Приложение №5 к Техническому заданию

**Инструкция по эксплуатации Версия 1.0**

**Содержание**

[Содержание. 2](#_Toc38634819)

[Требования к программному обеспечению 4](#_Toc38634820)

[Сервера приложений 4](#_Toc38634821)

[Сервера БД 4](#_Toc38634822)

[Сервера файлов 4](#_Toc38634823)

[Сервер балансировки 5](#_Toc38634824)

[Сервер кеша 5](#_Toc38634825)

[Настройка программного обеспечения 6](#_Toc38634826)

[Сервера приложений 6](#_Toc38634827)

[1. Установка ПО 6](#_Toc38634828)

[2. Настройка ПО (php+nginx) 9](#_Toc38634829)

[3. Настройка ПО (minio) 16](#_Toc38634830)

[4. Перенос сайта 17](#_Toc38634831)

[5. Настройка прокси для подключения к БД 18](#_Toc38634832)

[6. Настройка ПО только на app1 (supervisor) 19](#_Toc38634833)

[7. Настройка ПО (NTP) 21](#_Toc38634834)

[Сервера БД 21](#_Toc38634835)

[1. Установка ПО 21](#_Toc38634836)

[2. Настройка ПО (etcd) 21](#_Toc38634837)

[3. Настройка ПО (stolon) 22](#_Toc38634838)

[4. Настройка ПО (NTP) 28](#_Toc38634839)

[Сервера файлов 28](#_Toc38634840)

[1. Установка ПО 28](#_Toc38634841)

[2. Настройка ПО 29](#_Toc38634842)

[Сервер балансировки 31](#_Toc38634843)

[1. Установка ПО 31](#_Toc38634844)

[2. Настройка ПО 31](#_Toc38634845)

[Сервер кеша 33](#_Toc38634846)

[1. Установка ПО: 33](#_Toc38634847)

[2. Настройка ПО: 33](#_Toc38634848)

[Инструкции 35](#_Toc38634849)

[Увеличение ресурсов серверов при увеличении нагрузки 35](#_Toc38634850)

[Заполнение БД 36](#_Toc38634851)

[Таблицы приложения 36](#_Toc38634852)

[Таблицы модулей Laravel 36](#_Toc38634853)

[Логи приложения 37](#_Toc38634854)

[Форматы логов 38](#_Toc38634855)

[Используемые логгеры 39](#_Toc38634856)

[Возможные проблемы при работе с логами 40](#_Toc38634857)

[Логи программного обеспечения 40](#_Toc38634858)

[Релизы приложения 41](#_Toc38634859)

**Требования к программному обеспечению**

Для всех типов серверов необходимое программное обеспечение указано в таблицах ниже.

**Сервера приложений**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название ПО** | **Разработчик** | **Лицензия** | **Стоимость** |
| ОС CentOS | The CentOS Project | GPL | Бесплатно |
| OpenSSH | Проект OpenBSD | BSD | Бесплатно |
| PHP | The PHP Group | PHP Licence | Бесплатно |
| Nginx | NGINX, Inc. | BSD-like | Бесплатно |
| Nrpe | Ethan Galstad | GPLv2 | Бесплатно |
| Zabbix Agent | Zabbix SIA | GPLv2 | Бесплатно |
| Laravel | The MIT Project | The MIT License (MIT) Copyright © Taylor Otwell | Бесплатно |
| Git client | Junio Hamano, Linus Torvalds | GPLv2 | Бесплатно |

**Сервера БД**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название ПО** | **Разработчик** | **Лицензия** | **Стоимость** |
| ОС CentOS | The CentOS Project | GPL | Бесплатно |
| OpenSSH | Проект OpenBSD | BSD | Бесплатно |
| PostgreSQL | Global Development Group | GPL | Бесплатно |
| Nrpe | Ethan Galstad | GPLv2 | Бесплатно |
| Zabbix Agent | Zabbix SIA | GPLv2 | Бесплатно |

**Сервера файлов**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название ПО** | **Разработчик** | **Лицензия** | **Стоимость** |
| ОС CentOS | The CentOS Project | GPL | Бесплатно |
| OpenSSH | Проект OpenBSD | BSD | Бесплатно |
| Minio | Apache License | Apache License Version 2.0 | Бесплатно |
| Nrpe | Ethan Galstad | GPLv2 | Бесплатно |
| Zabbix Agent | Zabbix SIA | GPLv2 | Бесплатно |

**Сервер балансировки**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название ПО** | **Разработчик** | **Лицензия** | **Стоимость** |
| ОС CentOS | The CentOS Project | GPL | Бесплатно |
| OpenSSH | Проект OpenBSD | BSD | Бесплатно |
| OpenSSL | The OpenSSL Project | OpenSSL License | Бесплатно |
| Nginx | NGINX, Inc. | BSD-like | Бесплатно |
| Nrpe | Ethan Galstad | GPLv2 | Бесплатно |
| Zabbix Agent | Zabbix SIA | GPLv2 | Бесплатно |

**Сервер кеша**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название ПО** | **Разработчик** | **Лицензия** | **Стоимость** |
| ОС CentOS | The CentOS Project | GPL | Бесплатно |
| OpenSSH | Проект OpenBSD | BSD | Бесплатно |
| Manticore | Manticore | GNU GENERAL PUBLIC LICENSE Version 2 | Бесплатно |
| Nrpe | Ethan Galstad | GPLv2 | Бесплатно |
| Zabbix Agent | Zabbix SIA | GPLv2 | Бесплатно |

**Настройка программного обеспечения**

Далее расписана настройка ПО для каждого типа сервера.

**Сервера приложений**

**1. Установка ПО**

1.1 Перемещаем приложенные файлы в **/root/web/**

1.2 **cd /root/web** # Заходим в каталог с ПО.

1.3 **rpm -ivh \*** или **yum install \*.rpm** # Устанавливаем ПО.

1.4 **vi /etc/systemd/system/php7.3-fpm.service** # Создаем юнит **systemd** для запуска **php**. Вставляем в него содержимое:

[Unit]

Description=The PHP 7.3 FastCGI Process Manager

Documentation=man:php-fpm7.3(8)

After=syslog.target network.target

[Service]

Type=notify

PIDFile=/tmp/php7.3-fpm.pid

ExecStart=/opt/remi/php73/root/sbin/php-fpm --nodaemonize --fpm-config /etc/opt/remi/php73/php-fpm.conf

ExecReload=/bin/kill -USR2 $MAINPID

[Install]

WantedBy=multi-user.target

1.5 **systemctl daemon-reload && systemctl enable php7.3-fpm && systemctl start php7.3-fpm && systemctl enable nginx && systemctl start nginx** # Добавляем в автозапуск и запускаем **php**, **nginx**, **memcached**.

1.6 Перемещаем архив с ПО **supervisor** в **/root/**

1.7 **tar -xf supervisor-4.1.0.tar.gz** # Распаковываем архив

1.8 **cd supervisor-4.1.0** # Заходим в распакованный каталог

1.9 **python setup.py install** # Устанавливаем supervisor

1.10 **vi /etc/rc.d/init.d/supervisord** # Вставляем в него содержимое:

#!/bin/bash

. /etc/init.d/functions

DAEMON=/usr/bin/supervisord

PIDFILE=/var/run/supervisord.pid

[ -x "$DAEMON" ] || exit 0

start() {

echo -n "Starting supervisord: "

if [ -f $PIDFILE ]; then

PID=`cat $PIDFILE`

echo supervisord already running: $PID

exit 2;

else

daemon $DAEMON --pidfile=$PIDFILE -c /etc/supervisord.conf

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && touch /var/lock/subsys/supervisord

return $RETVAL

fi

}

stop() {

echo -n "Shutting down supervisord: "

echo

killproc -p $PIDFILE supervisord

echo

rm -f /var/lock/subsys/supervisord

return 0

}

case "$1" in

start)

start

;;

stop)

stop

;;

status)

status supervisord

;;

restart)

stop

start

;;

\*)

echo "Usage: {start|stop|status|restart}"

exit 1

;;

esac

exit $?

1.11 **chmod 755 /etc/rc.d/init.d/supervisord** # Правим права.

1.12 **systemctl enable supervisord** # Добавляем в автозагрузку.

1.13 **vi /etc/supervisord.conf** # Вставляем в него содержимое:

[unix\_http\_server]

file=/tmp/supervisor.sock ; the path to the socket file

[supervisord]

logfile=/tmp/supervisord.log ; main log file; default $CWD/supervisord.log

logfile\_maxbytes=50MB ; max main logfile bytes b4 rotation; default 50MB

logfile\_backups=10 ; # of main logfile backups; 0 means none, default 10

loglevel=info ; log level; default info; others: debug,warn,trace

pidfile=/tmp/supervisord.pid ; supervisord pidfile; default supervisord.pid

nodaemon=false ; start in foreground if true; default false

minfds=1024 ; min. avail startup file descriptors; default 1024

minprocs=200 ; min. avail process descriptors;default 200

[rpcinterface:supervisor]

supervisor.rpcinterface\_factory = supervisor.rpcinterface:make\_main\_rpcinterface

[supervisorctl]

serverurl=unix:///tmp/supervisor.sock ; use a unix:// URL for a unix socket

files = /etc/supervisor/conf.d/\*.conf

1.14 **mkdir -p /etc/supervisor/conf.d** # Создаем папку для конфигов **supervisor**.

1.15 **crontab -e -u nginx** # настраиваем крон. Вставляем в него:

\* \* \* \* \* /usr/bin/php -f /var/www/energosbyt/artisan schedule:run >/dev/null 2>/dev/null

1.16 **vi /etc/profile** # Меняем настройки для прокси для ПО, добавляем в конце файла:

MY\_PROXY\_URL="http://10.5.164.222:8080/"

HTTP\_PROXY=$MY\_PROXY\_URL

HTTPS\_PROXY=$MY\_PROXY\_URL

FTP\_PROXY=$MY\_PROXY\_URL

http\_proxy=$MY\_PROXY\_URL

https\_proxy=$MY\_PROXY\_URL

ftp\_proxy=$MY\_PROXY\_URL

export HTTP\_PROXY HTTPS\_PROXY FTP\_PROXY http\_proxy https\_proxy ftp\_proxy

**2. Настройка ПО (php + nginx)**

2.1 **> /etc/opt/remi/php73/php.ini** # Чистим конфиг **php.ini**

2.2 **vi /etc/opt/remi/php73/php.ini** # Вставляем в него содержимое:

[PHP]

engine = On

short\_open\_tag = Off

precision = 14

output\_buffering = 4096

zlib.output\_compression = Off

implicit\_flush = Off

unserialize\_callback\_func =

serialize\_precision = -1

disable\_functions = pcntl\_wait,pcntl\_wifexited,pcntl\_wifstopped,pcntl\_wifsignaled,pcntl\_wifcontinued,pcntl\_wtermsig,pcntl\_wstopsig,pcntl\_signal\_get\_handler,pcntl\_get\_last\_error,pcntl\_strerror,pcntl\_sigprocmask,pcntl\_sigwaitinfo,pcntl\_sigtimedwait,pcntl\_exec,pcntl\_getpriority,pcntl\_setpriority,

disable\_classes =

zend.enable\_gc = On

expose\_php = Off

max\_execution\_time = 60

max\_input\_time = 60

memory\_limit = 128M

error\_reporting = E\_ALL & ~E\_DEPRECATED & ~E\_STRICT

display\_errors = Off

display\_startup\_errors = Off

log\_errors = On

log\_errors\_max\_len = 1024

ignore\_repeated\_errors = Off

ignore\_repeated\_source = Off

report\_memleaks = On

html\_errors = On

variables\_order = "GPCS"

request\_order = "GP"

register\_argc\_argv = Off

auto\_globals\_jit = On

post\_max\_size = 8M

auto\_prepend\_file =

auto\_append\_file =

default\_mimetype = "text/html"

default\_charset = "UTF-8"

doc\_root =

user\_dir =

enable\_dl = Off

file\_uploads = On

upload\_max\_filesize = 10M

max\_file\_uploads = 20

allow\_url\_fopen = On

allow\_url\_include = Off

default\_socket\_timeout = 600

[CLI Server]

cli\_server.color = On

[Date]

[filter]

[iconv]

[imap]

[intl]

[sqlite3]

[Pcre]

[Pdo]

[Pdo\_mysql]

pdo\_mysql.default\_socket=

[Phar]

[mail function]

SMTP = localhost

smtp\_port = 25

mail.add\_x\_header = Off

[ODBC]

odbc.allow\_persistent = On

odbc.check\_persistent = On

odbc.max\_persistent = -1

odbc.max\_links = -1

odbc.defaultlrl = 4096

odbc.defaultbinmode = 1

[Interbase]

ibase.allow\_persistent = 1

ibase.max\_persistent = -1

ibase.max\_links = -1

ibase.timestampformat = "%Y-%m-%d %H:%M:%S"

ibase.dateformat = "%Y-%m-%d"

ibase.timeformat = "%H:%M:%S"

[MySQLi]

mysqli.max\_persistent = -1

mysqli.allow\_persistent = On

mysqli.max\_links = -1

mysqli.default\_port = 3306

mysqli.default\_socket =

mysqli.default\_host =

mysqli.default\_user =

mysqli.default\_pw =

mysqli.reconnect = Off

[mysqlnd]

mysqlnd.collect\_statistics = On

mysqlnd.collect\_memory\_statistics = Off

[OCI8]

[PostgreSQL]

pgsql.allow\_persistent = On

pgsql.auto\_reset\_persistent = Off

pgsql.max\_persistent = -1

pgsql.max\_links = -1

pgsql.ignore\_notice = 0

pgsql.log\_notice = 0

[bcmath]

bcmath.scale = 0

[browscap]

[Session]

session.save\_handler = files

session.use\_strict\_mode = 0

session.use\_cookies = 1

session.use\_only\_cookies = 1

session.name = PHPSESSID

session.auto\_start = 0

session.cookie\_lifetime = 0

session.cookie\_path = /

session.cookie\_domain =

session.cookie\_httponly =

session.cookie\_samesite =

session.serialize\_handler = php

session.gc\_probability = 0

session.gc\_divisor = 1000

session.gc\_maxlifetime = 1440

session.referer\_check =

session.cache\_limiter = nocache

session.cache\_expire = 180

session.use\_trans\_sid = 0

session.sid\_length = 26

session.trans\_sid\_tags = "a=href,area=href,frame=src,form="

session.sid\_bits\_per\_character = 5

[Assertion]

zend.assertions = -1

[COM]

[mbstring]

[gd]

[exif]

[Tidy]

tidy.clean\_output = Off

[soap]

soap.wsdl\_cache\_enabled=1

soap.wsdl\_cache\_dir="/tmp"

soap.wsdl\_cache\_ttl=86400

soap.wsdl\_cache\_limit = 5

[sysvshm]

[ldap]

ldap.max\_links = -1

[dba]

[opcache]

[curl]

[openssl]

2.3 **> /etc/opt/remi/php73/php-fpm.d/www.conf** #Чистим конфиг **pool**

2.4 **vi /etc/opt/remi/php73/php-fpm.d/www.conf** #Вставляем в него содержимое:

[energosbyt]

user = nginx

group = nginx

listen = /run/php7.3-fpm.sock

listen.owner = nginx

listen.group = nginx

pm = static

pm.max\_children = 200

pm.start\_servers = 30

pm.min\_spare\_servers = 20

pm.max\_spare\_servers = 150

pm.max\_requests = 200

pm.status\_path = /status

slowlog = /var/opt/remi/php73/log/php-fpm/es-slow.log

request\_slowlog\_timeout = 10

request\_terminate\_timeout = 60

rlimit\_core = unlimited

catch\_workers\_output = yes

php\_admin\_value[error\_log] = /var/opt/remi/php73/log/php-fpm/es-error.log

php\_admin\_flag[log\_errors] = on

php\_admin\_value[disable\_functions] = pcntl\_fork,pcntl\_signal,pcntl\_waitpid,pcntl\_wexitstatus,pcntl\_async\_signals,pcntl\_alarm,pcntl\_fork,pcntl\_waitpid,pcntl\_signal\_dispatch,

php\_admin\_value[default\_socket\_timeout] = 60

2.5 **> /etc/nginx/nginx.conf** #Чистим конфиг **nginx**.

2.6 **vi /etc/nginx/nginx.conf** # Вставляем в него содержимое:

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log;

pid /run/nginx.pid;

include /usr/share/nginx/modules/\*.conf;

worker\_rlimit\_nofile 25000;

events {

worker\_connections 4096;

multi\_accept on;

}

http {

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 65;

types\_hash\_max\_size 2048;

server\_tokens off;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

include /etc/nginx/conf.d/\*.conf;

}

2.7 **vi /etc/nginx/conf.d/energosbyt.conf** # Создаем конфиг сайта:

server {

listen 80;

set\_real\_ip\_from 10.6.11.40;

real\_ip\_header X-Real-IP;

real\_ip\_recursive on;

server\_name .esplus.ru;

client\_max\_body\_size 1024M;

client\_body\_buffer\_size 4M;

fastcgi\_read\_timeout 60s;

fastcgi\_send\_timeout 60s;

fastcgi\_connect\_timeout 60s;

root "/var/www/energosbyt/public";

index index.php index.html index.htm;

error\_log /var/log/nginx/energosbyt.error.log;

access\_log /var/log/nginx/energosbyt.access.log;

set $php\_sock unix:/run/php7.3-fpm.sock;

location / {

try\_files $uri $uri/ /index.php$is\_args$args;

}

location ~\* /\.(svn|hg|git) { deny all; }

location ~ \.php$ {

try\_files $uri /index.php =404;

fastcgi\_pass $php\_sock;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

}

2.8 **systemctl restart nginx** # Перезапускаем **nginx**.

2.9 **systemctl restart php7.3-fpm** # Перезапускаем **php**.

2.10 **vi /etc/selinux/config**

меняем в этом файле строку **SELINUX=enforcing** на **SELINUX=disabled** # Отключаем **SELinux**

2.11 **firewall-cmd --zone=public --add-port=443/tcp —permanent & firewall-cmd --zone=public --add-port=80/tcp —permanent & firewall-cmd —reload** # Открываем порты для **http** и **https**

2.12 **mv /etc/localtime /etc/localtime.bak & ln -s /usr/share/zoneinfo/Europe/Moscow /etc/localtime & ntpdate pool.ntp.org & chkconfig ntpd on & service ntpd start** # Настраиваем дату и автоматическое обновление.

2.13 **vi /etc/opt/remi/php73/php.d/10-opcache.ini** # Меняем конфиг **opcache** на:

zend\_extension=opcache

opcache.enable=1

opcache.enable\_cli=1

opcache.memory\_consumption=400

opcache.interned\_strings\_buffer=8

opcache.max\_accelerated\_files=10000

opcache.max\_wasted\_percentage=5

opcache.use\_cwd=1

opcache.validate\_timestamps=1

opcache.revalidate\_freq=0

opcache.blacklist\_filename=/etc/opt/remi/php73/php.d/opcache\*.blacklist

opcache.huge\_code\_pages=0

**3. Настройка ПО (minio)**

3.1 **vi /etc/nginx/conf.d/minio.conf** # Создаем конфиг с содержимым

server {

listen 9000 default\_server;

server\_name \_;

location / {

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header Host $http\_host;

proxy\_connect\_timeout 300;

proxy\_http\_version 1.1;

proxy\_set\_header Connection "";

proxy\_pass http://minio;

proxy\_buffering off;

}

}

3.2 **vi /etc/nginx/conf.d/upstream.conf** # Создаем конфиг с содержимым

upstream minio {

include include/minio.conf;

}

3.3 **mkdir /etc/nginx/include** # Создаем каталог

3.4 **vi /etc/nginx/include/minio.conf** # Создаем конфиг с содержимым

server 10.5.3.33:9000 max\_fails=3 fail\_timeout=30s;

server 10.5.3.34:9000 max\_fails=3 fail\_timeout=30s;

server 10.5.3.35:9000 max\_fails=3 fail\_timeout=30s;

server 10.5.3.36:9000 max\_fails=3 fail\_timeout=30s;

3.5 **nginx -t** # Смотрим, что конфигурация **nginx** корректна.

3.6 **nginx -s reload** # Перезапускаем **nginx**.

3.7 **mv minio\_bin/mc /usr/local/bin/** # Перемещаем клиент для **minio**

3.8 **chmod a+x /usr/local/bin/mc** # Даем права

3.9 **mc config host add minio-es http://127.0.0.1:9000** AE2VV3W6ZMKT9OL2XZ80 k6NRWG8Ojlwk6ia+svqD5hmsW9PENHWxeH54JdBw # Добавляем наши сервера в конфиги.

Примеры команд:

**mc admin user add minio-es backup z4HUpqwdaj** # Так можно добавить пользователя

**mc admin user list minio-es** # Так можно посмотреть список пользователей.

**mc ls minio-es/** # Так можно посмотреть, что есть в хранилище

**mc rm minio-es/test/file** # Так можно удалить файл из бакета.

**4. Перенос сайта**

4.1 **mkdir -p /var/www/energosbyt/** # создаем каталог сайта

4.2 Переносим архив с сайтом в **/root/**

4.3 **tar -xf site.tar.gz** # Распаковываем.

4.4 **rsync -avl /root/site/ /var/www/energosbyt/** # Копируем содержимое сайта в его каталог.

4.5 **chown -R qsoft\_support:nginx /var/www/energosbyt/** # Поправляем все права на сайте.

4.6 **chmod 775 -R /var/www/energosbyt/** # Поправляем все права на сайте.

**5. Настройка прокси для подключения к БД**

5.1 Перемещаем приложенные файл (из архива **systemd**) по местам:

**/etc/systemd/system/stolon-keeper.service**

и

**/etc/systemd/system/stolon-proxy.service**

5.2 Перемещаем приложенные каталоги (из архива **stolon**) по местам:

**opt/stolon/** в **/opt/stolon/**

5.3 Приводим содержимое файла **/opt/stolon/etc/stolon/stolon-proxy** к данному виду:

PATH=$PATH:/usr/lib/postgresql/10/bin:/opt/stolon/bin

# default configuration values

STPROXY\_CLUSTER\_NAME=es-production

STPROXY\_LISTEN\_ADDRESS=127.0.0.1

STPROXY\_PORT=5432

# store configuration

STPROXY\_STORE\_BACKEND=etcdv3

STPROXY\_STORE\_ENDPOINTS=10.5.3.30:2380,10.5.3.31:2380,10.5.3.32:2380

# set this for debugging

# STPROXY\_DEBUG=true

# add any extra command line options to pass the stolon-keeper binary

EXTRA\_OPTS=

5.4 **useradd -r stolon** # Добавляем пользователя.

5.5 **systemctl daemon-reload && systemctl enable stolon-proxy && systemctl start stolon-proxy** # добавляем в автозагрузку и запускаем прокси.

5.6 **psql -U postgres -h 127.0.0.1 -p 5432** # Проверяем, что подключается к БД.

**6. Настройка ПО только на app1 (supervisor)**

6.1 **vi /etc/supervisor/conf.d/laravel-worker1.conf** # Создаем конфиг и вставляем в него:

[program:laravel-worker1]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/energosbyt/artisan queue:work --queue=tickets --tries=5

autostart=true

autorestart=true

user=nginx

numprocs=1

redirect\_stderr=true

stdout\_logfile=/var/www/energosbyt/storage/logs/supervisor/tickets.log

stopwaitsecs=600

6.2 **vi /etc/supervisor/conf.d/laravel-worker2.conf** # Создаем конфиг и вставляем в него:

[program:laravel-worker2]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/energosbyt/artisan queue:work --queue=segmentation

autostart=true

autorestart=true

user=nginx

numprocs=10

redirect\_stderr=true

stdout\_logfile=/var/www/energosbyt/storage/logs/supervisor/segmentation.log

6.3 **vi /etc/supervisor/conf.d/laravel-worker3.conf** # Создаем конфиг и вставляем в него:

[program:laravel-worker3]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/energosbyt/artisan queue:work --queue=notification

autostart=true

autorestart=true

user=nginx

numprocs=10

redirect\_stderr=true

stdout\_logfile=/var/www/energosbyt/storage/logs/supervisor/notification.log

6.4 **vi /etc/supervisor/conf.d/laravel-worker4.conf** # Создаем конфиг и вставляем в него:

[program:laravel-worker4]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/energosbyt/artisan queue:work --queue=default

autostart=true

autorestart=true

user=nginx

numprocs=1

redirect\_stderr=true

stdout\_logfile=/var/www/energosbyt/storage/logs/supervisor/default.log

stopwaitsecs=600

6.5 **vi /etc/supervisor/conf.d/laravel-worker5.conf** # Создаем конфиг и вставляем в него:

[program:laravel-worker5]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/energosbyt/artisan queue:work --queue=personalization

autostart=true

autorestart=true

user=nginx

numprocs=1

redirect\_stderr=true

stdout\_logfile=/var/www/energosbyt/storage/logs/supervisor/personalization.log

stopwaitsecs=1800

6.6 **vi /etc/supervisor/conf.d/laravel-worker6.conf** # Создаем конфиг и вставляем в него:

[program:laravel-worker6]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/energosbyt/artisan queue:work --queue=notification\_cycle

autostart=true

autorestart=true

user=nginx

numprocs=10

redirect\_stderr=true

stdout\_logfile=/var/www/energosbyt/storage/logs/supervisor/notification\_cycle.log

6.7 **systemctl restart supervisord & supervisorctl reread & supervisorctl update** # Перезагружаем

**7. Настройка ПО (NTP)**

7.1 **yum install ntp** # Устанавливаем

7.2 **vi /etc/ntp.conf** # Меняем содержимое конфига на:

driftfile /var/lib/ntp/drift

restrict default nomodify notrap nopeer noquery

restrict 127.0.0.1

restrict ::1

server 10.5.4.110

includefile /etc/ntp/crypto/pw

keys /etc/ntp/keys

disable monitor

7.3 **service restart ntp** # Перезагружаем NTP.

**Сервера БД**

**1. Установка ПО**

1.1 Перемещаем приложенные файлы в **/root/db/**

1.2 **cd /root/db** # Заходим в каталог с ПО.

1.3 **rpm -ivh \*** или **yum install \*.rpm** # Устанавливаем ПО.

**2. Настройка ПО (etcd)**

2.1 (ДЛЯ 1 СЕРВЕРА) **vi /etc/etcd/etcd.conf** # Вставляем в него содержимое:

ETCD\_NAME="etcd-db1"

ETCD\_LISTEN\_CLIENT\_URLS="http://172.27.8.36:2379"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://172.27.8.36:2379"

ETCD\_LISTEN\_PEER\_URLS="http://172.27.8.36:2380"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://172.27.8.36:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="prod-cluster"

ETCD\_INITIAL\_CLUSTER="etcd-db1=http://172.27.8.36:2380,etcd-db2=http://172.27.7.65:2380,etcd-cache=http://172.27.7.66:2380"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

ETCD\_DATA\_DIR="/var/lib/etcd"

ETCD\_ELECTION\_TIMEOUT="5000"

ETCD\_HEARTBEAT\_INTERVAL="1000"

ETCD\_QUOTA\_BACKEND\_BYTES="4294967296"

ETCD\_AUTO\_COMPACTION\_RETENTION="10"

ETCD\_AUTO\_COMPACTION\_MODE="revision"

2.2 (ДЛЯ 2 СЕРВЕРА) **vi /etc/etcd/etcd.conf** # Вставляем в него содержимое:

ETCD\_NAME="etcd-db2"

ETCD\_LISTEN\_CLIENT\_URLS="http://172.27.7.65:2379"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://172.27.7.65:2379"

ETCD\_LISTEN\_PEER\_URLS="http://172.27.7.65:2380"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://172.27.7.65:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="prod-cluster"

ETCD\_INITIAL\_CLUSTER="etcd-db1=http://172.27.8.36:2380,etcd-db2=http://172.27.7.65:2380,etcd-cache=http://172.27.7.66:2380"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

ETCD\_DATA\_DIR="/var/lib/etcd"

ETCD\_ELECTION\_TIMEOUT="5000"

ETCD\_HEARTBEAT\_INTERVAL="1000"

ETCD\_QUOTA\_BACKEND\_BYTES="4294967296"

ETCD\_AUTO\_COMPACTION\_RETENTION="10"

ETCD\_AUTO\_COMPACTION\_MODE="revision"

2.3 **systemctl enable etcd && systemctl start etcd** # Перезапускаем **etcd**.

**3. Настройка ПО (stolon)**

3.1 Перемещаем приложенные файлы (из архива **systemd**) по местам:

**/etc/systemd/system/stolon-keeper.service**

и

**/etc/systemd/system/stolon-sentinel.service**

3.2 Перемещаем приложенные каталоги (из архива **stolon**) по местам:

**opt/stolon/ в /opt/stolon/**

3.3 (ДЛЯ 1 СЕРВЕРА) Приводим содержимое файла **/opt/stolon/etc/stolon/stolon-keeper** к данному виду:

PATH=$PATH:/usr/pgsql-10/bin:/opt/stolon/bin

# set this to use a deterministic id for this host

STKEEPER\_ID=es\_pg1

# default configuration values

STKEEPER\_CLUSTER\_NAME=es-production

STKEEPER\_DATA\_DIR=/var/lib/stolon-keeper

STKEEPER\_LISTEN\_ADDRESS=172.27.8.36

STKEEPER\_PORT=5431

# store configuration

STKEEPER\_STORE\_BACKEND=etcdv3

STKEEPER\_STORE\_ENDPOINTS=172.27.8.36:2380,172.27.7.65:2380,172.27.7.66:2380

# set this for debugging

# STKEEPER\_DEBUG=true

# postgres configuration

STKEEPER\_PG\_LISTEN\_ADDRESS=172.27.8.36

STKEEPER\_PG\_PORT=25432

# specify postgres binary path if not in $PATH

# STKEEPER\_PG\_BIN\_PATH=/usr/pgsql-9.5/bin

# specify a non default user configuration directory

# STKEEPER\_PG\_CONF\_DIR=/var/lib/stolon/postgres/conf.d

# these values are required for keeper to instantiate and manage

# the database and should be identical across the cluster

STKEEPER\_PG\_REPL\_USERNAME=replication

STKEEPER\_PG\_REPL\_PASSWORDFILE=/opt/stolon/etc/stolon/secrets/pgpass.replication

STKEEPER\_PG\_SU\_USERNAME=postgres

STKEEPER\_PG\_SU\_PASSWORDFILE=/opt/stolon/etc/stolon/secrets/pgpass.postgres

# add any extra command line options to pass the stolon-keeper binary

EXTRA\_OPTS=

3.3 (ДЛЯ 2 СЕРВЕРА) Приводим содержимое файла **/opt/stolon/etc/stolon/stolon-keeper** к данному виду:

PATH=$PATH:/usr/pgsql-10/bin:/opt/stolon/bin

# set this to use a deterministic id for this host

STKEEPER\_ID=es\_pg2

# default configuration values

STKEEPER\_CLUSTER\_NAME=es-production

STKEEPER\_DATA\_DIR=/var/lib/stolon-keeper

STKEEPER\_LISTEN\_ADDRESS=172.27.7.65

STKEEPER\_PORT=5431

# store configuration

STKEEPER\_STORE\_BACKEND=etcdv3

STKEEPER\_STORE\_ENDPOINTS=172.27.8.36:2380,172.27.7.65:2380,172.27.7.66:2380

# set this for debugging

# STKEEPER\_DEBUG=true

# postgres configuration

STKEEPER\_PG\_LISTEN\_ADDRESS=172.27.7.65

STKEEPER\_PG\_PORT=25432

# specify postgres binary path if not in $PATH

# STKEEPER\_PG\_BIN\_PATH=/usr/pgsql-9.5/bin

# specify a non default user configuration directory

# STKEEPER\_PG\_CONF\_DIR=/var/lib/stolon/postgres/conf.d

# these values are required for keeper to instantiate and manage

# the database and should be identical across the cluster

STKEEPER\_PG\_REPL\_USERNAME=replication

STKEEPER\_PG\_REPL\_PASSWORDFILE=/opt/stolon/etc/stolon/secrets/pgpass.replication

STKEEPER\_PG\_SU\_USERNAME=postgres

STKEEPER\_PG\_SU\_PASSWORDFILE=/opt/stolon/etc/stolon/secrets/pgpass.postgres

# add any extra command line options to pass the stolon-keeper binary

EXTRA\_OPTS=

3.4 (ДЛЯ 1 СЕРВЕРА) Приводим содержимое файла **/opt/stolon/etc/stolon/stolon-sentinel** к данному виду:

PATH=$PATH:/usr/pgsql-10/bin:/opt/stolon/bin

# default configuration values

STSENTINEL\_CLUSTER\_NAME=es-production

STSENTINEL\_KEEPER\_PORT=5431

STSENTINEL\_LISTEN\_ADDRESS=172.27.8.36

STSENTINEL\_PORT=6431

# store configuration

STSENTINEL\_STORE\_BACKEND=etcdv3

STSENTINEL\_STORE\_ENDPOINTS=172.27.8.36:2380,172.27.7.65:2380,172.27.7.66:2380

# this is only used for initialisation

STSENTINEL\_INITIAL\_CLUSTER\_CONFIG=/opt/stolon/etc/stolon/cluster-config.json

# set this to force discovery type, eg: store, kubernetes

# STSENTINEL\_DISCOVERY\_TYPE=store

# set this for debugging

# STSENTINEL\_DEBUG=true

# add any extra command line options to pass the stolon-keeper binary

EXTRA\_OPTS=

3.4 (ДЛЯ 2 СЕРВЕРА) Приводим содержимое файла **/opt/stolon/etc/stolon/stolon-sentinel** к данному виду:

PATH=$PATH:/usr/pgsql-10/bin:/opt/stolon/bin

# default configuration values

STSENTINEL\_CLUSTER\_NAME=kia-production

STSENTINEL\_KEEPER\_PORT=5431

STSENTINEL\_LISTEN\_ADDRESS=172.27.7.65

STSENTINEL\_PORT=6431

# store configuration

STSENTINEL\_STORE\_BACKEND=etcdv3

STSENTINEL\_STORE\_ENDPOINTS=172.27.8.36:2380,172.27.7.65:2380,172.27.7.66:2380

# this is only used for initialisation

STSENTINEL\_INITIAL\_CLUSTER\_CONFIG=/opt/stolon/etc/stolon/cluster-config.json

# set this to force discovery type, eg: store, kubernetes

# STSENTINEL\_DISCOVERY\_TYPE=store

# set this for debugging

# STSENTINEL\_DEBUG=true

# add any extra command line options to pass the stolon-keeper binary

EXTRA\_OPTS=

3.5 Приводим содержимое файла **/opt/stolon/etc/stolon/secrets/pgpass.postgres** к данному виду:

t3y234DFG1TYLfds

3.6 Приводим содержимое файла **/opt/stolon/etc/stolon/secrets/pgpass.replication** к данному виду:

8bhzsdf7gXeBg

3.7 Приводим содержимое файла **/opt/stolon/etc/stolon/cluster-config.json** к данному виду:

{

"request\_timeout": "10s",

"sleep\_interval": "5s",

"keeper\_fail\_interval": "20s",

"max\_standbys\_per\_sender": 3,

"synchronous\_replication": true,

"init\_with\_multiple\_keepers": false,

"use\_pg\_rewind": false,

"pg\_parameters": {

"max\_connections": 300,

"shared\_buffers": "16GB",

"effective\_cache\_size": "48GB",

"maintenance\_work\_mem": "2GB",

"checkpoint\_completion\_target": 0.7,

"wal\_buffers": "16MB",

"default\_statistics\_target": 100,

"random\_page\_cost": 1.1,

"effective\_io\_concurrency": 200,

"work\_mem": "5242kB",

"min\_wal\_size": "1GB",

"max\_wal\_size": "2GB",

"max\_worker\_processes": 32,

"max\_parallel\_workers\_per\_gather": 16,

"max\_parallel\_workers": 32

}

}

3.8 **useradd -r stolon** # добавляем пользователя для **stolon**

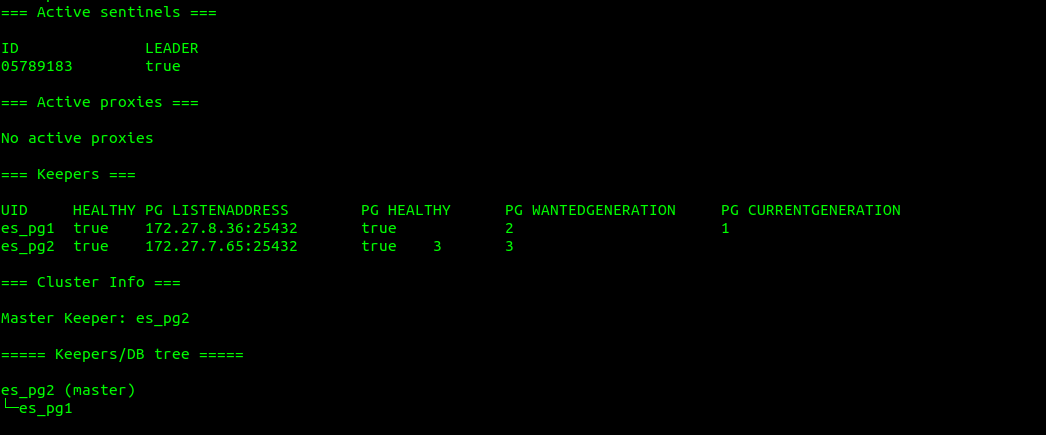
3.9 **mkdir /var/lib/stolon-keeper** # Создаем каталог для **stolon**.

3.10 **chown stolon. /var/lib/stolon-keeper** # Даем права каталогу.

3.11 **systemctl daemon-reload && systemctl enable stolon-keeper && systemctl enable stolon-sentinel && systemctl start stolon-keeper && systemctl start stolon-sentinel**# Добавляем в автозагрузку и запускаем.

3.12 **/opt/stolon/bin/stolonctl --store-backend etcdv3 --store-endpoints http://10.5.3.30:2379 --cluster-name es-production init** # Инициализируем **stolon**

3.13 **/opt/stolon/bin/stolonctl --store-backend etcdv3 --store-endpoints http://10.5.3.30:2379 --cluster-name es-production status** # Проверяем статус, должно быть так:

3.14 **/opt/stolon/bin/stolonctl --store-backend etcdv3 --store-endpoints http://10.5.3.30:2379 --cluster-name es-production update --patch '{ "pgParameters" : {"max\_connections" : "1200" } }'** # Так можно менять параметры БД.

**4. Настройка ПО (NTP)**

4.1 **yum install ntp** # Устанавливаем

4.2 **vi /etc/ntp.conf** # Меняем содержимое конфига на:

driftfile /var/lib/ntp/drift

restrict default nomodify notrap nopeer noquery

restrict 127.0.0.1

restrict ::1

server 10.5.4.110

includefile /etc/ntp/crypto/pw

keys /etc/ntp/keys

disable monitor

4.3 **service restart ntp** # Перезагружаем **NTP**.

**Сервера файлов**

**1. Установка ПО**

1.1 Перемещаем приложенные файлы в **/root/minio/**

1.2 **cd /root/minio** # Заходим в каталог с ПО.

1.3 **rpm -ivh \* или yum install \*.rpm** # Устанавливаем ПО.

1.4 **mv minio\_bin/\* /usr/local/bin/** # Копируем фалы для работы **minio**.

1.5 **chmod a+x /usr/local/bin/m\*** # Меняем права.

**2. Настройка ПО**

2.1 (для сервера 1) **vi /etc/default/minio** # Создаем конфиг с данным содержимым:

MINIO\_VOLUMES="http://10.5.3.33:9000/media/es http://10.5.3.34:9000/media/es http://10.5.3.35:9000/media/es http://10.5.3.36:9000/media/es"

MINIO\_OPTS="--address 10.5.3.33:9000"

# Access Key of the server.

MINIO\_ACCESS\_KEY=AE2VV3W6ZMKT9OL2XZ80

# Secret key of the server.

MINIO\_SECRET\_KEY=k6NRWG8Ojlwk6ia+svqD5hmsW9PENHWxeH54JdBw

MINIO\_STORAGE\_CLASS\_STATE=on

MINIO\_STORAGE\_CLASS\_STANDARD=EC:2

MINIO\_STORAGE\_CLASS\_RRS=EC:2

2.1 (для сервера 2) **vi /etc/default/minio** # Создаем конфиг с данным содержимым:

MINIO\_VOLUMES="http://10.5.3.33:9000/media/es http://10.5.3.34:9000/media/es http://10.5.3.35:9000/media/es http://10.5.3.36:9000/media/es"

MINIO\_OPTS="--address 10.5.3.34:9000"

# Access Key of the server.

MINIO\_ACCESS\_KEY=AE2VV3W6ZMKT9OL2XZ80

# Secret key of the server.

MINIO\_SECRET\_KEY=k6NRWG8Ojlwk6ia+svqD5hmsW9PENHWxeH54JdBw

MINIO\_STORAGE\_CLASS\_STATE=on

MINIO\_STORAGE\_CLASS\_STANDARD=EC:2

MINIO\_STORAGE\_CLASS\_RRS=EC:2

2.1 (для сервера 3) **vi /etc/default/minio** # Создаем конфиг с данным содержимым:

MINIO\_VOLUMES="http://10.5.3.33:9000/media/es http://10.5.3.34:9000/media/es http://10.5.3.35:9000/media/es http://10.5.3.36:9000/media/es"

MINIO\_OPTS="--address 10.5.3.35:9000"

# Access Key of the server.

MINIO\_ACCESS\_KEY=AE2VV3W6ZMKT9OL2XZ80

# Secret key of the server.

MINIO\_SECRET\_KEY=k6NRWG8Ojlwk6ia+svqD5hmsW9PENHWxeH54JdBw

MINIO\_STORAGE\_CLASS\_STATE=on

MINIO\_STORAGE\_CLASS\_STANDARD=EC:2

MINIO\_STORAGE\_CLASS\_RRS=EC:2

2.1 (для сервера 4) **vi /etc/default/minio** # Создаем конфиг с данным содержимым:

MINIO\_VOLUMES="http://10.5.3.33:9000/media/es http://10.5.3.34:9000/media/es http://10.5.3.35:9000/media/es http://10.5.3.36:9000/media/es"

MINIO\_OPTS="--address 10.5.3.36:9000"

# Access Key of the server.

MINIO\_ACCESS\_KEY=AE2VV3W6ZMKT9OL2XZ80

# Secret key of the server.

MINIO\_SECRET\_KEY=k6NRWG8Ojlwk6ia+svqD5hmsW9PENHWxeH54JdBw

MINIO\_STORAGE\_CLASS\_STATE=on

MINIO\_STORAGE\_CLASS\_STANDARD=EC:2

MINIO\_STORAGE\_CLASS\_RRS=EC:2

2.2 **vi /etc/systemd/system/minio.service** # Вставляем в него содержимое:

[Unit]

Description=MinIO

Documentation=https://docs.min.io

Wants=network-online.target

After=network-online.target

AssertFileIsExecutable=/usr/local/bin/minio

[Service]

WorkingDirectory=/usr/local/

User=minio

Group=minio

EnvironmentFile=/etc/default/minio

ExecStart=/usr/local/bin/minio server $MINIO\_OPTS $MINIO\_VOLUMES

# Let systemd restart this service always

Restart=always

# Specifies the maximum file descriptor number that can be opened by this process

LimitNOFILE=65536

# Disable timeout logic and wait until process is stopped

TimeoutStopSec=infinity

SendSIGKILL=no

[Install]

WantedBy=multi-user.target

2.3 **useradd -r minio** # Создаем пользователя для **minio**.

2.4 **mkdir /media/es** # Создаем каталог для файлов.

2.5 **chown -R minio. /media/es** # Даем ему нужные права.

2.6 **systemctl daemon-reload && systemctl enable minio && systemctl start minio** # Добавляем в автозагрузку и запускаем **minio**.

**Сервер балансировки**

**1. Установка ПО**

1.1 Перемещаем приложенные файлы в **/root/balans/**

1.2 **cd /root/balans** # Заходим в каталог с ПО.

1.3 **rpm -ivh \*** или **yum install \*.rpm** # Устанавливаем ПО.

1.4 **systemctl enable nginx && systemctl start nginx** # Добавляем в автозапуск и запускаем **nginx**.

**2. Настройка ПО**

2.1 **> /etc/nginx/nginx.con**f #Чистим конфиг **nginx**.

2.2 **vi /etc/nginx/nginx.conf** # Вставляем в него содержимое:

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log;

pid /run/nginx.pid;

include /usr/share/nginx/modules/\*.conf;

worker\_rlimit\_nofile 25000;

events {

worker\_connections 4096;

multi\_accept on;

}

http {

access\_log /var/log/nginx/access.log;

error\_log /var/log/nginx/error.log;

client\_max\_body\_size 2048m;

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 100s;

types\_hash\_max\_size 2048;

server\_tokens off;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

include /etc/nginx/conf.d/\*.conf;

upstream energosbyt {

ip\_hash;

server 10.6.11.41;

server 10.6.11.42;

}

server {

listen 80;

location / {

proxy\_pass http://energosbyt;

}

}

}

2.3 **systemctl restart nginx** # Перезапускаем **nginx**.

**Сервер кеша**

**1. Установка ПО:**

1.1 Перемещаем приложенные файлы в **/root/cache/**

1.2 **cd /root/cache** # Заходим в каталог с ПО.

1.3 **rpm -ivh \*** или **yum install \*.rpm** # Устанавливаем ПО.

1.4 **systemctl enable memcached && systemctl start memcached** # Добавляем в автозапуск и запускаем **memcached**.

**2. Настройка ПО:**

2.1 **> /etc/sysconfig/memcached** #Чистим конфиг **memcached**.

2.2 **vi /etc/sysconfig/memcached** # Вставляем в него содержимое:

PORT="11211"

USER="memcached"

MAXCONN="1024"

CACHESIZE="2048"

OPTIONS=""

2.3 **systemctl restart memcached** # Перезапускаем **memcached**.

2.4 **vi /etc/etcd/etcd.conf** # Вставляем в него содержимое:

ETCD\_NAME="etcd-cache"

ETCD\_LISTEN\_CLIENT\_URLS="http://10.5.3.32:2379"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.5.3.32:2379"

ETCD\_LISTEN\_PEER\_URLS="http://10.5.3.32:2380"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.5.3.32:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="prod-cluster"

ETCD\_INITIAL\_CLUSTER="etcd-db1=http://10.5.3.30:2380,etcd-db2=http://10.5.3.31:2380,etcd-cache=http://10.5.3.32:2380"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

ETCD\_DATA\_DIR="/var/lib/etcd"

ETCD\_ELECTION\_TIMEOUT="5000"

ETCD\_HEARTBEAT\_INTERVAL="1000"

ETCD\_QUOTA\_BACKEND\_BYTES="4294967296"

ETCD\_AUTO\_COMPACTION\_RETENTION="10"

ETCD\_AUTO\_COMPACTION\_MODE="revision"

2.5 (выполнять данную команду одновременно на **db1**, **db2**, **cache**) **systemctl enable etcd && systemctl start etcd** # Добавляем в автозапуск и запускаем **etcd**.

**Инструкции**

**Увеличение ресурсов серверов при увеличении нагрузки**

В таблице ниже указаны рекомендации по увеличению ресурсов серверов основанное на количестве пользователей в сутки. За начальное значение принята нагрузка из требований. Информация указана до увеличения пользователей в 2 раза от начального.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Компонент** | **Параметр** | **Количество пользователей в сутки** | | | | |
| **20000** | **25000** | **30000** | **35000** | **40000** |
| 1 | Сервера приложений №1, №2 | Кол-во ядер ЦПУ | 16 | 24 | 32 | 48 | 64 |
| ОЗУ (ГБ) | 16 | 16 | 24 | 32 | 32 |
| Диск (ГБ) | 250 | 250 | 250 | 250 | 250 |
| 2 | Сервера БД №1, №2 | Кол-во ядер ЦПУ | 32 | 32 | 32 | 32 | 32 |
| ОЗУ (ГБ) | 64 | 64 | 64 | 64 | 64 |
| Диск (ГБ) | 50 | 50 | 100 | 100 | 150 |
| 3 | Сервера файлов  №1, №2, №3, №4 | Кол-во ядер ЦПУ | 4 | 4 | 4 | 6 | 6 |
| ОЗУ (ГБ) | 8 | 8 | 8 | 12 | 12 |
| Диск (ГБ) | 120 | 120 | 160 | 160 | 200 |
| 4 | Сервер балансировки | Кол-во ядер ЦПУ | 4 | 4 | 4 | 4 | 4 |
| 4 | Сервер балансировки | ОЗУ (ГБ) | 4 | 4 | 4 | 4 | 4 |
| Диск (ГБ) | 20 | 20 | 20 | 20 | 20 |
| 5 | Сервер кеша | Кол-во ядер ЦПУ | 8 | 8 | 8 | 10 | 10 |
| ОЗУ (ГБ) | 20 | 20 | 24 | 28 | 32 |
| Диск (ГБ) | 20 | 20 | 30 | 30 | 40 |

**Заполнение БД**

**Таблицы приложения**

Для всех таблиц, которые были созданные непосредственно для ЛКК ЭнергосбыТ+ в качестве первичного ключа использованы колонки типа **uuid**. **Uuid** генерируются на уровне приложения, используется 4 версия (случайная генерация). Исходя из этого переполнения БД данными (как например могло быть при использовании **Int\*** типов с **auto\_increment**) – невозможно. Единственный ограничение лишь накладывает объем памяти на накопителях.

**Таблицы модулей Laravel**

Для системных таблиц, которые созданы для функционирования модулей **Laravel**:

1) Очереди:

У таблиц очередей (**jobs** и **failed\_jobs**) для первичного ключа используется тип **bigint** с **auto\_increment**. Теоретически, за время работы приложения переполнения произойти не может. Тем не менее, если будет достигнут лимит, то можно просто очистить таблицы (поскольку задачи не накапливаются, а выполняются сразу в фоне и при стабильном функционировании супервизора они будут пустыми), что сбросит величину **auto\_increment** на 1:

TRUNCATE TABLE jobs;

TRUNCATE TABLE failed\_jobs;

2) Миграции:

У таблицы миграций (**migrations**) для первичного ключа используется **int** с **auto\_increment**. На проекте никогда **не будет** **2 млрд.** миграций (за этап разработки их было **всего ~250**), так что нет смысла предусматривать переполнение таблицы.

3) Паспорт:

У таблиц токенов (**oauth\_access\_tokens, oauth\_auth\_codes, oauth\_refresh\_tokens**) ключами является строка – значение токена, модуль сам следить за уникальностью, переполнения невозможно. В таблицах клиентов (**oauth\_clients, oauth\_personal\_access\_clients**) уже созданы все необходимые клиенты, другие клиенты не предусмотрены.

**Логи приложения**

Все логи приложения расположены на обоих **app** по пути:

/var/www/energosbyt/storage/logs/

Некоторые **app** включаются в **.env** файле отдельно для каждого **app**:

/var/www/energosbyt/.env

После изменения **.env** файла нужно пересобрать кэш конфигов, поскольку при наличии кэша конфигов, **.env** не перечитывается при инициализации очередного процесса в приложении. Чтобы это сделать выполните команду из /var/www/energosbyt:

php artisan config:cache

Часть логов есть только на **app1**, часть и на **app1**, и на **app2** (будет указано для конкретного логгера). Если лог не включается через **.env**, то он пишется всегда. Для всех логгеров время ротации 30 дней (если не указана другая информация). Каждый день пишется в отдельный файл (с датой в названии).

**Форматы логов**

Логи имеют один из следующих форматов (будет указано для конкретного логгера):

1) [%datetime%] - %level\_name% %extra.memory\_usage% - [%uuid%]: %message%

где %datetime% - время в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС;

%level\_name% - текущая среда приложения;

%extra.memory\_usage% - объем памяти, потребляемый текущим процессом;

%uuid% - идентификатор процесса. Для каждого лога он свой (то есть лог, например, лог **payment** и лог **apple-pay** для одного процесса будут иметь разные идентификаторы). Все, что пишется в одинаковый лог внутри одного процесса имеет одинаковый **uuid.** Идентификатор необходим, чтобы можно было отличать логи разных процессов, которые одновременно используют одни и те же сервисы;

%message% - сообщение, которое пишется в лог. Может быть многострочным.

2) [%datetime%] - %level\_name%.%channel%: %message%

где %datetime% - время в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС;

%level\_name% - текущая среда приложения;

%channel% - канал информирования;

%message% - сообщение, которое пишется в лог. Может быть многострочным.

3) [%datetime%][%id%] %status%: %job%

где %datetime% - время в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС;

%id% - идентификатор задачи;

%status% - статус задачи;

%job% - название задачи.

**Используемые логгеры**

Описание всех логгеров, а точнее папок или конкретных файлов в некоторых случаях, куда логгер пишет логи:

1) app/ (app 1+2, формат 2) – общие ошибки приложения (например, ЛС уже зарегистрирован или неверный код подтверждения);

2) apple-pay/ (app 1+2, формат 1) – лог взаимодействия мобильного приложения с бекендом с банком при совершении платежей через **apple pay**;

3) bus/ (app 1+2, формат 1) – полный лог **xml** отправляемых в шину и получаемых из шины. Не рекомендуется держать включенным. Чтобы писались, нужно в **.env** на конкретном **app** включить **ES\_BUS\_DEBUG** в **true** и пересобрать кэш конфигов;

4) bus/bus-error.log (app 1+2, формат 2) – лог ошибок, пришедших из шины на уровне приложения. Не ротируется;

5) console/ (только app 1, формат 1) – лог работы фоновых задач на кроне (не путать с очередями). Например, очистка временных файлов;

6) default/ (app 1+2, формат 1) – на текущий момент используется только как заглушка для сервисов, где используется логирование, но не указан конкретный логгер. Таких сервисов сейчас нет, следовательно логов в этой папке (да и самой папки) быть не должно;

7) laravel.log (app 1+2, формат 2 с дополнительной трассировкой ошибки) – исключения, которые были не пойманы во время работы приложения. Не ротируется;

8) notifications/ (только app 1, формат 1) – логирование рассылки уведомлений, настроенных в административной части;

9) payment/ (app 1+2, формат 1) – логирование, связанное с оплатой (картой, авто платежи, рассылки по ним, в том числе и **apple pay**);

10) personalization/ (только app 1, формат 1) – логирование сегментирования пользователей;

11) send-notifications (app 1+2, формат 1) – полный лог писем, смс и пушей, которые отправляются пользователям. Не рекомендуется держать включенным. Чтобы писались, нужно в **.env** на конкретном **app** включить **EMAIL\_LOG\_ENABLED** в **true** для логирования писем, **SMS\_LOG\_ENABLED** в **true** для логирования смс и **PUSH\_LOG\_ENABLED** в **true** для логирования пушей и пересобрать кэш конфигов после любого изменения;

12) supervisor/ (только app 1, формат 3) – логирование работы фоновых задач в очередях. В названии лога указана очередь, к которой он относится. Не ротируются. В логи пишется вывод в консоль;

13) tickets/ (app 1+2, формат 1) – логирование запросов из шины по обновлению обращений.

**Возможные проблемы при работе с логами**

Если не хватает прав на чтение логов, то можно запустить скрипт для исправления прав и владельцев файлов на нужные на каждом **app**:

sudo /opt/priv.sh

**Логи программного обеспечения**

Также на серверах есть стандартные системные логи программного обеспечения. Подробная информация предоставлена в таблице.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Компонент | ПО | Уровень | Путь к логам |
| 1 | Сервера приложений №1, №2 | Nginx | Error | /var/log/nginx/energosbyt.error.log |
| Access | /var/log/nginx/energosbyt.access.log |
| Php-fpm | Error | /var/opt/remi/php73/log/php-fpm/error.log |
| 1 | Сервера приложений №1, №2 | Php-fpm | Slow | /var/opt/remi/php73/log/php-fpm/es-slow.log |
| Supervisor | Info | /var/log/supervisor/supervisord.log |
| Stolon-proxy | - | systemctl status stolon-proxy |
| 2 | Сервера БД №1, №2 | Stolon-keeper | - | systemctl status stolon-keeper |
| Stolon-proxy | - | systemctl status stolon-proxy |
| stolon-sentinel | - | systemctl status stolon-sentinel |
| 3 | Сервера файлов  №1, №2, №3, №4 | Minio | - | systemctl status minio |
| 4 | Cache сервер | Memcached | - | systemctl status memcached |

**Релизы приложения**

Обновления необходимо переносить и на **app1**, и на **app2.** Для переноса используется **git**, на обоих серверах он уже настроен на удаленный репозиторий **QSOFT**, в котором ведется разработка. Боевая ветка **master**.

Все следующие действия выполняются из папки **/var/www/energosbyt.**

Чтобы перенести новые правки, которые уже влиты в **master** в удаленном репозитории на каждом **app** необходимо выполнить:

**git pull origin master**

Затем, при необходимости, нужно вручную внести изменения в **.env** файл, установить обновления модулей, если они есть, пересобрать все оптимизационные кеши и поправить права на файлах. Для этого последовательно выполняются следующие команды:

**composer install --no-dev --optimize-autoloader**

**php artisan optimize**

**php artisan view:cache**

**sudo /opt/priv.sh**

Если скрипты и стили не были собраны заранее, то необходимо на каждом **app** запустить следующую команду для их сборки:

**npm run prod**

На **app1** дополнительно нужно установить миграции, если они есть. Для этого выполнить команду:

**php artisan migrate**

И перезапустить все процессы на супервизоре, для этого выполнить команду:

**sudo /opt/restart.sh**

Заказчик Исполнитель

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | / | К.Р. Азизов |  |  | / |  |
| м.п. |  |  |  | м.п. |  |  |