# Создание и настройка ВМ

ВМ размещаются на физических серверах afl-kvm11, afl-kvm12, afl-kvm21, afl-kvm22  
с именами afl-cms11, afl-cms12, afl-cms21, afl-cms22.

## Разбиение дисков

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ВМ** | **Устройство ВМ** | **Путь** | **Размер** | **Партиция/файл на хосте** | **Комментарий** |
| afl-cmsXX | /dev/vda1 | / | 100 GB | /srv/vm/X\_afl-cmsXX.raw |  |

## Параметры ВМ

|  |  |
| --- | --- |
| **Количество CPU** | 8 |
| **Объем ОЗУ** | 16384 Мб |

|  |  |  |  |
| --- | --- | --- | --- |
| **ВМ** | **IP-addr** | **GW** | **Hypervisor** |
| afl-cms11 | 192.168.80.12 | 192.168.80.1 | afl-kvm11 |
| afl-cms12 | 192.168.80.24 | 192.168.80.1 | afl-kvm12 |
| afl-cms21 | 192.168.81.12 | 192.168.81.1 | afl-kvm21 |
| afl-cms22 | 192.168.81.24 | 192.168.81.1 | afl-kvm22 |

## Инициализация ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22

### Настройка DNS

Настраиваем DNS на afl-site2:

vi /etc/bind/com.spb.hosts;  
afl-cms11 IN A 192.168.80.12  
afl-cms12 IN A 192.168.80.24  
afl-cms21 IN A 192.168.81.12  
afl-cms22 IN A 192.168.81.24  
  
/etc/init.d/bind9 reload;

Создание ВМ  
Присваиваем имя ВМ, создаем образ, запускаем ВМ согласно параметрам из таблицы в п.п. 1.2

VMNAME=2\_afl-cms11  
VMHOST=afl-cms11  
VMPART=[sdbXX](#_Для_виртуальных_машин:)  
VMDISK=/srv/vm/$VMNAME.raw  
VMDISKOPTS=sparse=false,size=100 # в гигабайтах  
#VMDISK=/dev/$VMPART  
#VMDISKOPTS=  
VCPUS=8  
RAM=16384  
VMIP=192.168.80.12  
VMGATEWAY=192.168.80.1  
MNT=$HOME/mnt  
  
virt-install --os-type=linux --machine=pc --name $VMNAME --vcpus $VCPUS --ram $RAM --disk path=$VMDISK,$VMDISKOPTS,bus=virtio --network=bridge:br0,model=virtio --hvm --virt-type=kvm --accelerate --graphics none --boot=hd,cdrom,menu=on -l http://afl-repo.com.spb.ru:8080/iso/boot/ --extra-args="

auto url=http://afl-repo.com.spb.ru/iso/boot/preseed/vm.seed

console=ttyS0,115200

mirror/country=RU

mirror/http/mirror=ru.archive.ubuntu.com

locale=en\_US.UTF-8

keyboard-configuration/layoutcode=en

netcfg/get\_ipaddress=$VMIP

netcfg/get\_gateway=$VMGATEWAY

netcfg/get\_netmask=255.255.255.0

netcfg/get\_nameservers=89.208.149.202 8.8.8.8

netcfg/disable\_autoconfig=true

netcfg/get\_hostname=new-install

netcfg/get\_domain=com.spb.ru";

Выключаем ВМ, извлекаем CD:

virsh shutdown $VMNAME;

# смотрим консоль, проверяем что ВМ выключилась

virsh console $VMNAME;

# выключаем ВМ

virsh destroy $VMNAME;

Добавляем интерфейс **tap**.

virsh edit $VMNAME

tapX - X заменить на порядковый номер ВМ.

<interface type='bridge'>

<mac address='52:54:00:eb:90:0e'/>

<source bridge='br0'/>

<target dev='tapX'/> <!-- ДОБАВИТЬ -->

<model type='virtio'/>

<address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>

</interface>

Выполняем сетевые настройки.

kpartx -a $VMDISK

mkdir -p $MNT;

#mount /dev/mapper/${VMPART}p1 $MNT;  
mount /dev/mapper/\*p1 $MNT;

echo $VMHOST > $MNT/etc/hostname;  
echo "$VMIP $VMHOST $VMHOST.com.spb.ru" > $MNT/etc/hosts;

mkdir -p $MNT/home/ubuntu/.ssh/;  
cat ~anovgorodov/.ssh/authorized\_keys ~mkostylev/.ssh/authorized\_keys >> $MNT/home/ubuntu/.ssh/authorized\_keys;

echo '%sudo ALL=(ALL:ALL) NOPASSWD: ALL' > $MNT/etc/sudoers.d/sudo\_nopasswd;  
chmod 0440 $MNT/etc/sudoers.d/sudo\_nopasswd;

sed -ie 's|^\(UUID=[^ ]\+[ ]\+\)/[ ]\+.\*|\1/ ext4 defaults,noatime,nodiratime,commit=30 0 1|' $MNT/etc/fstab;

echo 'APT::Install-Recommends "0";' > $MNT/etc/apt/apt.conf.d/norecommends;

umount $MNT;

kpartx -d $VMDISK;

Запускаем ВМ, добавляем в автозагрузку:

virsh start $VMNAME;

virsh autostart $VMNAME;

Настраиваем доступ к репозитории конфигов, генерируем ключ:

ssh ubuntu@$VMIP  
sudo bash  
  
ssh-keygen -N '' -f /root/.ssh/id\_rsa\_cfg\_replication -C cfg\_replication@$(hostname) && cat /root/.ssh/id\_rsa\_cfg\_replication.pub;

Заходим на afl-config (afl-site2)

echo -n 'command="/usr/bin/hg-ssh /home/config\_repo/review/\* /home/config\_repo/\*",no-pty,no-agent-forwarding,no-port-forwarding ' | sudo tee -a ~config\_pull/.ssh/authorized\_keys;  
sudo tee -a ~config\_pull/.ssh/authorized\_keys;

*<через консоль вставляем текст из id\_rsa\_cfg\_replication.pub>*

*Ctrl-D*

Проверяем корректность добавления

sudo tail -1 ~config\_pull/.ssh/authorized\_keys

### Подключение к Puppet

Конфигурируем puppetmaster для VM:

ssh puppet.com.spb.ru  
vi etc/puppet/manifests/nodes.pp  
  
#вместо $VMHOST подставить имя хоста  
node '$VMHOST' inherits basenode {  
 $config\_name = '$VMHOST'  
 include safe\_cmd  
 mailalias { 'root':  
 recipient => 'drdivano@morgan.com.spb.ru',  
 }  
 include postgres  
 postgres::initdb { "host": }  
 postgres::config { "host": }  
 postgres::enable { "host": }  
 postgres::hba { "host": }  
  
 package { 'libjpeg8': }  
 package { 'python-pkg-resources': }  
  
 include django\_cms::apache  
 include prod\_site::php5  
 package { 'libssl-dev': }  
  
 include backup  
 include backup::gnupg  
}

Включаем агент Puppet:

ssh ubuntu@$VMIP sudo puppet agent -t; #--debug;

На puppetmaster подписываем запрос на сертификат:

ssh puppet.com.spb.ru sudo puppet cert list;

ssh puppet.com.spb.ru sudo puppet cert sign $VMHOST;

На ВМ включаем агент puppet и запускаем конфигурирование:

ssh ubuntu@$VMIP sudo service puppet restart;

ssh ubuntu@$VMIP sudo puppet agent --enable;

ssh ubuntu@$VMIP sudo puppet agent -t; #--debug;

На puppetmaster смотрим отчет:

ssh puppet.com.spb.ru sudo report\_print.rb --log

## Настройка ОС и системного ПО

## Предварительная сборка стенда RC2

Для инициализации данных в Django CMS выполняем скрипт миграции данных из Drupal в Django CMS на стенде RC2:

#Хост afl-test-rc  
su - prod\_cms  
  
cp project/drupal-export/config.json.example project/drupal-export/config.json  
  
vi drupal-export/config.json  
{  
 "user" : "drupal",  
 "password": "XXXX",  
 "host": "127.0.0.1",  
 "database": "drupal"  
}  
  
#Настроить password на cms\_drupal  
su - old\_site;  
mysql mysql <<EOF  
GRANT ALL PRIVILEGES ON drupal.\* TO drupal@127.0.0.1 IDENTIFIED BY 'XXXX';  
FLUSH PRIVILEGES;  
EOF  
  
su - prod\_cms;

cd ~/cms-lib-src/mysql-connector-python-2.0.4;  
sudo python setup.py install;

AFL\_RC\_DIR=~/project;  
#RSYNC\_ARGS="-e ssh old\_site@192.168.83.24:";  
RSYNC\_ARGS="";  
DJANGO\_DATABASE\_NAME="prod\_cms";  
DJANGO\_DATABASE\_USER="prod\_cms";  
  
#Запуск скрипта миграции данных (длительная операция до 3-х часов)  
~/project/scripts/deploy/migrate.sh ${AFL\_RC\_DIR} "${RSYNC\_ARGS}";

## Установка пакетов

Установка пакетов на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22 выполняется Puppet. Автоматически устанавливаются следующие пакеты:

apt-get install libpq-dev;  
apt-get install python-psycopg2 python-dev libcurl4-openssl-dev;  
apt-get install libjpeg-dev libpng-dev libigraph0 libigraph-dev;  
apt-get install python-pkg-resources;

apt-get install postgresql;  
apt-get install apache2 libapache2-mod-wsgi;

## Настройка PostgreSQL

Настройку параметров postgresql производим на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22:

vi /etc/postgresql/9.3/main/postgresql.conf;

max\_connections = 200  
fsync = off  
full\_page\_writes = off

service postgresql restart;

## Установка Django CMS

Установку производим на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22:

useradd -m prod\_cms;

su - prod\_cms;

echo 'export PYTHONPATH=$HOME/cms-lib:$HOME/cms-lib-ubuntu14' >> ~/.profile;

# Для удобства миграции  
echo 'export AFL\_RC=192.168.83.24 # afl-test-rc' >> ~/.profile

. .profile;

ssh-keygen -C cms-migrate@$HOSTNAME -N '' -f $HOME/.ssh/id\_cms\_migrate;  
cat $HOME/.ssh/id\_cms\_migrate.pub;  
#Вставка в буфер

#на хосте $AFL\_RC:  
sudo -iu prod\_cms;  
tee -a $HOME/.ssh/authorized\_keys;  
#Копируем ключ через консоль из буфера  
**Ctrl-D**  
#На VM afl-cmsXX  
#Запоминаем номер ревизии CMS, собранной в п.п. 2.1  
REV=$(ssh -i .ssh/id\_cms\_migrate prod\_cms@$AFL\_RC 'cd ~/cms;svn info|grep Revision'|awk '{print $2}');

echo $REV;

svn co https://svn.com.spb.ru/afl\_rc/trunk/cms -r $REV cms;  
svn co https://svn.com.spb.ru/afl\_rc/trunk/lib -r $REV cms-lib;  
svn co https://svn.com.spb.ru/afl\_rc/trunk/lib-ubuntu14 -r $REV cms-lib-ubuntu14;  
svn co https://svn.com.spb.ru/afl\_rc/trunk/scripts -r $REV scripts;  
  
cp $HOME/cms/afl/settingslocal.py.example $HOME/cms/afl/settingslocal.py;

echo "DATABASES['default'] = {'ENGINE': 'django.db.backends.postgresql\_psycopg2', 'NAME': '$USER'}" >> $HOME/cms/afl/settingslocal.py;

# См. <https://docs.djangoproject.com/en/1.8/howto/deployment/checklist/>

dd if=/dev/urandom bs=40 count=1 | base64 > $HOME/cms/afl/secret-key;

chmod 640 $HOME/cms/afl/secret-key;

echo -e "\nSECRET\_KEY = open('$HOME/cms/afl/secret-key').read().strip()" >> $HOME/cms/afl/settingslocal.py;  
  
echo "DEBUG = False" >> $HOME/cms/afl/settingslocal.py;  
echo "TEMPLATE\_DEBUG = False" >> $HOME/cms/afl/settingslocal.py;

echo SITE\_URL = \"https://www.aeroflot.ru\" >> $HOME/cms/afl/settingslocal.py;  
echo SITE\_DOMAIN = \"www.aeroflot.ru\" >> $HOME/cms/afl/settingslocal.py;  
echo MOBILE\_SITE\_DOMAIN = \"m.aeroflot.ru\" >> $HOME/cms/afl/settingslocal.py;

cat >> $HOME/cms/afl/settingslocal.py <<- EOF  
BANNERS\_SETTINGS = {  
 "url": "%(protocol)s://www.aeroflot.ru/personal\_banners/v2.0.0/bulk",  
 "enforce\_https": False  
}  
EOF  
  
echo "CONN\_MAX\_AGE = 3600" >> $HOME/cms/afl/settingslocal.py;  
  
echo SECURE\_PROXY\_SSL\_HEADER = \(\'HTTP\_X\_FORWARDED\_PROTO\', \'https\'\) >> $HOME/cms/afl/settingslocal.py;

echo "MEDIA\_ROOT = '$HOME/media'" >> $HOME/cms/afl/settingslocal.py;

## Разворачивание cms-drupal (раздел AFL\_WOW)

Выполняем на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22:

useradd -m prod\_wow;  
su - prod\_wow;

svn co https://afl-site-svn.com.spb.ru/afl-site/cms/trunk/afl\_wow afl\_wow;

## Перенос данных из Drupal в Django CMS

## Инициализация БД (master Django-instance)

Базу данных (предварительно подготовленную в п.п. 2.1 Плана) копируем на ВМ afl-cms12.  
Примечание: после полной настройки Django CMS и вступления в силу запрета на внесение изменений в Drupal, стенд RC2 будет пересобран и данные с него будут повторно перенесены в Django CMS на afl-cms12.

#On the master django-instance only:  
su - postgres -c 'createuser -d prod\_cms';  
su - postgres -c 'createdb prod\_cms -O prod\_cms';  
#su - postgres -c 'psql prod\_cms -e "alter schema public owner to prod\_cms"';  
  
su - prod\_cms;  
  
ssh prod\_cms@$AFL\_RC -i /home/prod\_cms/.ssh/id\_cms\_migrate \  
 'pg\_dump -Ox prod\_cms | gzip -cf' > cms-dump.sql.gz;

#Если база не пустая, все удаляем:  
#psql -c 'drop schema public cascade;create schema public'  
pv cms-dump.sql.gz | gunzip -c | psql prod\_cms;

cd ~/cms;  
python manage.py collectstatic --noinput;

## Копирование статики

Каталог media (предварительно подготовленный в п.п. 2.1 Плана) копируем на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22:

su - prod\_cms;  
mkdir media  
rsync -auve 'ssh -i /home/prod\_cms/.ssh/id\_cms\_migrate' \  
 prod\_cms@$AFL\_RC:project/cms/media/\* /home/prod\_cms/media/;

## Настройка Apache - экземпляр Default

Apache настраивается через Puppet на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22. При установке конфигурации apache автоматически настраиваются все редиректы от cms-drupal (модифицированные под текущие пути доступа к каталогам), а также редиректы, связанные с переходом на cms-django.  
Для установки требуемой версии php настраиваем apt:

echo 'deb [arch=amd64 ] http://afl-repo.com.spb.ru:8080/ubuntu/ trusty local' > /etc/apt/sources.list.d/comspbru2.list;  
  
cat > /etc/apt/preferences.d/php5 <<-EOF;  
Package: php5 php5-common libapache2-mod-php5 libapache2-mod-php5filter php5-cgi php5-cli php5-dev php5-dbg php-pear php5-curl php5-gd php5-gmp php5-ldap php5-mhash php5-mysql php5-odbc php5-pgsql php5-pspell php5-recode php5-snmp php5-sqlite php5-sybase php5-tidy php5-xmlrpc php5-xsl php5-mcrypt  
Pin: version 5.2\*  
Pin-Priority: 1001  
EOF  
  
apt-get update;  
apt-get -y purge php5-common;  
puppet agent -t;  
  
su - prod\_wow;  
mkdir cgi-bin;  
chmod 755 cgi-bin;  
cp /usr/bin/php5-cgi cgi-bin/;  
  
#Проверяем версию  
cgi-bin/php5-cgi -v;  
PHP 5.2.17-3ubuntu~trusty with Suhosin-Patch 0.9.7

При работе Puppet автоматически устанавливаются и настраиваются следующие модули:

apt-get install apache2-suexec-custom;  
apt-get install libapache2-mod-wsgi;  
apt-get install libapache2-mod-fcgid;  
  
a2enmod proxy headers wsgi suexec;  
service apache2 restart;

## Настройка Apache - экземпляр Admin (мастер Django CMS)

Настройка выполняется на хосте ВМ afl-cms12 (мастер Django CMS).   
Дублируем инициализационные скрипты и конфигурацию для инстанса 'admin':

sudo sh /usr/share/doc/apache2/examples/setup-instance admin;  
sudo rm -rf /etc/apache2-admin;

Клонируем отдельную ветку для admin-инстанса:

sudo hg clone -b admin \  
 -e 'ssh -o BatchMode=yes -i /root/.ssh/id\_rsa\_cfg\_replication' \  
 ssh://config\_pull@afl-config//home/config\_repo/apache2-django-ubuntu14 \  
 /etc/apache2-admin;  
  
sudo /etc/init.d/apache2-admin start;  
update-rc.d apache2-admin defaults;

Добавляем пользователей для доступа к папке /mop (http basic authorization)

#On the master-django-instance only  
su - prod\_cms;  
vi ~/mop.htpasswd;  
<копируем из afl-drupal11:/home/prod\_site/mop.htpasswd>

## 2.9 Настройка crontab

Настройка выполняется на хосте ВМ afl-cms12 (мастер Django CMS).

cat >/etc/cron.d/cms\_publish\_ht <<EOF;  
SHELL=/bin/bash  
BASH\_ENV=/home/prod\_cms/.bashrc  
\*/30 \* \* \* \* prod\_cms source ~/.profile; cd cms && python manage.py publish\_ht > ~/cron.log 2>&1

EOF

# Настройка репликации базы данных

## Настройка репликации PostgreSQL

Файлы конфигурации PostgreSQL и SSL-сертификаты создаются при запуске Puppet.

## **Запуск репликации** **PostgreSQL**

На afl-cms12 (мастер Django CMS) настраиваем PostgreSQL:

vi /etc/postgresql/9.3/main/postgresql.conf;  
wal\_level = hot\_standby  
archive\_mode = on # allows archiving to be done  
 # (change requires restart)  
archive\_command = 'test ! -f /var/lib/postgresql/9.3/archiving\_active || cp -i %p /var/lib/postgresql/9.3/archive/%f'  
max\_wal\_senders = 3  
wal\_keep\_segments = 64  
  
vi /etc/postgresql/9.3/main/pg\_hba.conf  
host replication repl 192.168.80.12/32 md5  
host replication repl 192.168.81.12/32 md5  
host replication repl 192.168.81.24/32 md5  
  
createuser -s --replication -P repl;  
  
service postgresql restart;

На afl-cms11,21,22 (слейв) настраиваем PostgreSQL:

vi /etc/postgresql/9.3/main/postgresql.conf  
wal\_level = hot\_standby  
hot\_standby = on  
  
service postgresql stop;

Генерируем ключ afl-cms11,21,22, копируем ключ в буфер обмена.

su - postgres;  
ssh-keygen -C db\_replication@$(hostname) && cat ~/.ssh/id\_rsa.pub;  
<Копируем в буфер>

На afl-cms12 добавляем все ключи в файл ~postgres/.ssh/authorized\_keys:

su - postgres;  
tee -a ~/.ssh/authorized\_keys;  
<вставляем из буфера>  
Ctrl-D

На afl-cms11,21,22 (слейв) переносим базовый бекап с afl-cms12 (мастер), запускаем PostgreSQL:

su - postgres;  
pg\_basebackup -h afl-cms311 -D /var/lib/postgresql/9.3/backup -x -P -U repl;  
rsync -a --delete /var/lib/postgresql/9.3/backup/\* /var/lib/postgresql/9.3/main;  
  
cat > /var/lib/postgresql/9.3/main/recovery.conf <<- \_EOF1\_;  
standby\_mode = 'on'  
primary\_conninfo = 'host=afl-cms12 port=5432 user=repl password=PASSWD'  
trigger\_file = '/var/lib/postgresql/9.3/main/trigger'  
restore\_command = 'scp afl-cms311:/var/lib/postgresql/9.3/archive/%f "%p"'  
\_EOF1\_  
  
service postgresql start;

Проверяем, что репликация запустилась (psql должен вывести «t») и удаляем бэкап:

sudo -u postgres psql -tc 'select pg\_is\_in\_recovery()';

rm -fr /var/lib/postgresql/9.3/backup;

Выгружаем статику для cms на ВМ afl-cms11,21,22:

su - prod\_cms;  
cd ~/cms;  
python manage.py collectstatic --noinput;

## Настройка репликации файлов

Настраиваем репликацию файлов c afl-cms12 (master) на слейв-ВМ afl-cms11,21,22. Реплицируется папка ~prod\_cms/cms/media:

mkdir /var/log/cms

cat >/etc/cron.d/cms\_file\_sync <<EOF;  
\*/15 \* \* \* \* prod\_cms date >> /var/log/cms/sync-slave-$(date +\%F).log; \  
~/etc/sync\_prod\_site.sh >> /var/log/cms/sync-slave-$(date -I).log 2>&1

30 2 \* \* \* prod\_cms find /var/log/cms -type f -ctime +90 -delete;

EOF

su – prod\_cms  
mkdir etc  
  
cat >~/etc/sync\_prod\_site.sh <<EOF;  
#!/bin/sh –e  
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  
USER\_NAME=$(whoami)  
test $USER\_NAME = prod\_cms || {  
 echo "Script cannot be run from user $USER\_NAME, please run script from user prod\_site"  
 exit 1  
}  
test ${HOME##\*/} = prod\_cms || {  
 echo "HOME directory is not set correctly"  
 exit 2  
}  
cd  
FILES\_ALL="media"  
set +e  
for host in afl-cms11 afl-cms21 afl-cms22; do  
 for FILES in $FILES\_ALL ; do  
 rsync -av --del --no-perms $FILES/ $host:$FILES  
 done  
done  
  
EOF

# Мониторинг

## Мониторинг Nagios

Агент Nagios настраивается автоматически, с помощью Puppet.

На сервере afl-mon настраиваем мониторинг Nagios для ВМ afl-cms11,12,21,22.

1.4.1 Проверяем доступ с сервера мониторинга afl-mon:

ssh afl-mon.com.spb.ru

MHOST=<имя ВМ>

/usr/lib/nagios/plugins/check\_nrpe -H $MHOST -c check\_disk -a /

1.4.2 Настраиваем Nagios на сервере мониторинга afl-mon:

cd /etc/nagios3  
cp servers/afl-web1.cfg servers/$MHOST.cfg;

ln -s ../servers/$MHOST.cfg /etc/nagios3/servers-enabled/;

vi servers/$MHOST.cfg;

service nagios3 restart;

## Настройка мониторинга Django CMS

Системный администратор настраивает Nagios на хосте afl-mon для проверки работоспособности экземпляров Django, установленных на ВМ afl-cms11, afl-cms12, afl-cms21, afl-cms22. Для проверки используется специально установленный в Django CMS ping-плагин django-healthchecks 0.6.0 ([​https://pypi.python.org/pypi/django\_healthchecks](https://pypi.python.org/pypi/django_healthchecks)).

## Настройка мониторинга репликации

Системный администратор настраивает Nagios на хосте afl-mon на проверку активности следующих репликаций базы данных:  
afl-cms12 🡪 afl-cms11  
afl-cms12 🡪 afl-cms21  
afl-cms12 🡪 afl-cms22

## Агент Zabbix

1.5.1 Настраиваем zabbix-agent на ВМ afl-cms11,12,21,22:

cat >/etc/zabbix/zabbix\_agentd.conf <<EOF;  
PidFile=/var/run/zabbix/zabbix\_agentd.pid  
LogFile=/var/log/zabbix-agent/zabbix\_agentd.log  
LogFileSize=10  
DebugLevel=3  
EnableRemoteCommands=1  
Server=192.168.80.30,192.168.83.60,89.208.149.199  
StartAgents=3  
Timeout=25  
Include=/etc/zabbix/zabbix\_agentd.d/  
#ServerActive=127.0.0.1  
#Hostname=Zabbix server  
EOF  
  
mkdir /etc/zabbix/zabbix\_agentd.d;  
  
cat > /etc/zabbix/zabbix\_agentd.d/userparameter.conf <<EOF;  
UserParameter=http.py[\*],/usr/local/zabbix\_data\_mining/http.py "$1" "$2" "$3" "$4" "$5" "$6"  
UserParameter=iostat.py[\*],/usr/local/zabbix\_data\_mining/iostat.py "$1" "$2"  
UserParameter=memcache.sh[\*],/usr/local/zabbix\_data\_mining/memcache.sh "$1"  
UserParameter=apache2.py[\*],/usr/local/zabbix\_data\_mining/apache2.py "$1" "$2"  
UserParameter=postgres.py[\*],/usr/local/zabbix\_data\_mining/postgres.py "$1" "$2" "$3" "$4"  
UserParameter=netstat[\*],/bin/netstat -nt | grep -i $1 |awk '{print $$4}' | grep ":$2$$" | wc -l  
EOF  
  
service zabbix-agent restart;

1.5.2 Переносим скрипты zabbix (каталог /usr/local/zabbix\_data\_mining) c сервера afl-mon:

scp -r afl-mon.com.spb.ru:/usr/local/zabbix\_data\_mining /usr/local/;

# Настройка Backup

Скрипты бэкапа настраиваются автоматически через Puppet. После этого делается настройка в /usr/local/lib/backup. За основу берется настройка скриптов из afl-drupal11.

После установки необходимо отредактировать файл конфигурации options.sh:

ssh $VMIP;

sudo bash;

cd /usr/local/lib/backup;

cp options.sh.tmpl options.sh;

vi options.sh;

Бэкап базы данных делается ежедневно на afl-cms12 (на остальных хостах настроена репликация, бекап базы не требуется).   
Бэкап файлов: раз в 2 недели - полный бэкап, в остальные дни - дифференциальный бэкап.

#05 5 \* \* \* root nice ionice -c 3 pgsql\_db\_backup.sh

# полный бекап производится по нечетным неделям в субботу

55 3 \* \* 6 root expr `date +\%V` \% 2 >/dev/null && { purge\_backup.sh -f ; nice ionice -c 3 full\_backup.sh ; }

55 3 \* \* 6 root expr `date +\%V` \% 2 >/dev/null || { purge\_backup.sh ; nice ionice -c 3 diff\_backup.sh ; }

55 3 \* \* 0-5 root nice ionice -c 3 diff\_backup.sh ; purge\_backup.sh

На хосте afl-log4 настраиваем get-backups.sh и каталоги для архивов:

vi /var/backups/bin/get-backups.sh  
  
HOSTS="  
 kvm11 drupal11 cms11  
 kvm12 drupal12 mc12 cms12  
 kvm13 activity13 anketa13 app13 board13 map13  
 kvm14 app14 support ardoor mcpg14  
 kvm21 drupal21 cms21  
 kvm22 drupal22 mc22 activity22 cms22  
 kvm23 app23  
 kvm24 app24 mcpg24  
 kvm31 mc31  
 kvm32 test32  
 kvm41 mc41  
 kvm42 activity puppet test-rc  
 mon  
 site2  
"  
  
su - backup;  
  
mkdir /srv/backup2/afl-cms{11,12,21,22};  
  
cd /srv/backup;  
ln -s ../backup2/afl-cms11 afl-cms11 && ln -s ../backup2/afl-cms12 afl-cms12 && ln -s ../backup2/afl-cms21 afl-cms21 && ln -s ../backup2/afl-cms22 afl-cms22;

На ВМ afl-cms11,12,21,22 настраиваем учетную запись backup:

usermod -s /bin/sh backup;  
  
mkdir /var/backups/.ssh;  
chown root:backup /var/backups/.ssh;  
chmod 750 /var/backups/.ssh;  
mkdir /var/backups/bin;  
  
cat > /var/backups/bin/rsync-wrapper.sh <<- \_EOF1\_  
#!/bin/sh  
case "\$SSH\_ORIGINAL\_COMMAND" in  
 \*\&\*|\*\(\*|\*\{\*|\*\;\*|\*\>\*|\*\\\`\*|\*\|\*) echo Rejected ; exit 1 ;;  
 rsync\ --server\ --sender\*) \$SSH\_ORIGINAL\_COMMAND ;;  
 \*) echo Rejected ; exit 1 ;;  
esac  
\_EOF1\_  
  
chmod a+x /var/backups/bin/rsync-wrapper.sh;  
  
su - backup;  
  
cat > /var/backups/.ssh/authorized\_keys <<- \_EOF1\_  
command="/var/backups/bin/rsync-wrapper.sh" ssh-dss  backup@afl-log4  
\_EOF1\_

# Финальная сборка стенда RC2

По согласованию с Заказчиком после введения запрета на внесение изменений в CMS Drupal выполняется:

* подготовка дампа БД Drupal
* сборка стенда RC2 (п.п. 2.1 Плана).

# Финальный перенос данных в Django CMS

Повторно выполняются шаги 2.6.1 и 2.6.2 из плана.

# Обновление учетных записей в базе для сотрудников РИ

На afl-cms12 запускаем скрипт по смене паролей для учетных записей сотрудников РИ (таблица auth\_user).   
Копируем ключ на afl-mon (для доступа к скрипту отправки смс):

su - prod\_cms;  
cat ~/.ssh/id\_cms\_migrate.pub;  
<Копируем в буфер>

На afl-mon добавляем ключ в файл ~root/.ssh/authorized\_keys:

tee -a /root/.ssh/authorized\_keys;  
<вставляем из буфера>  
Ctrl-D

На основе Приложения 2 системный администратор готовит ri.csv файл в формате:  
username,password,pager  
foo,xxyyzz,9650000000  
  
Для генерации пароля использовать команду:

pwgen 10 1

Проверяем команду в скрипте по отправке смс:

su - prod\_cms;  
  
vi ~/scripts/deploy/change\_psw\_with\_send\_sms.sh;  
#Поправляем, если необходимо команду и текст:  
ssh -i ~/.ssh/id\_cms\_migrate root@afl-mon /usr/local/lib/nagios/send\_sms.sh random $pager \"New password for admin.aeroflot.ru: $password\";

Запускаем скрипт по смене паролей и рассылке смс на afl-cms12:

~/scripts/deploy/change\_psw\_with\_send\_sms.sh /home/prod\_cms/ri.csv

# Тестирование

Тестеры выполняют тестирование Django CMS и критических процессов Сайта Аэрофлота в соответствии с тест-кейсами, выложенными в релизную заявку.

Тестирование раздела администрирования Django CMS выполняется по адресу admin.aeroflot.ru. Тестирование пользовательского раздела Django CMS выполняется по адресу www.aeroflot.ru. Для доступа к адресам системный администратор настраивает varnish на afl-cms311:

vi /etc/varnish/backends.vcl  
  
backend default {  
 .host = "192.168.80.24"; #django afl-cms12  
 .port = "8080";  
 .connect\_timeout = 300s;  
 .first\_byte\_timeout = 300s;  
 .between\_bytes\_timeout = 300s;  
 .probe = {  
 .url = "/ru-ru/news/rss";  
 .timeout = 20s;  
 .interval = 120s;  
 .window = 5;  
 .threshold = 4;  
 }  
}  
backend reserve {  
 .host = "192.168.81.24"; #django afl-cms22  
 .port = "8080";  
 .connect\_timeout = 600s;  
 .first\_byte\_timeout = 600s;  
 .between\_bytes\_timeout = 600s;  
 .probe = {  
 .url = "/ru-ru/news/rss";  
 .timeout = 20s;  
 .interval = 120s;  
 .window = 5;  
 .threshold = 4;  
 }  
}  
backend cms\_master {  
 .host = "192.168.80.24"; #django afl-cms12  
 .port = "8080";  
 .connect\_timeout = 300s;  
 .first\_byte\_timeout = 300s;  
 .between\_bytes\_timeout = 300s;  
}  
  
/etc/init.d/varnish restart

Тестировщики прописывают в файле hosts:

185.69.82.13 www.aeroflot.ru aeroflot.ru www.aeroflot.com aeroflot.com m.aeroflot.ru pda.aeroflot.ru m.www.aeroflot.ru admin.aeroflot.ru

По результатам тестирования принимается решение о переключении трафика с CMS Drupal на Django CMS. Принятое решение согласуется с Заказчиком.

# Обновление учетных записей в базе для сотрудников Аэрофлота на время параллельной эксплуатации

На afl-cms12 запускаем скрипт по смене паролей для учетных записей сотрудников Аэрофорта (таблица auth\_user). Скрипт меняет пароли и рассылает смс.

На основе Приложения 3 системный администратор готовит afl.csv файл в формате:  
username,password,pager  
foo,xxyyzz,9650000000  
  
Для генерации пароля использовать команду:

pwgen 10 1

Запускаем скрипт по смене паролей и рассылке смс на afl-cms12:

su – prod\_cms  
~/scripts/deploy/change\_psw\_with\_send\_sms.sh /home/prod\_cms/afl.csv

# Переключение трафика на cms-Django

По согласованию с Аэфрофлотом системный администратор выполняет переключение трафика с cms-Drupal на cms-Django. Переключение производится путем смены в Varnish ветки в Mercurial на production\_rc. Операция выполняется одновременно на гипервизорах afl-kvm11,12,21,22:

ssh afl-kvm12 'cd /etc/varnish; sudo hg update production\_rc; sudo /etc/init.d/varnish restart';

ssh afl-kvm11 'cd /etc/varnish; sudo hg update production\_rc; sudo /etc/init.d/varnish restart';

ssh afl-kvm21 'cd /etc/varnish; sudo hg update production\_rc; sudo /etc/init.d/varnish restart';

ssh afl-kvm22 'cd /etc/varnish; sudo hg update production\_rc; sudo /etc/init.d/varnish restart';

После переключения распределение потоков клиентского трафика происходит следующим образом:

1. Varnish на afl-kvm11,12:   
   - основной инстанс cms-django - afl-cms12 (гипервизор afl-kvm12)   
   - резервный инстанс cms-django - afl-cms22 (гипервизор afl-kvm22)
2. Varnish на afl-kvm21,22:   
   - основной инстанс cms-django - afl-cms22 (гипервизор afl-kvm22)   
   - резервный инстанс cms-django - afl-cms12 (гипервизор afl-kvm12)
3. Для всех Varnish на домене "admin.aeroflot.ru" установлен специальный бэкенд "cms\_admin" - это инстанс, который отдельно запускается на afl-cms12 с отдельным apache-инстансом на порту 8081:  
   config.vcl:

if ( req.http.host == "admin.aeroflot.ru" ) {  
 if (req.http.x-forwarded-proto != "https" && (req.url ~ "^/(|../|..-../)admin" || req.url ~ "[?|&]edit") ) {  
 set req.http.x-Redir-Url = "https://" + req.http.host + req.url;  
 error 800 req.http.x-Redir-Url;

}  
 set req.backend = cms\_admin;  
 if (req.http.cookie ~ "sessionid" || req.http.cookie ~ "csrftoken") {  
 return (pass);  
 }  
 } else {  
 if (req.url ~ "^/(|../|..-../)admin" || req.url ~ "[?|&]edit" ) {  
 error 404 "Not Found";  
 }  
 }

backend.vcl:

backend cms\_admin {  
 .host = "192.168.80.22"; #afl-cms12  
 .port = "8081";  
 .connect\_timeout = 300s;  
 .first\_byte\_timeout = 300s;  
 .between\_bytes\_timeout = 300s;  
}

## Внесение правок на страницах Drupal

Администратор Drupal вносит следующие изменения на страницах Drupal:

1. Скрывает баннеры и блок подписки на уведомления о спецпредложениях на страницах раздела ЭТП (для всех языков):
   1. <http://www.aeroflot.ru/cms/etp/news/active>
   2. [http://www.aeroflot.ru/cms/etp/news/active/item/{номер лота}](http://www.aeroflot.ru/cms/etp/news/active/item/%7bномер%20лота%7d)
   3. <http://www.aeroflot.ru/cms/etp/news/results>
   4. [http://www.aeroflot.ru/cms/etp/news/archive/{параметры запроса}](http://www.aeroflot.ru/cms/etp/news/archive/%7bпараметры%20запроса%7d)
   5. [http://www.aeroflot.ru/cms/etp/news/results/item/{номер лота}](http://www.aeroflot.ru/cms/etp/news/results/item/%7bномер%20лота%7d)
2. Скрывает баннеры и блок подписки на уведомления о спецпредложениях на страницах калькуляторов набора и траты миль (для всех языков):
   1. <http://www.aeroflot.ru/cms/calculator>
   2. <http://www.aeroflot.ru/cms/calculator_awards>

# Переключение трафика c cms-Django на cms-Drupal в случае непредвиденного сбоя

Переключение производится путем смены в Varnish ветки в Mercurial на production.  
Операция выполняется одновременно на гипервизорах afl-kvm11,12,21,22:

ssh afl-kvm12 'cd /etc/varnish; sudo hg update production; sudo /etc/init.d/varnish restart';

ssh afl-kvm11 'cd /etc/varnish; sudo hg update production; sudo /etc/init.d/varnish restart';

ssh afl-kvm21 'cd /etc/varnish; sudo hg update production; sudo /etc/init.d/varnish restart';

ssh afl-kvm22 'cd /etc/varnish; sudo hg update production; sudo /etc/init.d/varnish restart';

# Распределение повышенной нагрузки

В случае возникновения повышенной нагрузки от клиентского трафика (определяется по результатам мониторинга работы инстансов cms-django) предусматривается следующая схема распределения потоков (каждый варниш настраивается на свой cms-инстанс):  
- varnish afl-kvm11: основной инстанс - afl-cms11, запасной - afl-cms21  
- varnish afl-kvm12: основной инстанс - afl-cms12, запасной - afl-cms22  
- varnish afl-kvm21: основной инстанс - afl-cms21, запасной - afl-cms11  
- varnish afl-kvm22: основной инстанс - afl-cms22, запасной - afl-cms12

Перенастройка выполняется системным администратором, изменяется конфигурационный файл варниш backends.vcl

# Настройка пути к статике в приложениях

Во всех бэкенд-приложениях, в ближайших релизах запланировать изменение параметра COMMON\_STATIC\_URL в конфигурации:

vi config.py;

COMMON\_STATIC\_URL = '//www.aeroflot.ru/media/common/' #уточнить наличие закрывающего слеша по результатам релиза ЛК 20.03.2016

# Обновление учетных записей в базе для сотрудников Аэрофлота после завершения параллельной эксплуатации

На afl-cms12 запускаем скрипт по смене паролей для учетных записей сотрудников Аэрофорта (таблица auth\_user). Скрипт меняет пароли и рассылает смс.

На основе Приложения 4 системный администратор готовит afl.csv файл в формате:  
username,password,pager  
foo,xxyyzz,9650000000  
  
Для генерации пароля использовать команду:

pwgen 10 1

Запускаем скрипт по смене паролей и рассылке смс на afl-cms12:

su – prod\_cms  
~/scripts/deploy/change\_psw\_with\_send\_sms.sh /home/prod\_cms/afl.csv

# Приложение 1. Календарный план с ответственными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название этапа** | **Дата начала** | **Дата окончания** | **Ответственный** |
|  | Предварительная сборка стенда RC2 | 21:00 15.03.2016 | 09:00 16.03.2016 | Степанов |
|  | Экспресс-тестирование стенда RC2 (проверка по последним исправленным замечаниям) | 09:00 16.03.2016 | 10:00 16.03.2016 | Макарова |
|  | Создание виртуальных машин, настройка ОС и системного ПО | 16.03.2016 | 20:00 17.03.2016 | Кечин |
|  | Фиксация состояния контента CMS Drupal, согласование cс Аэрофлотом запрета внесения правок в CMS Drupal |  | 20:00 17.03.2016 | Тюлькин |
|  | Подготовка дампа данных Drupal | 20:00 17.03.2016 | 20:20 17.03.2016 | Кечин (Костылев) |
|  | Финальная сборка стенда RC2 | 20:20 17.03.2016 | 09:00 18.03.2016 | Степанов |
|  | Экспресс-тестирование стенда RC2 (проверка основных функций CMS: вход в раздел администрирование, публикация страниц, новостей, хит-тарифов) | 09:00 18.03.2016 | 10:00 18.03.2016 | Макарова |
|  | Финальный перенос данных Django CMS с rc2 на промышленный стенд | 10:00 18.03.2016 | 13:00 18.03.2016 | Кечин |
|  | Перенос баннеров из CMS Drupal в СТБ | 09:00 18.03.2016 | 15:00 18.03.2016 | Мавлютов |
|  | Тестирование переноса баннеров в СТБ (тестирование только СТБ) | 13:00 18.03.2016 | 15:00 18.03.2016 | Макарова (Ребров) |
|  | Тестирование Django CMS, основных бизнес-процессов Сайта АФЛ | 15:00 18.03.2016 | 21:00 20.03.2016 | Макарова |
|  | Исправление замечаний по результатам тестирования | 15:00 18.03.2016 | 21:00 20.03.2016 | Степанов, Мавлютов, Кечин |
|  | Принятие решения о переводе Django CMS в промышленную эксплуатацию | 09:00 21.03.2016 | 12:00 21.03.2016 | Тюлькин, Новгородов |
|  | Отправка реквизитов доступа в раздел администрирования Django CMS сотрудникам Аэрофлота | 13:00 21.03.2016 | 14:00 21.03.2016 | Кечин |
|  | Переключение трафика с Drupal на Django CMS | 22:00 21.03.2016 | 22:30 21.03.2016 | Кечин (Костылев) |
|  | Модификация страниц ЭТП и калькуляторов в Drupal | 22:30 21.03.2016 | 23:30 21.03.2016 | Устимов |
|  | Тестирование переключения трафика | 22:30 21.03.2016 | 23:30 21.03.2016 | Кечин, Макарова |
|  | Мониторинг работы Django CMS | 22:00 21.03.2016 | 12:00 04.04.2016 | Кечин, Костылев |

Примечание: при возникновении необходимости внести срочные правки в контенте CMS Drupal во время проведения релиза - такие работы согласуются Заказчиком с А. Тюлькиным для того, чтобы сотрудники РИ выполнили эти работы в Django CMS.

# Приложение 2. Список сотрудников РИ, которым необходимо предоставить доступ в раздел администрирования Django CMS.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **ФИО** | **username** | **Номер телефона** |
| 1 | Новгородов А. А. | anovgorodov | +79254797270 |
| 2 | Мавлютов А. Ю. | amavlyutov | +79213244855 |
| 3 | Тюлькин А. | atyulkin | +79602380549 |
| 4 | Степанов В. В. | vstepanov | +79213062321 |
| 5 | Макарова Е. П. | emakarova | +79216330449 |
| 6 | Решетов И. | ireshetov | +79112978047 |
| 7 | Серазеева Ю. | yserazeeva | +79817998673 |
| 8 | Кечин В. | vkechin | +79659860186 |
| 9 | Костылев М. | mkostylev | +79643808725 |
| 10 | Кудряшов А. | akudryashov | +79046076982 |
| 11 | Якунин В. | vyakunin | +79255261115 |
| 12 | Фомин А. | afomin | +79218550400 |
| 13 | Филин В. | vfilin | +79095916211 |
| 14 | Ребров А. | arebrov | +79627250136 |
| 15 | Габдрахманова А. | agabdrakhmanova | +79817398280 |
| 16 | Робкова С. | srobkova | +79522776249 |
| 17 | Гурджинян Д. | dgurjinyan | +79268760480 |
| 18 | Устимов П. | pustimov | +79643350385 |
| 19 | Комаров А. | akomarov | +79119874484 |
| 20 | Фролова Л. | lfrolova | +79213184290 |

Всем пользователям из таблицы назначается роль «суперпользователь» (is\_superuser = true).

# Приложение 3. Список сотрудников Аэрофлота, которым необходимо предоставить доступ в раздел администрирования Django CMS после запуска параллельной эксплуатации.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **ФИО** | **Департамент** | **username** | **Номер телефона** |
| 1 | Землянко Е. | ООП | ezemlyanko | +79269251024 |
| 2 | Мулендеева О. | ДАБ | omulendeeva | +79858204244 |
| 3 | Березин Н.В. | ДОС | nberezin |  |

Всем пользователям из таблицы назначается роль «суперпользователь» (is\_superuser = true).

# Приложение 4. Список сотрудников Аэрофлота, которым необходимо предоставить доступ в раздел администрирования Django CMS после завершения параллельной эксплуатации.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **ФИО** | **Департамент** | **username** | **Номер телефона** |
| 1 | Степанова А.А. | ООП | astepanova |  |
| 2 | Москвина Н.А. | ООП | nmoskvina | +79067775263 |
| 3 | Уварова Л.В. | ООП | luvarova | +79031938343 |
| 4 | Костина М.А. | ООП | mkostina | +79165597516 |
| 5 | Линчик М.И. | ООП | mlinchik | +79262733345 |
| 6 | Чирков А.А. | ООП | aachirkov | +79264373056 |
| 7 | Тарасова Е. | ООП | etarasova | +79160748636 |
| 8 | Лысикова Е.Ю. | ДАБ | elysikova | +79165122764 |
| 9 | Шевченко К.А. | ДАБ | kshevchenko | +79637559225 |
| 10 | Беляева В.И. | ДАБ | vbelyaeva | +79055881516 |
| 11 | Модина А.М. | ДОС | amodina |  |
| 12 | Абанина В.Г. | ДОС | vabanina |  |
| 13 | Лукашин А. Ф. | ДОС | alukashin |  |
| 14 | Шабалина Т.С. | ДОС | tshabalina |  |
| 15 | Алещенко Е.А. | ДОС | eyurkovskaya |  |
| 16 | Савич Е.А. | ДОС | easavich |  |
| 17 | Перевалов И. В. | ДУПР | iperevalov | +79652586345 |
| 18 | Корчашкин Н. А. | ДУПР | nkorchashkin | +79161477526 |

Всем пользователям из таблицы назначается роль «суперпользователь» (is\_superuser = true).