QXYZ.RU руководство по инсталляции
(журнал инсталляции)

EustroSoft.org

Евстропов А.В. (Eustrop) 2023

Оглавление

Процесс инсталляции основного каскада серверов

Описание процесса инсталляции основного каскада виртуальных серверов. Выполняется в процессе инсталляции. Все виртуальные сервера основного каскада находятся на одном физическом сервере, система виртуализации proxmox, имя кластера — castor.

Этого каскада серверов достаточно для функционирования системы.

Второй (резервный) каскад серверов находится на втором физическом сервере (pollux), на нем находится постоянно функционирующие виртуальные сервера, на которые можно перевести нагрузку, если выходит из строя любой сервер из основного каскада серверов или весь кластер castor. Переключение нагрузки требует ручного вмешательства.

Состояние до начала работ

Дорога в 1000 ли начинается с первого шага,
а книга — с первого слова

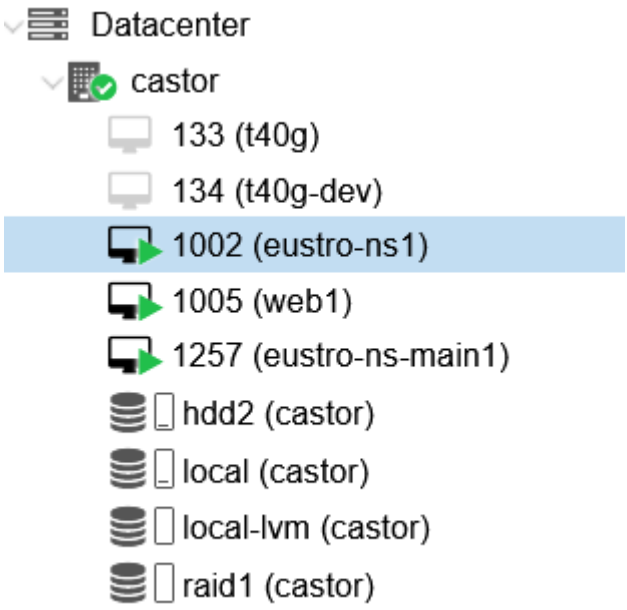


Рисунок 1: Состояние кластера castor до начала работ

Castor (100.76.200.101) — система виртуализации, установленная на физический сервер

t40g – шаблон виртуальной машины для продуктовых серверов

t40g-dev – шаблон виртуальной машины для серверов разработчиков

eustro-ns-main1 (100.76.208.7) – основной внутренний DNS сервер, источник всех DNS зон, на нем происходит получение SSL сертификатов (see /root/SSL/00readme)

eustro-ns1 (62.76.209.2) – первый публичный DNS-сервер, получает зоны с eustro-ns-main1

web1 (62.76.209.5, 100.76.209.5) — web1.qxyz.ru – первая (основная) публичная точка входа в систему. Порождена клонированием из t40g

План работ

1. настроить систему на web1 в составе
    1. nginx, принимающий запросы по https, для имен qxyz.ru, (aliases: web1.qxyz.ru, web2.qxyz.ru), qr.qxyz.ru, (aliases: q1.qxyz.ru, q2.qxyz.ru) и переадресующий их серверу приложений tomcat
    2. настроить сервер приложений tomcat, чтобы он принимал запросы для перечисленных выше имен
    3. установить систему qxyz.ru из репозитория и собрать ее
    4. установить проект ConcepTIS из репозитория
    5. переключить tomcat на собранные приложения qxyz.ru, qr.qxyz.ru
    6. инициализировать кластер postgresql, создать пользователей
    7. создать и проинициализировать базу данных qxyzdb на основе проекта ConcepTIS

8. настроить приложения на работу с БД qxyzdb
9. прописать в /etc/hosts имена w1,w2,tc1,tc2,pg1,pg2, и внутренние ip адреса им соответствующие, чтобы можно было быстро их вспомнить на каждом из клонированных серверов (всего таких серверов будет минимум 6).

2. клонировать web1 в tc1 и настроить tc1
   1. отключить nginx на tc1
   2. отключить tomcat and postgresql на web1
   3. переключить приложения на работу с БД находящейся на tc1
   4. переключить работу nginx на web1 на переадресацию запросов tomcat на tc1
3. клонировать tc1 в pg1 и настроить pg1
   1. отключить tomcat на pg1
   2. отключить postgresql на tc1
   3. настроить postgresql на pg1 так, чтобы снаружи можно было войти в БД qxyzdb только пользователями не имеющими прав суперпользователя/владельца БД, с серверов tc1 и tc2.
   4. Выделить дисковое пространство для БД (200 GB для начала?) и перенести на него директорию /var/db/postgresql/
   5. Переключить приложения, находящиеся на на tc1 на взаимодействие с БД на pg1

В дальнейшем: Сервера web2, tc2 и pg2 будут порождаться клонированием и перенастройкой серверов настроенных выше (через резервное копирование и восстановление на втором кластере proxmox (pollux)). Далее надо перейти к настройке репликации pg1 → pg2. Это предмет отдельного документирования, в отдельном разделе.

Также: порождение тестового каскада серверов test-web1, test-tc1, test-tc2, test-pg1, test-pg2 будет порождаться клонированием и перенастройкой серверов web2, tc2 и pg2 на втором кластере proxmox (pollux). Возможно, test-pg2 будет перенесен на castor. Это предмет отдельного документирования, в отдельном разделе.

Кроме того: существуют каскады серверов для разработчиков

Настройка web1.qxyz.ru

Сервер клонирован,

net0 → vmbr0 vlan1208

net1 → vmbr1 vlan1256

далее, в одно пользовательском режиме /etc/rc.conf

```
ifconfig_vtnet0="inet 100.76.209.5 netmask 255.255.254.0"
ifconfig_vtnet1="inet 62.76.209.5/28"
#defaultrouter="100.76.208.1"
defaultrouter="62.76.209.1"
```

также удаляем ssh ключи сервера, чтобы при перезагрузке их сгенерировали заново

```
# rm /etc/ssh/ssh_host_*
```

Перезагружаемся и заходим по ssh, запускаем screen и становимся root (use su(1) for this)

Запускаем tomcat & nginx

добавляем tomcat9_enable="YES" в /etc/rc.conf и запускаем tomcat

```
# /usr/local/etc/rc.d/tomcat9 start
```

проверяем его работу на порту http://web1.qxyz.ru:8080/ (заходим браузером)

добавляем nginx_enable="YES" в /etc/rc.conf и запускаем nginx

```
# /usr/local/etc/rc.d/nginx start
```

проверяем его работу на порту http://web1.qxyz.ru:80/ (заходим браузером)

Устанавливаем актуальные сертификаты для nginx

Настраиваем https в nginx, для этого актуальные сертификаты копируем с сервера eustro-ns-main1(100.76.208.7), находясь на нем (сертификаты уже были сгенерированы ранее):

```
root@eustro-ns-main1:~/SSL # tar -czf qxyz.ru.2023-10-31.tgz 2023-10-
31/qxyz.ru/
root@eustro-ns-main1:~/SSL # scp qxyz.ru.2023-10-31.tgz
yourname@web1.qxyz.ru:./
```

Возвращаемся на web1 и устанавливаем скопированные сертификаты

```
root@web1:/usr/local/etc/nginx # cd /usr/local/etc/nginx/
root@web1:/usr/local/etc/nginx # mkdir cert
root@web1:/usr/local/etc/nginx # chmod 700 cert
root@web1:/usr/local/etc/nginx # cd cert
root@web1:/usr/local/etc/nginx/cert # mkdir live
root@web1:/usr/local/etc/nginx/cert # tar -pxzf ~yourname/qxyz.ru.2023-
10-31.tgz
root@web1:/usr/local/etc/nginx/cert # cd live/
root@web1:/usr/local/etc/nginx/cert/live # ln -s ../2023-10-
31/qxyz.ru/ ./
root@web1:/usr/local/etc/nginx/cert/live # ll
total 0
lrwxr-xr-x  1 root  wheel  22  7 нояб. 20:24 qxyz.ru@ -> ../2023-10-
31/qxyz.ru/
```

Сертификаты будут находится в директории /usr/local/etc/nginx/cert/live/qxyz.ru/ когда придет время их обновлять будем размещать их в директорию, чье имя есть дата генерации, а символическую ссылку переключать директорию qxyz.ru/ в ней.

Внимание! После установки нового сертификата надо перезагрузить nginx

Настройка nginx на работу по ssl и переадресацию запросов tomcat

Конфигурируем nginx на работу по ssl и переадресацию запросов по обычному http на https

Добавляем в конфигурационный файл /usr/local/etc/nginx/nginx.conf

```
server {
 listen 62.76.209.5:443 ssl;
 server_name qxyz.ru *.qxyz.ru;
 client_max_body_size    3m;
 ssl_certificate_key /usr/local/etc/nginx/cert/live/qxyz.ru/privkey.pem;
 ssl_certificate     /usr/local/etc/nginx/cert/live/qxyz.ru/cert.pem;
      location / {
          proxy_pass       http://127.0.0.1:8080;
          proxy_set_header Host $http_host;
```

```
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Host $http_host;
            proxy_set_header X-Forwarded-Proto $scheme;
            #
            proxy_connect_timeout 120;
            proxy_send_timeout 120;
            proxy_read_timeout 180;
            access_log      /var/log/nginx/qxyz.ru.log;
        }
}
```

перезапускаем nginx

```
# /usr/local/etc/rc.d/nginx restart
```

Проверяем работоспособность, заходим на https://web1.qxyz.ru/

(теперь здесь должно отобразится то-же, что ранее мы видели на http://web1.qxyz.ru:8080/)

Настройка nginx на переадресацию всех запросов по http на https:

Добавляем в конфигурационный файл /usr/local/etc/nginx/nginx.conf

```
    server {
        listen 62.76.209.5:80;
        server_name qxyz.ru *.qxyz.ru;
        return  301     https://$host$request_uri;
    }
```

перезапускаем nginx

```
# /usr/local/etc/rc.d/nginx restart
```

Проверяем работоспособность, заходим на http://web1.qxyz.ru/

(теперь здесь нас должно перенаправить на https://web1.qxyz.ru и отобразится то-же, что ранее мы видели на http://web1.qxyz.ru:8080/)

nginx настроен, переходим к извлечению и сборке проектов.

Сборка и установка qxyz.ru

Пускай все файлы приложений (исходные коды и собранные приложения) принадлежат пользователю `operqxyz` и `operqxyz`. Работы по сборке так-же будем производить под этим пользователем.

Создадим группу `operqxyz`. Для этого добавим в файл `/etc/group` строку

```
operqxyz:*:1118:operqxyz
```

Внимание! В дальнейшем надо добавить это в шаблоны виртуальных машин t40g и t40g-dev на обоих кластерах castor & pollux.

```
root@web1:/s # cd /s
root@web1:/s # mkdir www
root@web1:/s # mkdir proj
root@web1:/s # chown -R operqxyz:operqxyz www/
root@web1:/s # chown -R operqxyz:operqxyz proj/
root@web1:/s # su -l operqxyz
```

Извлекаем репозитории:

```
operqxyz@web1:/s/www $ cd /s/www/
operqxyz@web1:/s/www $ git clone git@bitbucket.org:eustrop/qxyz.ru.git
operqxyz@web1:/s/www $ git clone git@repo.qxyz.ru:qxyz/qxyz.ru.git
operqxyz@web1:/s/www $ cd /s/proj/
operqxyz@web1:/s/proj$ git clone git@repo.qxyz.ru:eustrop/ConcepTIS.git
operqxyz@web1:/s/proj$ git clone git@bitbucket.org:eustrop/conceptis.git
operqxyz@web1:/s/proj$ mv conceptis ConcepTIS
```

NOTE: для пользователя operqxyz настроен файл /home/operqxyz/.gitconfig чтобы можно было вносить изменения в проект и коммитить их в репозиторий. Так-же публичный ssh ключ пользователя operqxyz помещен в список разрешенных для пользователя eustrop на bitbucket. Все это сделано из предположения, что доступ к продуктовым серверам имеют и учетным записям администраторов на них имеют сотрудники с максимальной степенью доверенности.

Переходим к сборке:

```
operqxyz@web1:/s/www $ cd /s/www/qxyz.ru
operqxyz@web1:/s/www/qxyz.ru $ make all
operqxyz@web1:/s/www/qxyz.ru $ mv work prod.installed
```

теперь в директории `/s/www/qxyz.ru/prod.installed` находятся скомпилированные и установленные приложения `webapps` для `qxyz.ru` и `webapps.qr` для `qr.qxyz.ru`, а также `webapps.not-configured` для нераспознанных сайтов

В каждом из них, кроме `webapps.not-configured`, следует настроить файл `ROOT/WEB-INF/web.xml`, установив пути к БД, надстройки пользователей БД и т. п.

Настройка tomcat для имен qxyz.ru и qr.qxyz.ru

Настроим директории для хранения журналов доступа к сайтам (под пользователем root)

```
root@web1:/s # cd www/qxyz.ru/
root@web1:/s/www/qxyz.ru # mkdir -p prod.log/qxyz.ru
root@web1:/s/www/qxyz.ru # mkdir -p prod.log/qr.qxyz.ru
root@web1:/s/www/qxyz.ru # mkdir -p prod.log/not-configured
root@web1:/s/www/qxyz.ru # chown -R www prod.log/
root@web1:/s/www/qxyz.ru # chmod 750 prod.log/
```

Отредактируем файл /usr/local/apache-tomcat-9.0/conf/server.xml

добавим коннекторы для IP адресов, на которых будет слушать сервер, вместо того чтобы он слушал на всех адресах:

```
> <!-- default connector disabled for qxyz.ru
73a75,99
> -->
>     <Connector port="8080" protocol="HTTP/1.1"
>                connectionTimeout="20000"
>                redirectPort="8443"
>                maxParameterCount="1000"
>                address="127.0.0.1"
>                />
>     <Connector port="8080" protocol="HTTP/1.1"
>                connectionTimeout="20000"
```

```
>                redirectPort="8443"
>                maxParameterCount="1000"
>                address="100.76.209.5"
>                />
>     <Connector port="8080" protocol="HTTP/1.1"
>                connectionTimeout="20000"
>                redirectPort="8443"
>                maxParameterCount="1000"
>                address="100.76.208.80"
>                />
>     <Connector port="8080" protocol="HTTP/1.1"
>                connectionTimeout="20000"
>                redirectPort="8443"
>                maxParameterCount="1000"
>                address="100.76.208.180"
>                />
```

Устанавливаем defaultHost="not-configured"

141c167

```
<     <Engine name="Catalina" defaultHost="localhost">
---
>     <Engine name="Catalina" defaultHost="not-configured">
```

Удаляем Host name="localhost"

160,174c186,192

```
<
<       <Host name="localhost"  appBase="webapps"
<             unpackWARs="true" autoDeploy="true">
<
<         <!-- SingleSignOn valve, share authentication between web
applications
<              Documentation at: /docs/config/valve.html -->
<         <!--
<         <Valve
className="org.apache.catalina.authenticator.SingleSignOn" />
<         -->
<
<         <!-- Access log processes all example.
<              Documentation at: /docs/config/valve.html
<              Note: The pattern used is equivalent to using
pattern="common" -->
<         <Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
<                prefix="localhost_access_log" suffix=".txt"
```

Добавляем конфигурации для трех Host : name="not-configured", name="qxyz.ru, name="qr.qxyz.ru"

```
>       <Host name="not-configured"
appBase="/s/www/qxyz.ru/prod.installed/webapps.not-configured"
```

```
unpackWARs="false" autoDeploy="true">
>         <Valve className="org.apache.catalina.valves.RemoteIpValve"
>             remoteIpHeader="x-forwarded-for"
>             proxiesHeader="x-forwarded-by"
>             protocolHeader="x-forwarded-proto" />
>         <Valve className="org.apache.catalina.valves.AccessLogValve"
directory="/s/www/qxyz.ru/prod.log/not-installed"
>                prefix="not-installed.access_log" suffix=".txt"
176d193
<
177a195,217
>       <Host name="qxyz.ru"
appBase="/s/www/qxyz.ru/prod.installed/webapps" unpackWARs="false"
autoDeploy="true">
>         <Valve className="org.apache.catalina.valves.RemoteIpValve"
>             remoteIpHeader="x-forwarded-for"
>             proxiesHeader="x-forwarded-by"
>             protocolHeader="x-forwarded-proto" />
>         <Valve className="org.apache.catalina.valves.AccessLogValve"
directory="/s/www/qxyz.ru/prod.log/qxyz.ru"
>                prefix="qxyz.ru_access_log" suffix=".txt"
>                pattern="%h %l %u %t &quot;%r&quot; %s %b" />
>          <Alias>web1.qxyz.ru</Alias>
>          <Alias>web2.qxyz.ru</Alias>
>       </Host>
>       <Host name="qr.qxyz.ru"
appBase="/s/www/qxyz.ru/prod.installed/webapps.qr" unpackWARs="false"
autoDeploy="true">
>         <Valve className="org.apache.catalina.valves.RemoteIpValve"
>             remoteIpHeader="x-forwarded-for"
>             proxiesHeader="x-forwarded-by"
>             protocolHeader="x-forwarded-proto" />
>         <Valve className="org.apache.catalina.valves.AccessLogValve"
directory="/s/www/qxyz.ru/prod.log/qr.qxyz.ru"
>                prefix="qr.qxyz.ru_access_log" suffix=".txt"
>                pattern="%h %l %u %t &quot;%r&quot; %s %b" />
>          <Alias>q1.qxyz.ru</Alias>
>          <Alias>q2.qxyz.ru</Alias>
>       </Host>
```

Перезапускаем tomcat:

root@web1:/s/www/qxyz.ru # /usr/local/etc/rc.d/tomcat9 restart

проверяем https://web1.qxyz.ru, https://q1.qxyz.ru http://site1.qxyz.ru

а также http://web1.qxyz.ru:8080/ (должен быть недоступен)

Можно переходить к настройке БД postgresql

Настройка сервера для кластера postgresql

Перемонтируем отдельную файловую систему в директорию /var/db/postgres/

```
/dev/da0p13    1,9G     16M    1,8G     1%    /var/c1

# umount /var/c1
```

редактируем /etc/fstab

```
#/dev/da0p13    /var/c1         ufs     rw      2       2
/dev/da0p12     /var/db         ufs     rw      2       2
/dev/da0p13     /var/db/postgres    ufs     rw      2       2
```

монтируем /var/db/postgres/

```
# mount /var/db/postgres/
Передаем владение смонтированной файловой системой
# chown postgres:postgres /var/db/postgres
```

добавим в /etc/login.conf описание класса для пользователя postgres, следуя рекомендациям из

pkg info -D postgresql13-server

Однако, выбираем русский язык, поскольку наша инсталляция системы ориентирована на русскоязычных пользователей.

```
#
# postgres class from fudo (eustrop 2023-05-45)
# with :setenv=LC_COLLATE=C:\ since 2023-11-09
#
postgres:\
        :lang=ru_RU.UTF-8:\
        :setenv=LC_COLLATE=C:\
        :tc=default:
```

Внимание! Не забываем пересобрать базу /etc/login.conf.db

```
# cap_mkdb /etc/login.conf
```

Назначаем созданный класс пользователю postgres через vipw

postgres:*:770:770:**postgres**:0:0:PostgreSQL Daemon:/var/db/postgres:/bin/sh

Инициализация кластера postgresql

```
root@web1:/s/www/qxyz.ru # su -l postgres
$ initdb --locale ru_RU.UTF-8 -D /var/db/postgres/data13
Файлы, относящиеся к этой СУБД, будут принадлежать пользователю
"postgres".
От его имени также будет запускаться процесс сервера.

Кластер баз данных будет инициализирован с локалью "ru_RU.UTF-8".
Кодировка БД по умолчанию, выбранная в соответствии с настройками:
"UTF8".
Выбрана конфигурация текстового поиска по умолчанию "russian".

Контроль целостности страниц данных отключён.

создание каталога /var/db/postgres/data13... ок
создание подкаталогов... ок
выбирается реализация динамической разделяемой памяти... posix
выбирается значение max_connections по умолчанию... 100
выбирается значение shared_buffers по умолчанию... 128MB
выбирается часовой пояс по умолчанию... W-SU
```

```
создание конфигурационных файлов... ок
выполняется подготовительный скрипт... ок
выполняется заключительная инициализация... ок
сохранение данных на диске... ок

initdb: предупреждение: включение метода аутентификации "trust" для
локальных подключений
Другой метод можно выбрать, отредактировав pg_hba.conf или используя
ключи -A,
--auth-local или --auth-host при следующем выполнении initdb.

Готово. Теперь вы можете запустить сервер баз данных:

    pg_ctl -D /var/db/postgres/data13 -l файл_журнала start
```

Выше приведено реальное сообщение об инициализации, с которым можно сверится

Запуск кластера postgresql

Добавляем в /etc/rc.conf

```
postgresql_enable=YES
postgresql_class="postgres"
```

запускаем кластер

```
root@web1:/s/www/qxyz.ru # /usr/local/etc/rc.d/postgresql start
2023-11-09 18:38:50.010 MSK [29992] СООБЩЕНИЕ:  завершение вывода в
stderr
2023-11-09 18:38:50.010 MSK [29992] ПОДСКАЗКА:  В дальнейшем протокол
будет выводиться в "syslog".
```

Внимание! Надо будет вернуться к настройке доступа в файлах `/var/db/postgres/data13/pg_hba.conf` и `/var/db/postgres/data13/postgresql.conf`

сделаем это после создания пользователей, БД, и ее инициализации

Создание пользователей и БД

```
operqxyz@web1:/s/www/qxyz.ru $ cd /s/proj/ConcepTIS/
operqxyz@web1:/s/proj/ConcepTIS $ cd src/sql/PGSQL/
operqxyz@web1:/s/proj/ConcepTIS $ make all
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ more
initdb/1_create_users.sql
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql postgres postgres
psql (13.12)
Введите "help", чтобы получить справку.
postgres=# \i initdb/1_create_users.sql
postgres=# \q
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ more
initdb/2_create_db.sql
Но базу данных мы будем создавать с другим именем - qxyzdb
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql postgres postgres
psql (13.12)
Введите "help", чтобы получить справку.
postgres=# CREATE DATABASE qxyzdb OWNER tisc ENCODING 'UTF8';
```

```
CREATE DATABASE
postgres=# \q
```

База данных создана! Теперь необходимо установить пароли всем созданным пользователям

Установка паролей пользователям БД:

```
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql postgres postgres
psql (13.12)
Введите "help", чтобы получить справку.

postgres=# \password  (press TAB to see users)
pg_execute_server_program  qtisadmin                tis_users
pg_monitor                 qtisguest                tisc
pg_read_all_settings       qtisnobody               tiscpool1
pg_read_all_stats          qtisoperator             tiscpool2
pg_read_server_files       qtisreplicator           tiscuser1
pg_signal_backend          qtisuser1                tiscuser2
pg_stat_scan_tables        qtisuser2                tiscuser3
pg_write_server_files      qtisuser3                tiscuser4
postgres                   qtiswww
postgres=# \password qtisadmin
Введите новый пароль для пользователя "qtisadmin":
Повторите его:
postgres=# \password qtisguest
postgres=# \password qtisnobody
postgres=# \password qtisoperator
postgres=# \password qtisreplicator
postgres=# \password qtisuser1
postgres=# \password qtisuser2
postgres=# \password qtisuser3
postgres=# \password qtiswww
postgres=# \password tisc
postgres=# \password tiscpool1
postgres=# \password tiscpool2
postgres=# \password tiscuser1
postgres=# \password tiscuser2
postgres=# \password tiscuser3
postgres=# \password tiscuser4
postgres=# \q
```

все пароли установлены

Инициализируем БД qxyzdb

```
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql qxyzdb postgres
psql (13.12)
Введите "help", чтобы получить справку.

qxyzdb=# \i initdb/3_config_db.sql
DROP SCHEMA
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql qxyzdb tisc
psql (13.12)
Введите "help", чтобы получить справку.
```

```
qxyzdb=> \i initdb/
1_create_users.sql  3_config_db.sql     drop_all.sql
2_create_db.sql     4_init_db.sql
qxyzdb=> \i initdb/4_init_db.sql
psql:initdb/4_init_db.sql:10: ОШИБКА:  база данных "conceptisdb" не
существует
psql:initdb/4_init_db.sql:11: ОШИБКА:  база данных "conceptisdb" не
существует
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
CREATE SCHEMA
GRANT
psql:initdb/4_init_db.sql:52: ОШИБКА:  расширение "plpgsql" уже
существует
qxyzdb=>
operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql qxyzdb tisc
psql (13.12)
Введите "help", чтобы получить справку.

qxyzdb=> REVOKE ALL ON DATABASE qxyzdb FROM PUBLIC;
REVOKE
qxyzdb=> GRANT CONNECT ON DATABASE qxyzdb TO tis_users;
GRANT

operqxyz@web1:/s/proj/ConcepTIS/src/sql/PGSQL $ psql qxyzdb tisc
psql (13.12)
Введите "help", чтобы получить справку.

qxyzdb=> \i step_1_dic.sql
qxyzdb=> \i step_2_SAM.sql
qxyzdb=> \i step_3_TIS.sql
qxyzdb=> \i step_4_TISC.sql
qxyzdb=> \i step_5_FS.sql
qxyzdb=> \i step_6_QR.sql
qxyzdb=> \i step_7_MSG.sql
qxyzdb=> select sam.inital_configuration_once();
     inital_configuration_once
---------------------------------
 Ok, local installation configured
(1 строка)
qxyzdb=> select sam.inital_local_configuration_once();
                              inital_local_configuration_once
-----------------------------------------------------------------------
```

```
--------------------
 NOT IMPLEMENMTED, write your configuration procedure
SAM.inital_local_configuration_once()
(1 строка)
```

Выполним остановку кластера postgresql и резервное копирование всех файлов кластера, чтобы можно было откатиться на это состояние:

```
root@web1:/s/www/qxyz.ru # cd /var/db/postgres/
root@web1:/var/db/postgres # /usr/local/etc/rc.d/postgresql stop
root@web1:/var/db/postgres # tar -czf web1_data13_20231116.tgz data13/
root@web1:/var/db/postgres # chmod 600 web1_data13_20231116.tgz
```

Самое время написать скрипт инициализации для qxyzdb

Однако, тут было принято решение перенести все репозитории на собственный сервер.

Инсталляция repo.qxyz.ru и перенос репозиториев.

1. Выделим ip 62.76.209.7
2. Виртуальный сервер разместим на кластере castor
3. ID виртуальной машины 1007
4. клонируем и размещаем виртуальную машину на кластере castor
5. загружаемся в одно-пользовательском режиме, настраиваем сеть, имя (в /etc/rc.conf) и удаляем ключи sshd (rm /etc/sshd_host*)
6. перезагружаемся
7.

```
# vipw
добавляем пользователя
git:*:1099:1099::0:0:git repositories
owner:/repo/git:/usr/local/bin/git-shell

root@repo: # mkdir /repo
root@repo: # umount /var/c1/
Настраиваем /etc/fstab
#/dev/da0p13    /var/c1         ufs     rw      2       2
/dev/da0p13     /repo           ufs     rw      2       2
root@repo: # mount /repo/
root@repo: # mkdir /repo/git
root@repo: # chown git:developers /repo/git

здесь была выполнена настройка директорий внутри /repo/git и перенос
репозиториев с bitbucket.org. Настройка ssh ключей и т. п. Смотри
содержимое /repo/git

root@repo: # mkdir /s/git_backup
Настраиваем резервное копирование (см /s/git_backup/bin и /etc/crontab)
# git repositories backup
1       6    *  *  *   root    /s/git_backup/bin/repo_backup_daily.sh
1       12   *  *  6   root    /s/git_backup/bin/repo_backup_weekly.sh
1       11   1  *  *   root    /s/git_backup/bin/repo_backup_mounthly.sh
```

```
root@repo:~ # cat /s/git_backup/bin/repo_backup_daily.sh
#!/bin/sh
FILENAME=repo.qxyz.ru.daily.tgz
DATE_MARK=`date "+%w"` #weekday
tar -Pczf /s/git_backup/${FILENAME} /repo/git/
scp /s/git_backup/${FILENAME}
git_backup@mnemosyne.qxyz.ru:/s/git_backup/${FILENAME}.${DATE_MARK}
```

Теперь на сервере mnemosyne.qxyz.ru будет накапливаться резервные копии за последние 7 дней. (а также за каждый месяц года)

Некоторые настройки остались за пределами этого отчета

Настройка приложений на работу с БД

Сделаем резервные копии оригинальных файлов перед их изменением

```
root@web1:/var/db/postgres/data13 # cp pg_hba.conf pg_hba.conf.orig
root@web1:/var/db/postgres/data13 # cp postgresql.conf
postgresql.conf.orig
```

внесем изменения в эти файлы

также внесем специальные имена в /etc/hosts

```
127.0.0.1       qxyz-pg-server
127.0.0.1       qr-qxyz-pg-server
```

в дальнейшем, именно через /etc/hosts будем направлять приложения в нужную БД

```
root@web1:/var/db/postgres/data13 # /usr/local/etc/rc.d/postgresql
restart

root@web1:/var/db/postgres/data13 # cd /usr/local/apache-tomcat-9.0/lib/
root@web1:/usr/local/apache-tomcat-9.0/lib # ln -s
/usr/local/share/java/classes/postgresql.jar ./
root@web1:/usr/local/apache-tomcat-9.0/lib # /usr/local/etc/rc.d/tomcat9
restart
```

В общем, все работает, осталось «доработать напильником» приложения и произвести настройку прав доступа и личных кабинетов для пользователей.

Напоследок посмотрим разницу в конфигурации кластера postgresql

```
root@web1:/var/db/postgres/data13 # diff postgresql.conf.orig
postgresql.conf
59a60
> listen_addresses = 'localhost,qxyz-pg-server,qr-qxyz-pg-server'
root@web1:/var/db/postgres/data13 # diff pg_hba.conf.orig pg_hba.conf
90c90,91
< host    all             all             127.0.0.1/32            trust
---
> host    all             all             127.0.0.1/32            md5
> host    all             all             100.76.0.1/16            md5
92c93
< host    all             all             ::1/128                 trust
---
> #host    all             all             ::1/128                 trust
```

Настройка файла /etc/hosts перед клонированием web1 → tc1 → pg1 → pg2

Вместо DNS, чтобы всегда было под рукой в /etc/hosts на всех клонированных машинах

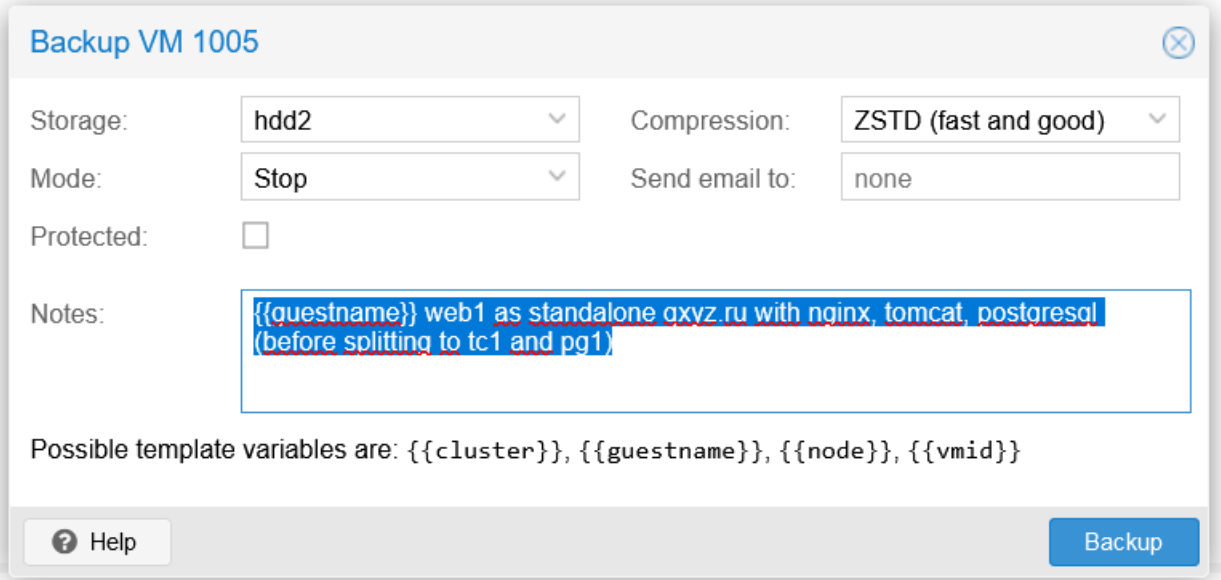
```
# qxyz inner prod servers
## main cascade
100.76.209.5    w1
100.76.208.80   tc1
100.76.210.84   pg1
## reserve cascade
100.76.209.19   w2
100.76.208.180  tc2
100.76.210.184  pg2
#qxyz inner test servers
## main test cascade
100.76.225.51   w1-test
100.76.224.80   tc1-test
100.76.226.84   pg1-test
## reserve test cascade
100.76.225.52   w2-test
100.76.224.180  tc2-test
100.76.226.184  pg2-test
```

Делаем резервную копию средствами proxmox

```
Перезагружаем сервер web1.qxyz.ru и проверяем, что все работает
root@web1:/var/db/postgres/data13 # uptime
19:55  up 12 days, 22:44, 1 user, load averages: 0,19 0,20 0,23
root@web1:/var/db/postgres/data13 # shutdown -r now
```

Вроде все работает. Выключаем виртуальную машину и делаем ее резервную копию

описание:
{{guestname}} web1 as standalone qxyz.ru with nginx, tomcat, postgresql (before splitting to tc1 and pg1)



Клонируем основной каскад серверов web1 → tc1 → pg1

Клонируем web1 → tc1, перенастраиваем оба

Выполняем клонирование web1 в tc1 (ID 1180, net1:vlan1210, RAM 4GB)

Размещаем его на hdd2, формат диска - raw

Сервер клонирован,

net0 → vmbr0 vlan1208

net1 → vmbr1 vlan1210

далее, в одно пользовательском режиме /etc/rc.conf

```
hostname="tc1.qxyz.ru"
keymap="ru.kbd"
ifconfig_vtnet0="inet 100.76.208.80/23"
ifconfig_vtnet1="inet 100.76.210.80/24"
defaultrouter="100.76.208.1"
sshd_enable="YES"
ntpdate_enable="YES"
ntpd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
#
tomcat9_enable="YES"
#nginx_enable="YES"

# postgresql
#postgresql_enable=YES
#postgresql_class="postgres"

переключаем путь к postgresql в /etc/hosts

# qxyz postgreses
100.76.210.84   qxyz-pg-server
100.76.210.84   qr-qxyz-pg-server
```

также удаляем ssh ключи сервера, чтобы при перезагрузке их сгенерировали заново

```
# rm /etc/ssh/ssh_host_*
```

Перезагружаемся и заходим по ssh, запускаем screen и становимся root (use su(1) for this)

…

Запускаем web1 и настраиваем его

Отключаем запуск tomcat и postgresql в /etc/rc.conf

```
#tomcat9_enable="YES"
nginx_enable="YES"
# postgresql
#postgresql_enable=YES
```

```
#postgresql_class="postgres"
```

Отключаем записи-указатели на postgresql в etc/rc.conf

```
# qxyz postgreses
#127.0.0.1      qxyz-pg-server
#127.0.0.1      qr-qxyz-pg-server
```

переключаем адрес server tomcat at /usr/local/etc/nginx/nginx.conf

```
    server {
        listen 62.76.209.5:443 ssl;
        server_name qxyz.ru *.qxyz.ru;
        ssl_certificate_key
/usr/local/etc/nginx/cert/live/qxyz.ru/privkey.pem;
        ssl_certificate
/usr/local/etc/nginx/cert/live/qxyz.ru/cert.pem;
        location / {
               #proxy_pass      http://127.0.0.1:8080;
               proxy_pass      http://tc1:8080;
               proxy_set_header Host $http_host;
...
```

```
Перезагружаем web1 и проверяем работоспособность
(вообще теперь можно удалить /s/www/qxyz.ru and /s/proj/ConcepTIS,
видимо они никогда не будут использоваться здесь)
```

Клонируем tc1 → pg1 и настраиваем его

Выполняем клонирование tc1 в pg1 (ID 1184, net0:vlan1211, RAM 4GB)

Размещаем его на raid1, формат диска - raw

Сервер клонирован,

net0 → vmbr0 vlan1211

net1 → vmbr1 vlan1210

далее, в одно пользовательском режиме /etc/rc.conf

```
hostname="pg1.qxyz.ru"
keymap="ru.kbd"
ifconfig_vtnet0="inet 100.76.211.84/23"
ifconfig_vtnet1="inet 100.76.210.84/24"
defaultrouter="100.76.210.1"
sshd_enable="YES"
ntpdate_enable="YES"
ntpd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
#
#tomcat9_enable="YES"
#nginx_enable="YES"
```

```
# postgresql
postgresql_enable=YES
postgresql_class="postgres"
```

переключаем настраиваем доступ к postgresql

```
root@pg1:/home/eustrop # diff
/var/db/postgres/data13/pg_hba.conf.orig
/var/db/postgres/data13/pg_hba.conf
90c90,92
< host    all             all             127.0.0.1/32            trust
---
> host    all             all             127.0.0.1/32            md5
> host    all             all             100.76.210.0/24            md5
> host    all             all             100.76.211.0/24            md5
92c94
< host    all             all             ::1/128                 trust
---
> #host    all             all             ::1/128                 trust
```

настраиваем postgresql слушать на **pg1,pg2,pg1-test,pg2-test (добавлено после настройки pg2. В резервной копии этого нет)**

```
root@pg2:/home/eustrop # diff
/var/db/postgres/data13/postgresql.conf.orig
/var/db/postgres/data13/postgresql.conf
59a60
> listen_addresses = 'localhost,pg1,pg2,pg1-test,pg2-test'
```

также удаляем ssh ключи сервера, чтобы при перезагрузке их сгенерировали заново

```
# rm /etc/ssh/ssh_host_*
```

Перезагружаемся и заходим по ssh, запускаем screen и становимся root (use su(1) for this)

…

Теперь надо будет добавить большой диск и перенести на него данные postgresql (directory /var/db/postgres/) но мы отложим это до клонирования pg2 и pg1-test pg2-test, чтобы решать эту задачу на каждом из них отдельно, когда потребуется

Делаем резервные копии web1,tc1,pg1 средствами proxmox

Останавливаем все эти сервера и делаем резервную копию каждого

комментарий к резервной копии

{{guestname}} main cascade configured 2023-11-16 22:40

Создаем второй каскад серверов pg2,tc2,web2 на кластере pollux

Копируем резервные копии серверов web1,tc1,pg1 с castor на pollux

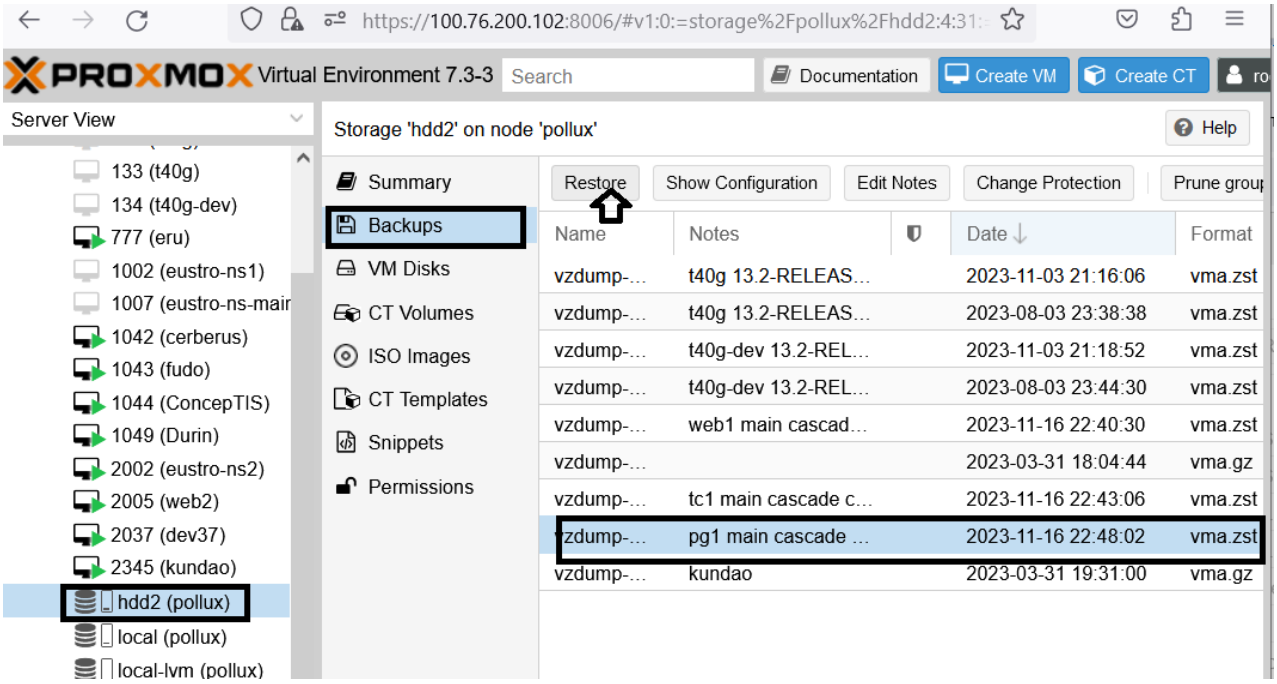
Копируем резервные копии серверов web1,tc1,pg1 с castor на pollux

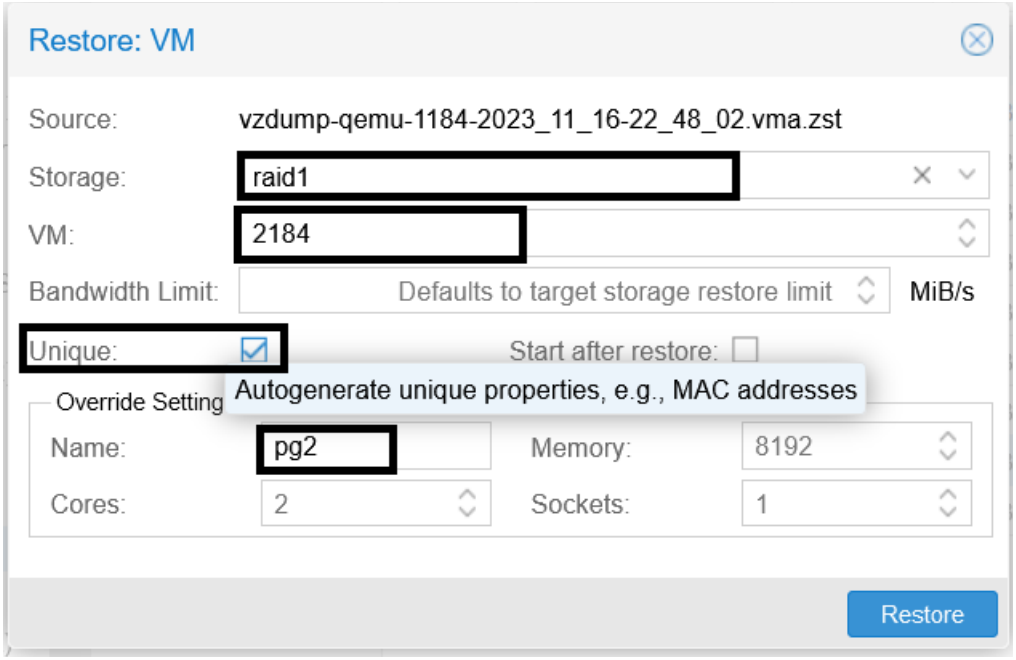
Внимание! Действуем медленно и предельно аккуратно! Проверяем каждую каждую команду 3 (три) раза, чтобы ничего не испортить. Это касается всех действий выполняемых в

консоли на системах виртуализации из-под root.

```
root@castor:/disk/hdd2/dump# scp vzdump-qemu-1005-2023_11_16-22_40_30.*
pollux:/disk/hdd2/dump/
root@pollux's password:
vzdump-qemu-1005-2023_11_16-22_40_30.log
100% 3532     5.8MB/s   00:00
vzdump-qemu-1005-2023_11_16-22_40_30.vma.zst
100% 3003MB 111.9MB/s   00:26
vzdump-qemu-1005-2023_11_16-22_40_30.vma.zst.notes
100%   45   171.7KB/s   00:00
root@castor:/disk/hdd2/dump# scp vzdump-qemu-1180-2023_11_16-22_43_06.*
pollux:/disk/hdd2/dump/
root@pollux's password:
vzdump-qemu-1180-2023_11_16-22_43_06.log
100% 3424     5.9MB/s   00:00
vzdump-qemu-1180-2023_11_16-22_43_06.vma.zst
100% 3002MB 112.0MB/s   00:26
vzdump-qemu-1180-2023_11_16-22_43_06.vma.zst.notes
100%   44   217.3KB/s   00:00
root@castor:/disk/hdd2/dump# scp vzdump-qemu-1184-2023_11_16-22_48_02.*
pollux:/disk/hdd2/dump/
root@pollux's password:
vzdump-qemu-1184-2023_11_16-22_48_02.log
100% 3321     4.8MB/s   00:00
vzdump-qemu-1184-2023_11_16-22_48_02.vma.zst
100% 3005MB 112.0MB/s   00:26
vzdump-qemu-1184-2023_11_16-22_48_02.vma.zst.notes
100%   44   168.7KB/s   00:00
```

Создаем pg2 из резервной копии pg1 на pollux





Установим VMID, хранилище, имя машины и попросим сгенерировать новые уникальные MAK адреса. Ну, попробуем?

Вроде получилось. Проконтролируем уникальность MAC адресов (см. следующий раздел), запустим машину в одно пользовательском режиме и изменим конфигурацию

далее, в одно пользовательском режиме /etc/rc.conf

```
hostname="pg2.qxyz.ru"
keymap="ru.kbd"
ifconfig_vtnet0="inet 100.76.211.184/24"
ifconfig_vtnet1="inet 100.76.210.184/24"
defaultrouter="100.76.210.1"
sshd_enable="YES"
ntpdate_enable="YES"
ntpd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
#
#tomcat9_enable="YES"
#nginx_enable="YES"

# postgresql
postgresql_enable=YES
postgresql_class="postgres"
```

настраиваем postgresql слушать на **pg2**

```
root@pg2:/home/eustrop # diff
/var/db/postgres/data13/postgresql.conf.orig
/var/db/postgres/data13/postgresql.conf
59a60
> listen_addresses = 'localhost,pg1,pg2,pg1-test,pg2-test'
```

также удаляем ssh ключи сервера, чтобы при перезагрузке их сгенерировали заново

```
# rm /etc/ssh/ssh_host_*
```

Перезагружаемся и заходим по ssh, запускаем screen и становимся root (use su(1) for this)

…

Теперь надо будет добавить большой диск и перенести на него данные postgresql (directory /var/db/postgres/) но мы отложим это до клонирования pg2 и pg1-test pg2-test, чтобы решать эту задачу на каждом из них отдельно, когда потребуется

Контроль уникальности MAC адресов

Контроль в пределах одного кластера можно выполнить так

```
root@pollux:/disk/hdd2/dump# cat /etc/pve/nodes/pollux/qemu-
server/*.conf | grep '^net' | awk -F"[=,]" '{print $2}' | sort | uniq -c
```

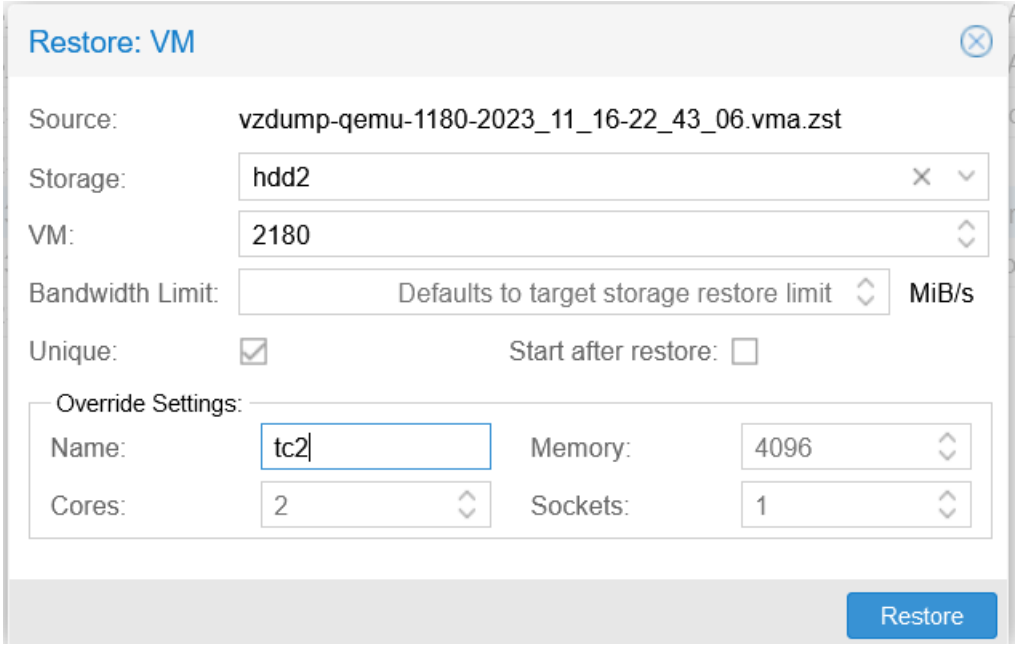
В пределах двух это можно выполнить так:

```
root@pollux:/disk/hdd2/dump# sh -c "ssh castor 'cat
/etc/pve/nodes/castor/qemu-server/*.conf'; ssh pollux 'cat
/etc/pve/nodes/pollux/qemu-server/*.conf'" |  grep '^net' | awk -F"[=,]"
'{print $2}' | sort | uniq -c  | awk '($1 > 1){print $0}'
```

Замечание: И здесь есть дубли, но не для машины pg2. Дубли касаются машин tg40, tg40-dev, eustro-ns1, полные копии которых присутствуют на обоих кластерах, но никогда не должны работать одновременно (в каждой по 2 сетевых интерфейса, на двух первых и 3 на eustro-ns1, итого 7 дублей по MAC)

Создаем tc2 из резервной копии tc1 на pollux

Действуем по аналогии с Создаем pg2 из резервной копии pg1 на pollux на стр 20



VMID 2180, name: tc2, storage: hdd2

далее, в одно пользовательском режиме /etc/rc.conf

```
hostname="tc2.qxyz.ru"
keymap="ru.kbd"
ifconfig_vtnet0="inet 100.76.208.180/23"
ifconfig_vtnet1="inet 100.76.210.180/24"
defaultrouter="100.76.208.1"
sshd_enable="YES"
ntpdate_enable="YES"
ntpd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
#
tomcat9_enable="YES"
#nginx_enable="YES"

# postgresql
#postgresql_enable=YES
#postgresql_class="postgres"

не переключаем путь к postgresql в /etc/hosts, оставляем его как есть.
Оба tomcat будут по-умолчанию работать с одной и той-же БД.

# qxyz postgreses
100.76.210.84   qxyz-pg-server
100.76.210.84   qr-qxyz-pg-server
```

также удаляем ssh ключи сервера, чтобы при перезагрузке их сгенерировали заново

```
# rm /etc/ssh/ssh_host_*
```

Перезагружаемся и заходим по ssh, запускаем screen и становимся root (use su(1) for this)

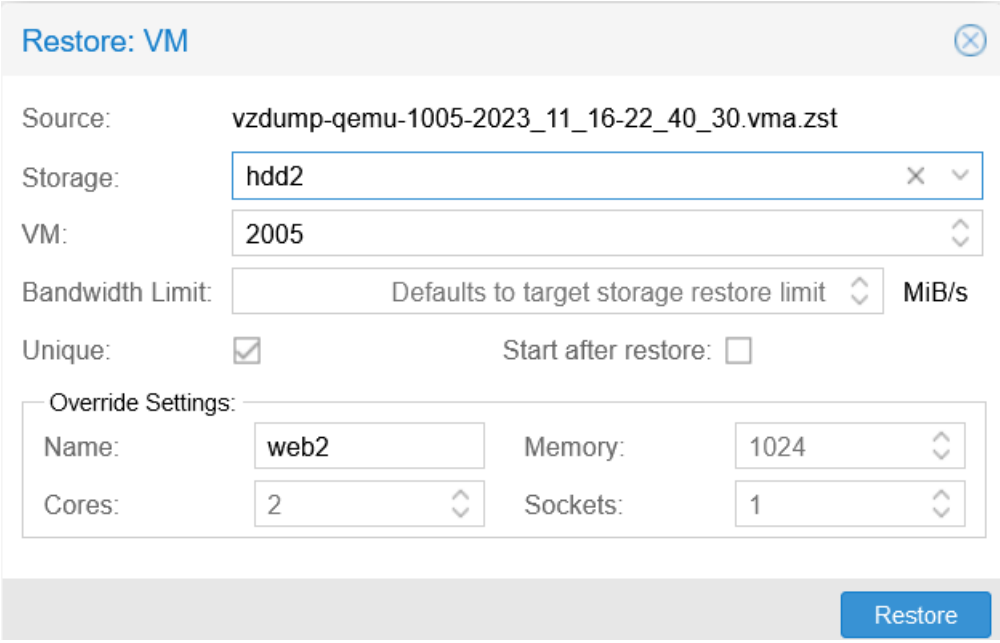
…

Делаем резервную копию старого web2 и удаляем его

У нас есть старая версия сервера web2 от 2023-06-14. Сделаем его резервную копию и удалим, чтобы создать заново, по аналогии с pg2,tc2

description: {{guestname}} old and broken version of web2 since 2023-06-14

Создаем web2 из резервной копии web1 на pollux



VMID 2005, name: web2, storage: hdd2

net1 → vlan1257

далее, в одно пользовательском режиме /etc/rc.conf

```
hostname="web2.qxyz.ru"
keymap="ru.kbd"
ifconfig_vtnet0="inet 100.76.209.19 netmask 255.255.254.0"
ifconfig_vtnet1="inet 62.76.209.19/29"
#defaultrouter="100.76.208.1"
defaultrouter="62.76.209.17"
sshd_enable="YES"
ntpdate_enable="YES"
ntpd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
#
#tomcat9_enable="YES"
nginx_enable="YES"

# postgresql
#postgresql_enable=YES
#postgresql_class="postgres"
```

Вносим изменения в /usr/local/etc/nginx/nginx.conf

```
root@web2:/home/eustrop # diff /usr/local/etc/nginx/nginx.conf-dist
/usr/local/etc/nginx/nginx.conf
121c121,145
<
---
>     server {
>         listen 62.76.209.19:80;
>       server_name qxyz.ru *.qxyz.ru;
>       return  301     https://$host$request_uri;
```

```
>     }
>     server {
>         listen 62.76.209.19:443 ssl;
>       server_name qxyz.ru *.qxyz.ru;
...
>       location / {
>             #proxy_pass      http://127.0.0.1:8080;
>             proxy_pass       http://tc2:8080;
...
```

Делаем резервные копии web2,tc2,pg2 средствами proxmox

Останавливаем все эти сервера и делаем резервную копию каждого

комментарий к резервной копии

{{guestname}} reserve cascade configured 2023-11-17 20:04

Создание тестового каскада серверов

Тестовый каскад серверов состоит из

Сервер	ID	Net0	Net1	gw
test1.qxyz.ru	3005	vlan1224 inet 100.76.225.51/23	vlan1258 inet 62.76.209.51/27	62.76.209.33
tc1-test	3180	vlan1224 inet 100.76.224.80/23	vlan1226 inet 100.76.226.80/24	100.76.224.1
pg1-test	3184	vlan1226 inet 100.76.226.84/24	vlan1227 inet 100.76.227.84/24	100.76.226.1
test2.qxyz.ru	4005	vlan1224 inet 100.76.225.52/23	vlan1258 inet 62.76.209.52/27	62.76.209.33
tc2-test	4180	vlan1224 inet 100.76.224.180/23	vlan1226 inet 100.76.226.180/24	100.76.224.1
pg2-test	4184	vlan1226 inet 100.76.226.184/24	vlan1227 inet 100.76.227.184/24	100.76.226.1

Виртуальные машины порождаем на кластере pollux, хранилище hdd2 для всех машин, клонирование web2 → test1 → test2; tc2 → tc1-test → tc2-test; pg2 → pg1-test → pg2-test

перенастраиваем nginx на test1 и test2, tomcat на tc1-test and tc2-test, postgresql на pg1-test and pg2-test

отличие от продуктовых каскадов: tc2-test будет смотреть в pg2-test, для отладочных целей

детального документирования процесса вести не будем, для экономии времени и бумаги

2023-11-17 20:50 начало работ

2023-11-17 21:40 – готов каскад test2 → tc2-test → pg2-test

останавливаем все, делаем резервные копии всех трех машин

description: {{guestname}} second cascade of test servers ready 2023-11-17 21:48

и клонируем test1,tc1-test,pg1-test

2023-11-17 21:59 завершено резервное копирование

2023-11-17 22:03 завершено клонирование трех серверов

2023-11-17 22:33 завершена настройка первого каскада test1 → tc1-test → pg1-test

останавливаем все, делаем резервные копии всех трех машин

description: {{guestname}} primary cascade of test servers ready 2023-11-17 22:35

2023-11-17 22:43 завершено резервное копирование

2023-11-17 22:47 все сервера работают

Настройка серверов для разработки (dev38,dev37,dev39)

Dev38.qxyz.ru 62.76.209.38, 100.76.225.38 – сервер для eustrop

Dev37.qxyz.ru 62.76.209.37, 100.76.225.37 – for yadzuka and lysrt

Dev39.qxyz.ru 62.76.209.39, 100.76.225.39 – for yadzuka

Также зарезервированы сервера dev36, dev35,dev34 (deb34)

Начинаем с dev38, далее из него клонируем dev37 и присоединяем к нему диск старого dev37-old монтируя все в /dev37-old. Dev39 предназначен для экспериментов с более новыми версиями пакетов и будет порожден клонированием из dev37, на него будет установлена java11 и все что еще нужно для экспериментов.

Основная база данных будет расположена на dev37 (?)

В основном при настройке следуем протоколу настройки web1

public vlan 1258 net 62.76.209.32/27 gw 62.76.209.33

inner vlan 1224 net 100.76.224.0/23 gw 100.76.224.1

DNS zones for devXX.qxyz.ru delegated to NS service at the same server (IP 62.76.209.XX)

dev38.qxyz.ru

Настройка DNS

настраиваем bind в качестве мастера для зоны dev38.qxyz.ru

(см /usr/local/etc/namedb/)

diff named.conf named.conf.sample

Порождение сертификата

Для порождения сертификата

```
root@dev38:~/SSL # pkg install py39-certbot-dns-rfc2136
root@dev38:~/SSL # certbot certonly --manual --preferred-challenges dns
--server https://acme-v02.api.letsencrypt.org/directory  -d
'*.dev38.qxyz.ru' -d dev38.qxyz.ru
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): noc@eustrosoft.org

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You
must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
(Y)es/(N)o: y
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully
issued, to
share your email address with the Electronic Frontier Foundation, a
founding
partner of the Let's Encrypt project and the non-profit organization
that
develops Certbot? We'd like to send you email about our work encrypting
the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: n
Account registered.
```

Далее, следуя инструкции, дважды обновляем зону dev38.qxyz.ru и каждый раз перезагружаем ее

```
root@dev38:/usr/local/etc/namedb # vi master/dev38.qxyz.ru
root@dev38:/usr/local/etc/namedb # /usr/local/etc/rc.d/named reload


Successfully received certificate.
Certificate is saved at:
/usr/local/etc/letsencrypt/live/dev38.qxyz.ru/fullchain.pem
Key is saved at:
/usr/local/etc/letsencrypt/live/dev38.qxyz.ru/privkey.pem
This certificate expires on 2024-04-17.
```

Полагая, для простоты будем использовать сертификаты оттуда, где они размещены по-умолчанию. Это отличается от производственной системы

Настраиваем nginx

За основу берем конфиг от web1

```
root@dev38:/usr/local/etc/nginx # diff nginx.conf nginx.conf-dist
121,146c121
<     server {
<         listen 62.76.209.38:80;
<       server_name dev38.qxyz.ru *.dev38.qxyz.ru;
<       return  301     https://$host$request_uri;
<     }
<     server {
<         listen 62.76.209.38:443 ssl;
<       server_name dev38.qxyz.ru *.dev38.qxyz.ru;
<       client_max_body_size    3m;
<       ssl_certificate_key
/usr/local/etc/letsencrypt/live/dev38.qxyz.ru/privkey.pem;
<       ssl_certificate
/usr/local/etc/letsencrypt/live/dev38.qxyz.ru/cert.pem;
<       location / {
<             proxy_pass       http://127.0.0.1:8080;
<             #proxy_pass      http://tc1:8080;
```

```
<               proxy_set_header Host $http_host;
<               proxy_set_header X-Real-IP $remote_addr;
<               proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
<               proxy_set_header X-Forwarded-Host $http_host;
<               proxy_set_header X-Forwarded-Proto $scheme;
<               #
<               proxy_connect_timeout 120;
<               proxy_send_timeout 120;
<               proxy_read_timeout 180;
<               access_log      /var/log/nginx/qxyz.ru.log;
<       }
<     }
```

Сборка и установка qxyz.ru

В соответствии с протоколом для web1 за исключением, mv work prod-installed. Будем запускать все приложения из work

Настраиваем tomcat

SIC! На серверах разработки включаем автоматическую перезагрузку приложений при изменении библиотек в WEB-INF/lib/. По аналогии со старым dev37

```
eustrop@dev37:/usr/local/apache-tomcat-9.0/conf % diff context.xml
context.xml.sample
19,20c19
< <!-- reloadable="true" 2023-11-20 eustrop -->
< <Context reloadable="true" >
---
> <Context>
```

В остальном, за основу берем конфиг от tc1 и дорабатываем его до:

root@dev38:/usr/local/apache-tomcat-9.0/conf # diff server.xml server.xml.sample

Настраиваем postgresql

БД восстанавливаем из резервной копии от 2023-08-03

производим ее до-настройку сверяясь с журналом инсталляции web1 и работами над ошибками проделанными на текущую дату 2024-01-19

Наполнение БД

База данных заполнена описаниями пользователе, областей и т. п. Так чтобы разработчики могли продолжить работу.

Выполняем резервное копирование

Замена сертификатов SSL

Это — регулярная процедура, которую надо выполнять каждые 2 месяца, в последних числах марта,мая,июля,сентября,ноября, января или в первых числах следующего месяца (апрель,июнь, август, октябрь, декабрь,февраль). Срок действия сертификата — 3 месяца, один месяц мы оставляем себе для резерва времени, если что-то пойдет не так (например нам перестанут выдавать бесплатные сертификаты Let's Encrypt)

Для всех задействованных доменов мы получаем wild-card сертификаты (*.qxyz.ru, *.qxyz.su и т. д.). Данный процесс пока не поддается полной автоматизации.

Процедура выполняется на внутреннем DNS сервере eustro-ns-main1 100.76.208.7 (хранилище оригиналов зон dns)

Для машин разработки (dev37.qxyz.ru, dev38.qxyz.ru) операция выполняется на них самих по упрощенной процедуре.

1. Для замены сертификатов заходим на машину 100.76.208.7 под пользователем operqxyz
2. запускаем или восстанавливаем сессию screen
3. создаем два виртуальных терминала(term0 and term1) , в каждом становимся root
4. в term0 заходим в директорию /root/SSL/letsencrypt/ и следуем логике системного администрирования запрограммированной в ней, в поддиректориях и Makefile в них
5. в term1 - cd /usr/local/etc/namedb/
6. создаем еще 4 виртуальных терминала
7. term2 – ssh eustro-ns1.qxyz.ru, становимся root and cd /usr/local/etc/namedb/
8. term3 – ssh eustro-ns3.qxyz.ru, становимся root and cd /usr/local/etc/namedb/
9. term4 – ssh web1.qxyz.ru, становимся root
10. term5 - ssh web1.qxyz.ru, становимся root
11. term6 – ssh -p 2222 kundao.eustrosoft.org, становимся root (пока не рассматриваем здесь, на эту машину пока нельзя зайти под operqxyz.ru)

Обновление сертификатов на примере qxyz.su

Term0 - make build

```
root@eustro-ns-main1:~/SSL/letsencrypt # cd qxyz.su/
root@eustro-ns-main1:~/SSL/letsencrypt/qxyz.su # make build
Step 1/3 - building cert for qxyz.su
certbot certonly --manual --preferred-challenges dns --server
https://acme-v02.api.letsencrypt.org/directory  -d '*.'qxyz.su -d
qxyz.su
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Renewing an existing certificate for *.qxyz.su and qxyz.su

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
Please deploy a DNS TXT record under the name:
```

```
_acme-challenge.qxyz.su.

with the following value:

GT2ofXNv0dRNVH3XTkucIZC5ufAUX5m7es9evLh6n60

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
Press Enter to Continue
```

term1 vi master/qxyz.su; /usr/local/etc/rc.d/named reload

```
root@eustro-ns-main1:/usr/local/etc/namedb # vi master/qxyz.su
;
; This is named.db for qxyz.su domain
;                       Eustrop
;                       2018/04/16 - started from eustrosoft.org

$TTL    3600

@       IN      SOA     ns3.qxyz.ru. root.ns3.qxyz.ru.  (
                                55      ; Serial
                                1200    ; Refresh
                                300     ; Retry
                                3600000 ; Expire
                                1200 )  ; Minimum
        IN      NS      ns1
        IN      NS      ns3
        IN      A       62.76.209.5
; MX
        IN      MX      5       mx02.nicmail.ru.
        IN      MX      10      mx01.nicmail.ru.
        IN      MX      20      mx03.nicmail.ru.
;_acme-challenge.qxyz.ru.       IN      TXT
-XMhDZ6LgFYMO7Z88AKCaZT1HDuyYdG4q5oq4l7mEnM
;_acme-challenge.qxyz.ru.       IN      TXT     zkXY-3-0J4ey_CS-
B1qcORd3AtQ4UCJKqT6SQ9awDAY
;
_acme-challenge.qxyz.su.        IN      TXT
GT2ofXNv0dRNVH3XTkucIZC5ufAUX5m7es9evLh6n60
;_acme-challenge.qxyz.su.       IN      TXT
GCe_7rd_acxFkiJU4ZqD8bvr5_sjfOwcJSL8n8resSY

root@eustro-ns-main1:/usr/local/etc/namedb # /usr/local/etc/rc.d/named
reload
root@eustro-ns-main1:/usr/local/etc/namedb # grep named
/var/log/messages
```

Проверяем, что все в порядке

Term2 rm slave/qxyz.su; /usr/local/etc/rc.d/named restart

```
root@eustro-ns1:/usr/local/etc/namedb # rm slave/qxyz.su
root@eustro-ns1:/usr/local/etc/namedb # /usr/local/etc/rc.d/named
restart
```

```
Stopping named.
Waiting for PIDS: 12621, 12621.
Starting named.
root@eustro-ns1:/usr/local/etc/namedb # grep named /var/log/messages
```

Проверяем, что все в порядке

Term3 rm slave/qxyz.su; /usr/local/etc/rc.d/named restart

Term0 pressing enter:

```
Please deploy a DNS TXT record under the name:

_acme-challenge.qxyz.su.

with the following value:

dv-IJGe7hgO4Y1dmt6tqCgZLLbIziupRvKTjFyfVlGs

(This must be set up in addition to the previous challenges; do not
remove,
replace, or undo the previous challenge tasks yet. Note that you might
be
asked to create multiple distinct TXT records with the same name. This
is
permitted by DNS standards.)

Before continuing, verify the TXT record has been deployed. Depending on
the DNS
provider, this may take some time, from a few seconds to multiple
minutes. You can
check if it has finished deploying with aid of online tools, such as the
Google
Admin Toolbox: https://toolbox.googleapps.com/apps/dig/#TXT/_acme-
challenge.qxyz.su.
Look for one or more bolded line(s) below the line ';ANSWER'. It should
show the
value(s) you've just added.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
Press Enter to Continue
```

term1 vi master/qxyz.su; /usr/local/etc/rc.d/named reload

term2 rm slave/qxyz.su; /usr/local/etc/rc.d/named restart

term3 rm slave/qxyz.su; /usr/local/etc/rc.d/named restart

Term0 pressing enter to continue

```
Successfully received certificate.
Certificate is saved at:
/usr/local/etc/letsencrypt/live/qxyz.su/fullchain.pem
```

```
Key is saved at:
/usr/local/etc/letsencrypt/live/qxyz.su/privkey.pem
This certificate expires on 2024-04-27.
These files will be updated when the certificate renews.

NEXT STEPS:
- This certificate will not be renewed automatically. Autorenewal of
--manual certificates requires the use of an authentication hook script
(--manual-auth-hook) but one was not provided. To renew this
certificate, repeat this same certbot command before the certificate's
expiry date.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -
```

term0 make pkg; make install

```
root@eustro-ns-main1:~/SSL/letsencrypt/qxyz.su # make pkg
root@eustro-ns-main1:~/SSL/letsencrypt/qxyz.su # make install
```

Установка сертификатов на примере qxyz.ru

Term0 make install_warning

```
ATTENTION! go to hosts above and finish installation process! with next
actions:
# tar -pxzf /home/operqxyz/certs/qxyz.ru.2024-01-28.cert.tgz -C
/usr/local/etc/nginx/cert/
# rm /usr/local/etc/nginx/cert/live/qxyz.ru
# ln -s /usr/local/etc/nginx/cert/2024-01-28/qxyz.ru
/usr/local/etc/nginx/cert/live/qxyz.ru
# /usr/local/etc/rc.d/nginx restart
```

Term4 (web1.qxyz.ru)

```
# tar -pxzf /home/operqxyz/certs/qxyz.ru.2024-01-28.cert.tgz -C
/usr/local/etc/nginx/cert/
# rm /usr/local/etc/nginx/cert/live/qxyz.ru
# ln -s /usr/local/etc/nginx/cert/2024-01-28/qxyz.ru
/usr/local/etc/nginx/cert/live/qxyz.ru
# /usr/local/etc/rc.d/nginx restart
```

term2 (web2.qxyz.ru)

```
# tar -pxzf /home/operqxyz/certs/qxyz.ru.2024-01-28.cert.tgz -C
/usr/local/etc/nginx/cert/
# rm /usr/local/etc/nginx/cert/live/qxyz.ru
# ln -s /usr/local/etc/nginx/cert/2024-01-28/qxyz.ru
/usr/local/etc/nginx/cert/live/qxyz.ru
# /usr/local/etc/rc.d/nginx restart
```

Работа над ошибками

2023-11-20 исправляем /etc/login.conf класс russian (and postgres). Устанавливаем lc_messages = 'C' в postgresql.conf

Проблема 1/2: испорчен класс russian

```
#
# Russian Users Accounts. Setup proper environment variables.
#
russian|Russian Users Accounts:\
        :charset=UTF-8:\
        :tc=default:
#
# postgres class from fudo (eustrop 2023-05-45)
# with :setenv=LC_COLLATE=C:\ since 2023-11-09
#
postgres:\
        :lang=ru_RU.UTF-8:\
        :setenv=LC_COLLATE=C:\
        :tc=default:
```

Должно быть

```
#
# Russian Users Accounts. Setup proper environment variables.
#
russian|Russian Users Accounts:\
        :charset=UTF-8:\
        :lang=ru_RU.UTF-8:\
        :tc=default:
#
# postgres class from fudo (eustrop 2023-05-45)
# with :setenv=LC_COLLATE=C:\ since 2023-11-09
#
postgres:\
        :charset=UTF-8:\
        :lang=ru_RU.UTF-8:\
        :setenv=LC_COLLATE=C:\
        :tc=default:
```

Не уверен по поводу класса postgres но сделаем так везде, для единообразия

Решение : внести изменение и выполнить cap_mkdb /etc/login.conf

Проблема 2/2: нечитаемые сообщения от postgresql в syslog (несовместимость с UTF-8)

Решение : внести изменение в /var/db/postgres/data13/postgresql.conf (lc_messages = 'C')

Host (prod)	time	notes
web1	18:21:00	
tc1	18:23:00	
pg1	18:24:00	lc_messages = 'C' (/var/db/postgres/data13/postgresql.conf)

web2	18:26:00	
tc2	18:27:00	
pg2	18:30:00	lc_messages = 'C' (/var/db/postgres/data13/postgresql.conf)
Host (test)	**time**	**notes**
test1	18:46:00	
tc1-test	18:49:00	
pg1-test	18:50:00	lc_messages = 'C' (/var/db/postgres/data13/postgresql.conf)
test2	18:52:00	
tc2-test	18:56:00	
pg2-test	18:58:00	lc_messages = 'C' (/var/db/postgres/data13/postgresql.conf)
host(misc)		
repo.qxyz.ru		Нет проблемы
mnemosyne		Нет проблемы
t40g		Нет проблемы
t40g-dev		Нет проблемы
eustro-ns-main1		Нет проблемы
eustro-ns1		Нет проблемы
eustro-ns2		Нет проблемы
Dev37.qxyz.ru	19:00:00	lc_messages = 'C' (/var/db/postgres/data13/postgresql.conf)

2023-12-27 nginx.conf – разрешаем тело запроса 3 мб

```
server {
    listen 62.76.209.5:443 ssl;
    server_name qxyz.ru *.qxyz.ru;
    client_max_body_size    3m;
```

Тело POST запроса при загрузке файла в БД чуть больше 2 мб

исправлено на dev37, web1 и web2

Раздел 1

Приложения

Приложение А

Литература

[1] TIS-SQL - Манифест разработчика (2008 г) http://eustrosoft.org/projects/TIS-SQL/TIS2007/TIS-SQL-Manifesto.pdf

[2] ...

[3] исходный код проекта ConcepTIS https://bitbucket.org/eustrop/conceptis

История версий документа

- ...
- v0.1 : 2023/08/10 – первая версия шаблона 3 стр, 33 абзацев, 273 слов, 1000 знака, 54 строк, итого /Eustrop/