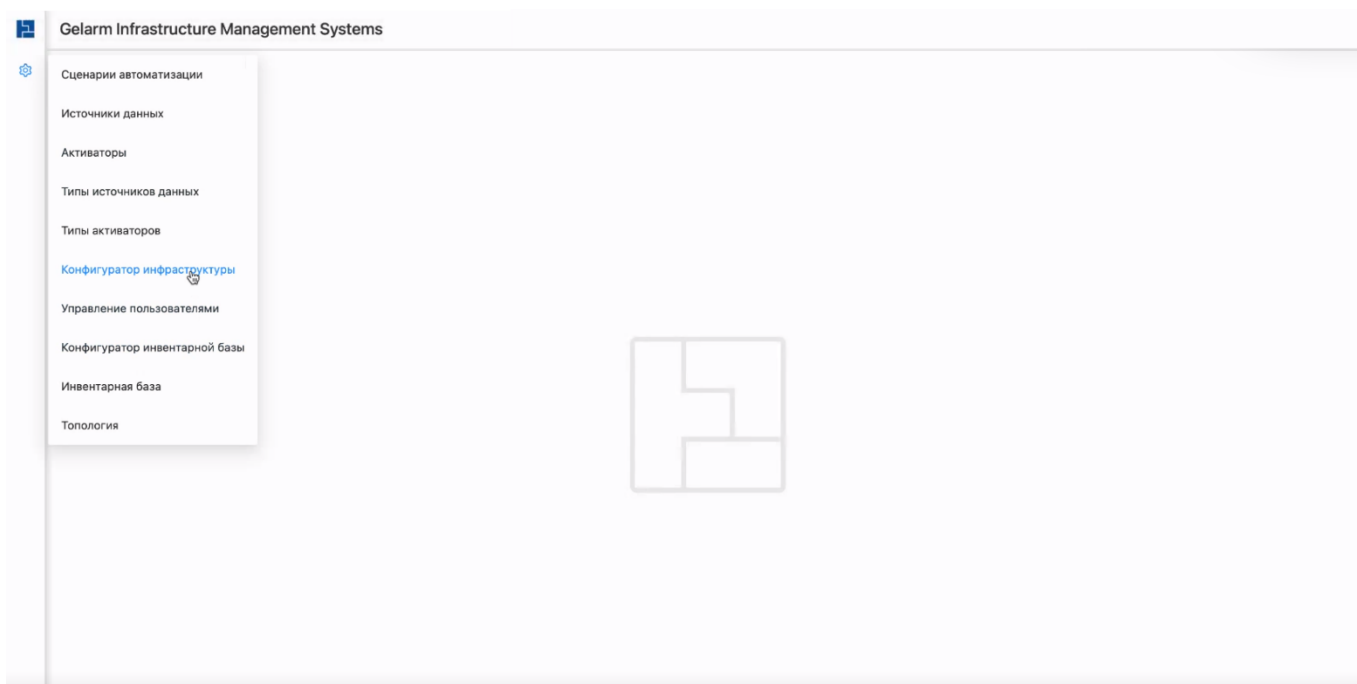
Примечание: После авторизации в Системе смените пароль на более безопасный. (Процесс смены пароля описан в Руководстве пользователя в разделе 2.3.8 «Управление пользователями»).

2.5 НАСТРОЙКА ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА AUTOMATION

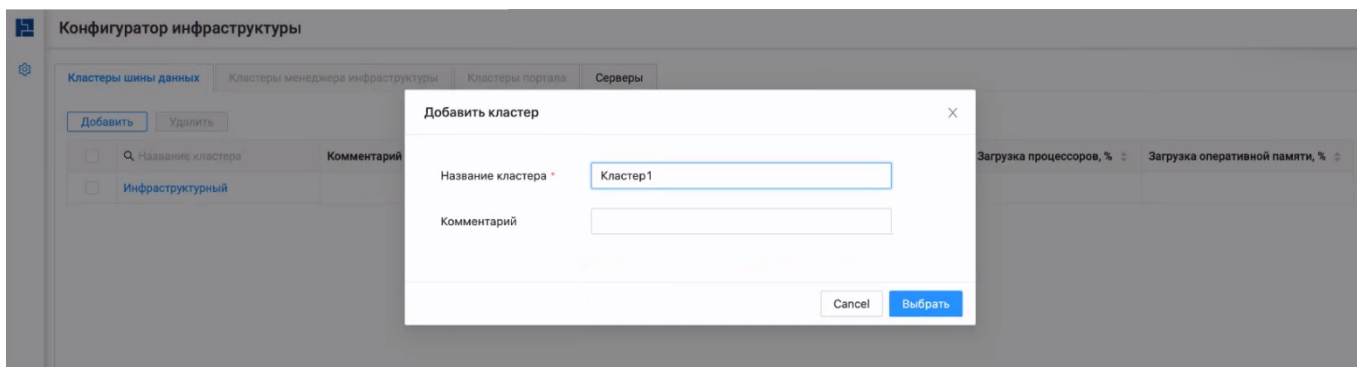
Шаг 1. Запустите веб-браузер и введите в адресную строку URL http://адрес_сервера.
Выполните авторизацию в системе.



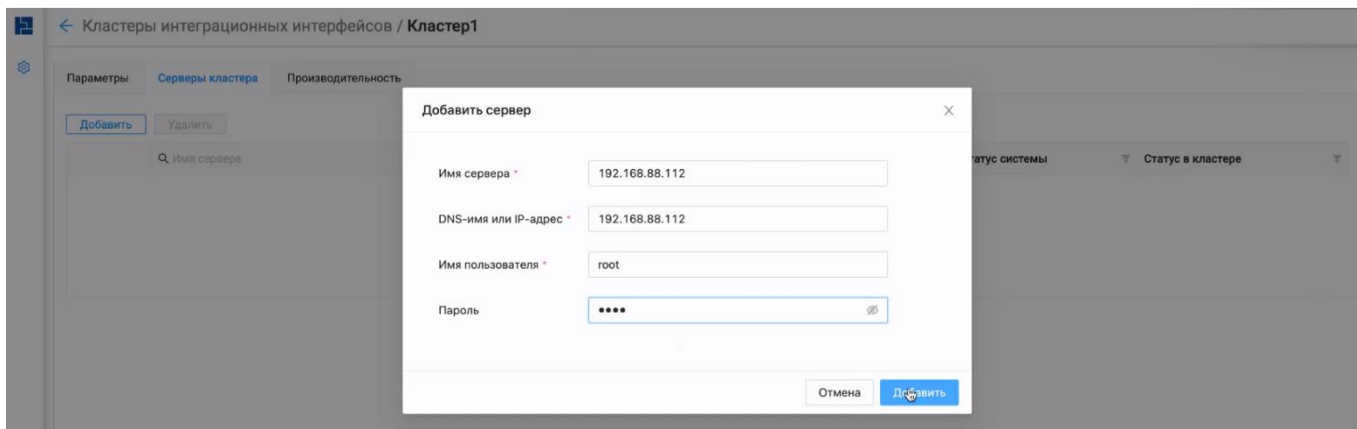
Шаг 2. В меню выберите пункт **Конфигуратор инфраструктуры**.



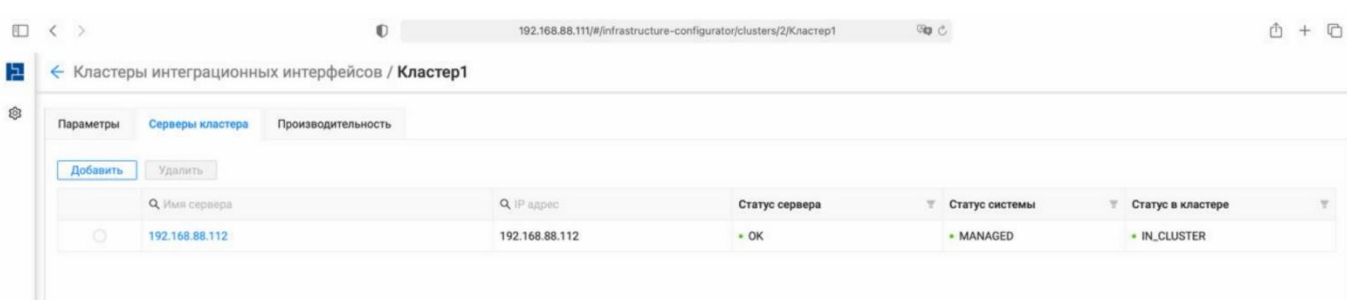
Шаг 3. На вкладке Параметры нажмите кнопку **Добавить** и в открывшейся форме введите название кластера. Нажмите кнопку **Выбрать**.



Шаг 4. На вкладке Серверы кластера нажмите кнопку **Добавить** и в открывшейся форме введите: Название сервера, DNS-имя или IP-адрес, Имя пользователя с правами root. Нажмите кнопку **Добавить**.



Шаг 5. Система выполнит проверку подключения к серверу. Как только сервер будет готов к эксплуатации в колонке **Статус в кластере** в строке с названием сервера будет статус **IN_CLUSTER**.



2.6 ОТКАТ К ПРЕДЫДУЩЕЙ ВЕРСИИ

Откат к предыдущей версии выполняется путем выполнения команды инсталляции дистрибутива предыдущей версии. Последовательность инсталляции идентична процессу описанному в разделе 2.3.



3 ТРЕБОВАНИЯ К ОБЕСПЕЧЕНИЮ МОНИТОРИНГА

3.1 ТРЕБОВАНИЯ К МОНИТОРИНГУ ПРОИЗВОДИТЕЛЬНОСТИ

№	ПАРАМЕТР МОНИТОРИНГА	УРОВЕНЬ КРИТИЧНОСТИ	ПОРОГОВЫЕ ЗНАЧЕНИЯ
1	Дисковое пространство	Critical	Значение $\geq 25\%$ – normal Менее 25% – warning Менее 5% – critical
2	Наличие памяти	Critical	Значение $\geq 25\%$ – normal Менее 25% – warning Менее 5% – critical
3	Загрузка процессора	Critical	Загрузка более 90% на всех CPU
4	Размер swap памяти	Critical	Значение $< 10\%$ – normal Выше 10% – warning

3.2 ТРЕБОВАНИЯ К МОНИТОРИНГУ КОМПОНЕНТ СИСТЕМЫ

№	ПАРАМЕТР МОНИТОРИНГА	УРОВЕНЬ КРИТИЧНОСТИ	КОНТРОЛИРУЕМЫЙ ПОКАЗАТЕЛЬ
1	Работа процессов БД	Critical	Сервис запущен и работает – normal Любое отклонение от normal – warning
2	Работоспособность RabbitMQ	Critical	Сервис запущен и работает – normal Любое отклонение от normal – warning
3	Работоспособность процессов портала управляемых через supervisor	Critical	Запрос - supervisorctl status all logview – в статусе отличном от RUNNING - статус Minor celery_monitor – в статусе отличном от RUNNING - статус Minor celery_automation – в статусе отличном от RUNNING – статус Major celery_beat – в статусе отличном от RUNNING - статус Major portal – в статусе отличном от RUNNING – статус Critical



			<pre># Пример ответа при корректной работе automation:celery_automation RUNNING pid 10864, uptime 0:00:16 automation:celery_beat RUNNING pid 10873, uptime 0:00:16 automation:celery_monitor RUNNING pid 10874, uptime 0:00:16 automation:portal RUNNING pid 10875, uptime 0:00:16 logview RUNNING pid 10863, uptime 0:00:16</pre>
4	PG balancer	Critical	Сервис запущен и работает – normal Любое отклонение от normal – warning

4 ТРЕБОВАНИЯ К БЕКАПИРОВАНИЮ СИСТЕМЫ

Бекапирование Системы рекомендуется осуществлять путем создания snapshot'a серверов, на которых развернута Система. Определение частоты бекапирования осуществляется на основании частоты изменений конфигураций системы.

На первых этапах эксплуатации системы рекомендуется выполнять бекапирование Системы с периодичностью раз в неделю.

5 ПРИЛОЖЕНИЕ

Ниже приведен перечень команд, которые выполняются системой при развертывании. Если установка системы производится не под пользователем root, пользователю необходимо предоставить права на выполнение ниже перечисленных команд.

Deploy:



Настройка пользователя и добавление ssh-ключа для gims

```
sudo useradd gims
sudo su - gims -c "chmod go+x ~"
sudo su - gims -c "mkdir .ssh"
sudo su - gims -c "chmod 700 .ssh"
sudo su - gims -c "ssh-keygen -f /home/gims/.ssh/id_rsa -N \"\" <<< y"
sudo su - gims -c "echo \"{public_key}\" >> ~/.ssh/authorized_keys"
sudo su - gims -c "chmod 700 .ssh/authorized_keys"
```

Настройка прав архива дистрибутива

```
sudo chown gims:gims /home/gims/gims-distrib.tgz
```

Настройка файла хоста

```
sudo hostnamectl set-hostname gims-automation-1-1
sudo chmod go+w /etc/hosts
sudo echo "{ip server} $(hostname -s)" >> /etc/hosts
```

Создание каталога, если его не существует

```
sudo mkdir -p /home/root
```

Настройка yum репозитория

```
sudo echo -e "[gims-contrib]\nname=GIMS\nbaseurl=file:///home/gims/gims-
contrib\nenabled=1\nngpgcheck=0" > /home/root/gims-contrib.repo
sudo mv -f /home/root/gims-contrib.repo /etc/yum.repos.d/
sudo chown root:root /etc/yum.repos.d/gims-contrib.repo
```

Установка компонентов системы из yum репозитория

```
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install python3
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install rabbitmq-server
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install java-1.8.0-openjdk
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install supervisor
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install postgresql12-server
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install pgbouncer
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y --noplugins groupinstall
"Development Tools"
sudo LANG=en_US.utf8 yum --disablerepo=* --enablerepo=gims-contrib -y install python3-devel
openldap-devel
```

Создание виртуального окружения python и установка пакетов питон из архива

```
sudo su - gims -c "python3 -m venv env"
sudo su - gims -c "source ~/env/bin/activate && pip install -r ~/gims-automation/requirements.txt --no-
index -f ~/gims-python-packages/"
```



Настройка rabbitmq-server

```
sudo systemctl enable rabbitmq-server --now
sudo rabbitmq-plugins enable rabbitmq_management
sudo rabbitmqctl add_user mqadmin adminpwd
sudo rabbitmqctl set_user_tags mqadmin administrator
sudo rabbitmqctl set_permissions -p / mqadmin ".*" ".*" ".*"
sudo systemctl restart rabbitmq-server
```

Настройка postgresql-12

```
sudo systemctl stop postgresql-12
sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
sudo cp /home/gims/gims-automation/contrib/postgresql/automation_cluster/pg_hba.conf
/var/lib/pgsql/12/data/
sudo cp /home/gims/gims-automation/contrib/postgresql/automation_cluster/postgresql.conf
/var/lib/pgsql/12/data/
sudo systemctl enable postgresql-12 --now
```

Настройка supervisord

```
sudo systemctl stop supervisord
sudo mkdir -p /var/log/celery
sudo mkdir -p /var/run/celery
sudo chown gims:gims /var/log/celery
sudo chown gims:gims /var/run/celery
sudo cp /home/gims/gims-automation/contrib/supervisor/automation.ini /etc/supervisord.d/
```

Настройка pgbouncer

```
sudo systemctl stop pgbouncer
sudo cp /home/gims/gims-automation/contrib/postgresql/automation_cluster/pgbouncer.ini
/etc/pgbouncer/
sudo touch /etc/pgbouncer/userlist.txt
sudo chmod 600 /etc/pgbouncer/userlist.txt /etc/pgbouncer/pgbouncer.ini
sudo chown pgbouncer:pgbouncer /etc/pgbouncer/userlist.txt /etc/pgbouncer/pgbouncer.ini
sudo systemctl enable pgbouncer --now
```

Отключение firewalld

```
sudo systemctl disable firewalld --now
```

Отключение selinux

```
sudo setenforce 0
sed -i 's/^SELINUX=.*SELINUX=disabled/' /etc/selinux/config
```



```
# Настройка linux limits для пользователя gims
sudo cp -f /etc/security/limits.conf /home/root/
sudo chmod o+w /home/root/limits.conf
sudo echo -e "gims hard nofile 4096\nngims soft nofile 4096\nngims hard nproc 8192\nngims soft nproc 8192" >> /home/root/limits.conf
sudo chmod o-w /home/root/limits.conf
sudo mv -f /home/root/limits.conf /etc/security/

# Настройка syslog
sudo cp /home/gims/gims-automation/contrib/syslog/remote.conf /etc/rsyslog.d/
sudo systemctl restart rsyslog

# Настройка logrotate
sudo cp -r /home/gims/gims-automation/contrib/logrotate/* /etc/logrotate.d/

# Настройка разрешения на использование python высоких портов без root
setcap 'cap_net_bind_service=+ep' /usr/bin/python3.6

# Настройка кластера rabbitmq-server
sudo rabbitmqctl set_cluster_name rabbit@MQ_Automation_Cluster_1
sudo rabbitmqctl set_policy --apply-to queues --priority 10 celery_tasks '^([a-z0-9]){32}_script_run'
'{"ha-mode":"exactly","ha-sync-mode":"automatic","ha-params":1,"expires":3600000}'
sudo rabbitmqctl set_policy --apply-to all --priority 1 ha "" '{"ha-mode":"exactly","ha-sync-
mode":"automatic","ha-params":2}'
sudo cat /var/lib/rabbitmq/.erlang.cookie
sudo echo -n "{hosts_file}" >> /etc/hosts
sudo mv -f /var/lib/rabbitmq/.erlang.cookie /var/lib/rabbitmq/.erlang.cookie_original
sudo cp -f /home/gims/gims-automation/contrib/rabbitmq/1/.erlang.cookie /var/lib/rabbitmq/
sudo chown rabbitmq:rabbitmq /var/lib/rabbitmq/.erlang.cookie
sudo chmod 400 /var/lib/rabbitmq/.erlang.cookie
sudo systemctl restart rabbitmq-server
sudo rabbitmqctl stop_app
sudo rabbitmqctl join_cluster rabbit@{hostname}
sudo rabbitmqctl start_app

# Настройка pgbouncer, postgresql-12 создание структуры и БД, логическая репликация
sudo su - postgres -c "psql -c \"CREATE USER {db_user_name} WITH PASSWORD
'{db_password}';\""
sudo su - postgres -c "psql -qAtX -F' ' -c \"select '\\\"' || rolname || '\\\" '\\\"' || rolpassword || '\\\"' from
pg_authid where rolpassword is not null\" > /etc/pgbouncer/userlist.txt
sudo mv -f /home/root/userlist.txt /etc/pgbouncer/
sudo chown pgbouncer:pgbouncer /etc/pgbouncer/userlist.txt
```



```
sudo chmod 600 /etc/pgbouncer/userlist.txt
sudo su - postgres -c "psql -c \"CREATE DATABASE {db_name} WITH OWNER {db_user_name}
ENCODING 'UTF8' LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8' TEMPLATE
template0;\""
sudo su - gims -c "cd ~/gims-automation && source ~/env/bin/activate && source .env/core.sh &&
python manage.py createcachetable"
sudo su - gims -c "cd ~/gims-automation && source ~/env/bin/activate && source .env/core.sh &&
python manage.py migrate"
sudo su - postgres -c "psql -d {db_name} -c \"CREATE SUBSCRIPTION gims_subscription_1_1
CONNECTION 'host={main_server_ip} port=5432 password={db_password} user={db_user_name}
dbname={main_db_name}' PUBLICATION gims_publication_1_1;\""
sudo ls /var/lib/pgsql/12/data/log/
sudo cat /var/lib/pgsql/12/data/log/{log_file} | grep ОШИБКА:
```

Запуск supervisord

```
sudo systemctl enable supervisord --now
sudo supervisorctl status automation:*
```

```
-----
UnDeploy:
-----
```

Остановка supervisord

```
sudo supervisorctl stop automation:*
sudo supervisorctl status automation:*
```

Отключение rabbitmq-server, возврат настроек

```
sudo rabbitmqctl stop_app
sudo rabbitmqctl reset
sudo systemctl disable rabbitmq-server --now
sed -i 's/^{ip server} .*//' /etc/hosts
sed -i 's/^{cluster ip server} .*//' /etc/hosts
sudo chmod go-w /etc/hosts
sudo mv -f /var/lib/rabbitmq/.erlang.cookie_original /var/lib/rabbitmq/.erlang.cookie
sudo rm -f /var/lib/rabbitmq/.erlang.cookie
sudo LANG=en_US.utf8 yum -y remove rabbitmq-server
sudo rm -rf /var/lib/rabbitmq
```

Отключение pgbouncer, возврат настроек

```
sudo systemctl disable pgbouncer --now
sudo rm -f /etc/pgbouncer/userlist.txt
```



```
sudo touch /etc/pgbouncer/userlist.txt
sudo chmod 600 /etc/pgbouncer/userlist.txt
sudo chown pgbouncer:pgbouncer /etc/pgbouncer/userlist.txt
sudo systemctl restart postgresql-12

# Отключение postgresql-12, возврат настроек, удаление структуры и БД
sudo cat /home/gims/gims-automation/.env/core.sh | grep "export DATABASE_"
sudo su - postgres -c "psql -d {db_name} -c \"DROP SUBSCRIPTION IF EXISTS
gims_subscription_1_1;\""
sudo su - postgres -c "psql -c \"DROP DATABASE IF EXISTS {db_name};\""
sudo su - postgres -c "psql -c \"DROP USER IF EXISTS {db_user_name};\""
sudo systemctl disable postgresql-12 --now

# Удаление yum репозитория
sudo rm -f /etc/yum.repos.d/gims-contrib.repo

# Отключение supervisord, возврат настроек
sudo systemctl disable supervisord --now
sudo rm -f /etc/supervisord.d/automation.ini

# Возврат настроек logrotate
sudo rm -f /etc/logrotate.d/remote /etc/logrotate.d/celery

# Возврат настроек rsyslog
sudo rm -f /etc/rsyslog.d/remote.conf
sudo systemctl restart rsyslog

# Удаление пользователя и каталога данных
sudo userdel -r -f gims

# Удаление логов системы
sudo rm -rf /var/log/celery /var/run/celery

# Возврат настроек linux limits для пользователя gims
sed -i 's/^gims .*//' /etc/security/limits.conf

# Удаление разрешения на использование python высоких портов без root
sudo setcap -r /usr/bin/python3.6
```