



III. ArchLinux («Архитектура системы на уровне ОС»).

Версия 3.

В данном разделе представлена техническая информация об имеющихся физических и виртуальных серверах Проекта.

Наименование проекта и раздела данной технической документации, доступ к ней. Техническое наименование, дата, версия и автор документа:

1. Проект: **Abirix**
2. Техническое наименование настоящего документа: **Abirix-TechDoc-ArchLinux-3**
3. Дата последнего внесения изменений: **2018-05 -16**
4. Версия: рабочая (это последняя рабочая версия на дату последнего внесения изменений)
5. Автор: Василий Конаков, email:, phone: **+41.33**
6. Последняя версия настоящего документа постоянно доступна по адресу <http://abirix.online> (для доступа к документации потребуется авторизованный доступ).

Указанные здесь параметры являются рабочими! Возможен переезд на другой хостинг. До момента промышленной эксплуатации схема вряд ли поменяется (может быть только дополнена либо в крайнем случае поменяется полностью).

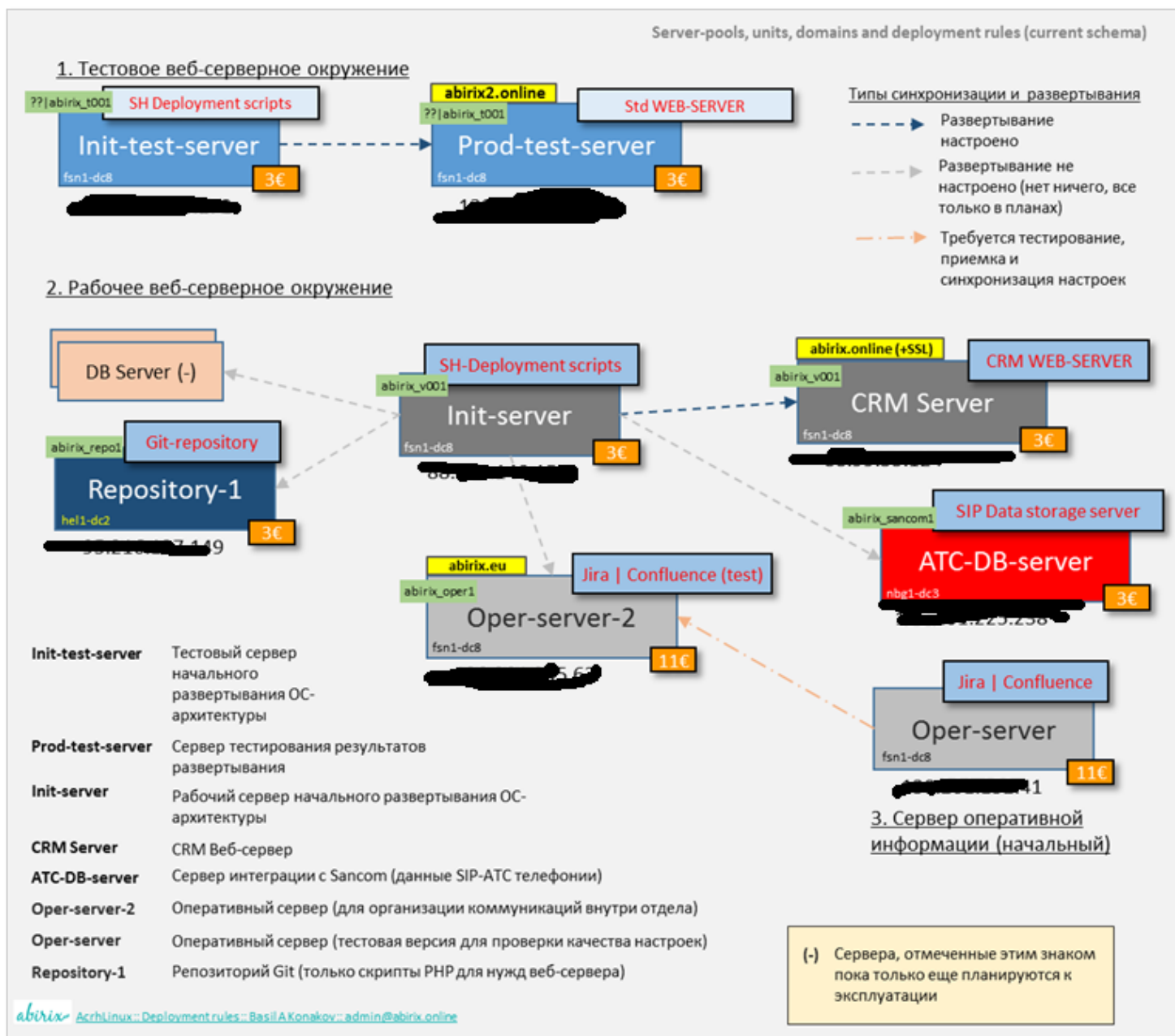
Содержание:

III. ArchLinux («Архитектура системы на уровне ОС»). Версия 3.	1
1. Юниты, пулы, домены и правила инициализации системы.....	2
2. Технические параметры эксплуатации юнитов системы.....	3
3. Инструкция по развертыванию ПО на юнит-сервере	4
4. Описание «чистого развертывания» на примере связки серверов abirixinit-abirixprod.....	7
5. Инструкция по развертыванию ПО на локальной машине разработчика	8
6. Пример создания скриптов развертывания на локальной машине разработчика	9
7. Создание и тестирование скриптов инициализации для новых версий пула.....	9
8. Транзакции развертывания рабочих серверов Проекта	10
9. Системные транзакции, необходимые для комплексного развития системы	11
10. Планы по развитию системы в плане интеграции	13
11. Конфигурация всех учетных записей системы с учетом локальной разработки	15

1. Юниты, пулы, домены и правила инициализации системы

В соответствии с имеющимися потребностями в серверных ресурсах текущая архитектура системы на уровне ОС представлена в виде набора из XXX-ми виртуальных серверов, арендуемых в ДЦ Hetzner. При этом общая стоимость затрат на аренду пока не превышает XXX евро в месяц, но должна будет существенно вырасти к моменту начала промышленной эксплуатации системы Проекта.

В рамках текущей ситуации задействовано 3 домена. Все сервера системы объединены в три пула, причем первые два из них (тестовое и рабочее окружение) имеют независимые друг от друга методы инициализации системы, а третий пул состоит из всего лишь одного сервера, на котором производится тестирование настроек, связанных с планами по эксплуатации Jira и Confluence:



2. Технические параметры эксплуатации юнитов системы

Параметры работы отдельных физических и виртуальных серверов системы на уровне ОС должны быть сопоставимы с параметрами работы других серверов Проекта. По сути, это памятка относительно того, чего можно ожидать в плане производительности каждого из юнитов, а также она позволяет определить возможности интеграции внутренних модулей системы друг с другом и с внешними модулями.

Server-pools, units, domains and deployment rules (current schema)

1. Тестовое веб-серверное окружение

abirix2.online

Типы синхронизации и развертывания

Server-pool unit parameters (current data)

Тестовое веб-серверное окружение

Init-test-server

acc: <apxi
Hardware
CPU1: Ir
Memor
Disk /de
Total ca
Network
eth0 LI
MAC
IP: :
IPv6:
Virtio
OS: Debia
Hostnam

Prod-test-server

acc: <ap
Hardwa
CPU1:
Memc
Disk /i
Total c
Networ
eth0 I
M/
IP:
IPv
Vir
OS: Deb
Hostnai

Сервер оперативной информации (начальный)

Oper-server

acc: <ap>
Hardwar
CPU1: l
(Skylake)
Memor
Disk /d
Total ca
Network
eth0 LI
MA
IP:
IPv6
Virt
OS: Cent
Hostnam

Сервер регистрации данных CC (SIP-ATC)

ATC-DB-server

acc: <apxi
Hardware
CPU1: Ir
(Skylake)
Memor
Disk /de
Total ca
Network
eth0 LI
MAC
IP: 1
IPv6:
Virtio
OS: Debia
Hostnam

Рабочее веб-серверное окружение

Init-server

acc: <apxi
Hardware
CPU1: In
Memory
Disk /de
Total cap
Network d
eth0 LIN
MAC:
IP: 8
IPv6:
Virtio
OS: Debian
Hostname

CRM Server

acc: <a
Hardw
CPU:
Memr
Disk,
Total
Netwo
eth0
N
IP
IP
V
OS: De
Hostni

Oper-server-2

acc: <архинов>
Hardware data:
CPU1: Inte
(Skylake, IBI
Memory:
Disk /dev/
Total capa
Network da
eth0 LINK
MAC: :
IP: 13
IPv6: 2
Virtio r
OS: Debian-
Hostname:

Repository-1

acc: <архинов>
Hardware
CPU1: Ir
(Skylake)
Memor
Disk /de
Total ca
Network
eth0 LI
MAC
IP: 9
IPv6
Virtio
OS: Debia
Hostnam: www.abirix.ru

3. Инструкция по развертыванию ПО на юнит-сервере

Ввиду сжатых сроков разработки полноценной инструкции по развертыванию системы на серверах рабочего и тестового пулов в настоящее время не подготовлено! Требуются усилия технического писателя. Тем не менее в сжатом виде памятка по развертыванию все же имеется.

Page 4 of 15

1) КЛЮЧЕВЫЕ ПАРАМЕТРЫ РАЗВЕРТЫВАНИЯ (параметров всего 3 штуки):

<version> *версия деплоя (другими словами это идентификатор сценария развертывания, который применим к тому или иному серверу или группе серверов, в независимости от того рабочая это машина или виртуальная машина на компьютере разработчика)*

Для пула рабочих серверов версия деплоя в настоящее время подготовлена только одна: **abirix_v001** («Рабочий деплой веб-сервера версии 1»). Данная версия деплоя является стабильной и достаточной для конфигурирования «Рабочего сервера начального развертывания ОС-архитектуры» (юнит **Init-server**), а также автоматизированного развертывания отдельных модулей CRM-системы, фактически реализованной на базе «CRM Веб-сервера» с помощью chroot-окружения (используется пока только один юнит **Prod-server**).

Для тестового пула серверов версия деплоя также подготовлена и протестирована пока только одна: **abirix_t001** («Тестовый деплой веб-сервера версии 1»). Она создана для ознакомления с методикой развертывания модулей CRM-системы devops-ами Проекта.

Версии **abirix_t001** и **abirix_v001** абсолютно идентичны (за исключением конфигов развертывания – то есть какой-либо пул серверов не имеет информации о другом пуле серверов, что имеет первоочередное значение с т.з. требований безопасности).

Далее необходим апгрейд данной версии деплоя (в новых версиях, но на базе существующей) в сторону:

- 1) Возможностей добавления новых баз данных Maria Db и их пользователей на лету (скрипт готов, надо только подключить);
- 2) Использования докер-контейнеров (на Proxu, репликация БД, интеграция с заббикс т.д.);
- 3) Адаптации скриптов под CentOS и возможно Фряху;
- 4) Добавления нового функционала (возможности работы с БД PostgreSQL, Jira, Confluence и BitBucket).

Применение: Фактически манипулирование версиями происходит через основной скрипт деплоя **_install.sh** (при его вызове в командной строке **bash**, а также в самом **_install.sh** – дублирование сделано не случайно, возможно потом надо убрать!!!). Версия деплоя также указывается в файлах-памятках **version.bklv** и **notes.bklv**, но в итоге это ни на что не влияет!

В ближайшее время планируются к проработке дополнительные версии:

- **abirix_sancom1** (Абирикс «Сервер для интеграции с Sancom версии 1») – для развертывания и конфигурирования ОС на сервере **ATC-DB-server**;
- **abirix_oper1** (Абирикс «Сервер получения данных SIP-ATC Sancom версии 1») – для развертывания и конфигурирования ОС на сервере **ATC-DB-server**;

<PRJ_SRCIP> IP-адрес init-сервера текущего пула серверов (unit-a, на котором лежат скрипты всех версий и конфиги unit-ов для данного пула, то есть «с которого» происходит развертывание)

IP-адрес init-сервера для каждого из пулов можно узнать из данной технической документации (см. общую схему).

<PRJ_CODE> Принимает одно из двух значений:
- **hosting_hetzner** – если запускаем на удаленном сервере в ДЦ Hetzner;
- **local_vagrant** - если запускаем локально на рабочей машине разработчика в виртуальной машине с помощью Vagrant (нужно иметь дополнительные скрипты!)

Без указания параметра **PRJ_CODE** скрипты до конца не отработают!!!

2.1) Описание файлов конфигурации и скриптов разворачивания версий abirix_t001 и abirix_v001:

2.1.1) ВСПОМОГАТЕЛЬНЫЕ СКРИПТЫ НА PHP

abirix_t001/php/<version>/	Данные скрипты необходимы для обработки файлов yml.
abirix_t001/php/<version>/initConfByIP.php	В последнее время эти скрипты фактически не менялись, но все равно каждый раз храним их в данной папке для каждой из версий поскольку возможны изменения в структуре yml (если такие изменения будут то мы просто оставим их в папке текущей версии деплоя и они не будут перезаписаны).
abirix_t001/php/<version>/nfByIP.php	
abirix_t001/php/<version>/spyc.php	

2.1.2) ОСНОВНЫЕ СКРИПТЫ РАЗВЕРТЫВАНИЯ BASH (BOURNE AGAIN SHELL) – Debian Only!!!

abirix_t001/_system/sh/<version>/	
abirix_t001/_system/sh/<version>/install.sh	главный скрипт разворачивания (с точки зрения разворачивания версии внутри определенного пула серверов здесь должны быть корректно указаны параметры bash-переменных <version> и <srcip>, иначе будут ошибки)
abirix_t001/_system/sh/<version>/breakout.sh	дополнительный скрипт (здесь не нужно что-либо менять, скрипт не содержит каких-либо переменных)
abirix_t001/_system/sh/<version>/params.sh	рабочий скрипт с внутренними переменными BASH (ничего там не меняем, иначе сервер потом не заведется!!!)
abirix_t001/_system/sh/<version>/functions.sh	функции для главного скрипта разворачивания
abirix_t001/_system/sh/<version>/notes.bklv	ФАЙЛ-ПАМЯТКА с командой инициализации разворачивания на хостинге - проверяем версию <version> и IP-адрес init-сервера <PRJ_SRCIP>!!!
abirix_t001/_system/sh/<version>/version.bklv	ФАЙЛ-ПАМЯТКА - здесь хранится параметр <version> (наличие файла и его содержимое ни на что не влияет)

2.1.3) ПАПКА С КОНФИГАМИ ПУЛА СЕРВЕРОВ ПРИ РАЗВЕРТЫВАНИИ ОПРЕДЕЛЕННОЙ ВЕРСИИ

abirix_t001/_system/sh/<version>/	
В тестовой версии (для примера) указаны самоподписанные сертификаты для условного IP-адреса «u47vc_std_v002», имеющего только один хост 127.0.1 и домены (abirix.lc и др.) IP-адрес «u47vc_std_v002» является условным и используется для разворачивания рабочего окружения программиста на локальном компьютере с определенным набором локальных доменов. Таким образом с помощью условного u47vc_std_v002 каждый разработчик получает именно то окружение, которое нужно именно ему!!!	

2.1.4) ОСНОВНЫЕ СКРИПТЫ РАЗВЕРТЫВАНИЯ BASH (BOURNE AGAIN SHELL) – Debian Only!!!

abirix_t001/_system/sh/<version>/	
abirix_t001/_system/sh/<version>/initConfByIP.yml	файл памятка с идентификаторами участников бизнес-процессов как отдельных бизнес-единиц (файл чисто технический (вроде бы ни на что не влияет))
abirix_t001/_system/sh/<version>/nfByIP.yml	файл с конфигурацией сервер-фтип_аккаунт-домен для каждого из IP-адресов + дополнительные настройки (ВАЖНО! IP-адреса могут выступать в качестве доменов. При использовании данной версии для разворачивания окружения на Init-сервере здесь должен быть указан <PRJ_SRCIP>)
abirix_t001/_system/sh/<version>/nfByIP.yml	файл с конфигурацией сервер-БД-БД_Юзер + прочие настройки подключения к БД для каждого из IP-адресов (пароли к БД могут генериться автоматически или задаваться заранее).

4. Описание «чистого развертывания» на примере связки серверов abirixinit-abirixprod

Имеются два виртуальных веб-сервера на хостинг Hetzner, а также удаленный доступ скриптам деплоя нужной версии (на другом сервере – условно пусть это будет IP = *.220 – некий архивный сервер, а версия – abirix_v001). Поэтому здесь и далее описаны действия по развертыванию фактически «голой» системы на базе ОС Debian с параметрами развертывания ПО как в Разделе 2. «Технические параметры эксплуатации юнитов системы».

1. Заходим в соответствующий аккаунт Hetzner;
2. Инициализируем режим Rescue для юнитов abirixinit и abirixprod;
3. Ребутим сервера юнитов abirixinit и abirixprod;
4. Заходим на сервер abirixinit по руту по SSH, запускаем команду installimage, вводим параметры инсталляции ОС в соответствии с конфигами машин: OS: Debian-94-stretch-64-minimal, Hostname: abirixinit, ребутим сервер после завершения процедуры установки параметров инсталляции ОС;
5. После переустановки системы заходим по FTP на сервер abirixinit и копируем из репозитория *.220 папку XXXXXXXXXX в корень сервера;
6. Проверяем ключевые параметры:
 - Файл /_ XXXXXXXXXX /_install.sh – ключевые параметры version (**abirix_v001**), srcip (IP-адрес сервера abirixinit);
 - Файл /_ XXXXXXXXXX /notes.bklv - PRJ_SRCIP (IP-адрес сервера abirixinit), PRJ_VERSION (**abirix_v001**), PRJ_CODE="hosting_hetzner" (поскольку развертывание происходит на серверах, размещенных в ДЦ Hetzner);
 - В папке /_ XXXXXXXXXX v/ssl в соответствии с каждым из IP-адресов (в том числе и условных локальных) должны быть размещены сертификаты SSL, например:
 - 1) Официально выпущенные сертификаты (подписанные одним из авторизованных центром сертификации), должны быть размещены в виде:
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/EXTERNAL/<domain>.cert
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/EXTERNAL/<domain>.key
 - 2) Самоподписанные сертификаты (самостоятельно подписанные какой-либо компанией), должны быть размещены в виде (указано для двух условных доменов <domain1> и <domain2>, подписанных <ABIRIX>):
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/LINKED /<ABIRIX>@<domain1>
 - /_ XXXXXXXXXX m/ssl/<IP-адрес>/LINKED/<ABIRIX>@<domain2>
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/PUBLIC/CA/<ABIRIX>.cert
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/SECRET/CA/<ABIRIX>.key
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/SECRET/CA/<ABIRIX>-<domain1>.srl
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/SECRET/CA/<ABIRIX>-<domain2>.srl
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/PUBLIC/sites/<domain1>.cert
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/PUBLIC/sites/<domain1>.key
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/PUBLIC/sites/<domain2>.cert
 - XXXXXXXXXX /ssl/<IP-адрес>/PUBLIC/sites/<domain2>.key
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/SECRET /sites/<domain1>.csr
 - /_ XXXXXXXXXX /ssl/<IP-адрес>/SECRET /sites/<domain2>.csr

**) Примечание: первые два файла – это пустые файлы-привязки доменов <domain1> и <domain1> с подписавшим для них сертификаты центром <ABIRIX>, содержимое которых не имеет значения!*

- Файл /_ XXXXXXXXXX /abirix_v001/confByIP.yml может использоваться для конфигурирования данных web-окружения и члут аккаунтов;
- Файл /_ XXXXXXXXXX /abirix_v001/MySQLbyIP.yml может содержать данные о разворачиваемых на серверах БД и их пользователях.

7. Заходим по SSH на сервер abirixinit и в командной строке запускаем команды, которые хранятся в файле «/_bklv/sh/abirix_v001/notes.bklv», например:

```
clear; clear; PRJ_SRCIP=<IP-адрес Init-сервера>; PRJ_VERSION=abirix_v001;  
PRJ_CODE="hosting_hetzner"; i=/_XXXXXXXXXX /${PRJ_VERSION}; mkdir -p ${i}; scp  
root@${PRJ_SRCIP}:${i}/_install.sh $i; bash $i/_install.sh ${PRJ_CODE} ${PRJ_VERSION} ${PRJ_SRCIP};
```

8. По мере работы скрипта указываем параметры работы сервера. ВАЖНО! Указанный при работе скрипта новый root-пароль автоматически станет новым паролем доступа этого пользователя! **На этом начальная настройка Init-сервера на уровне ОС будет полностью завершена!**
9. Заходим по SSH на сервер abirixprod и в командной строке запускаем точно такие же команды, которые указаны в п.7, вот пример (в обоих случаях скрипты должны быть абсолютно идентичны, везде должен быть указан IP-адрес Init-сервера):

```
clear; clear; PRJ_SRCIP=<IP-адрес Init-сервера>; PRJ_VERSION=abirix_v001;  
PRJ_CODE="hosting_hetzner"; i=/_XXXXXXXXXX PRJ_VERSION}; mkdir -p ${i}; scp  
root@${PRJ_SRCIP}:${i}/_install.sh $i; bash $i/_install.sh ${PRJ_CODE} ${PRJ_VERSION} ${PRJ_SRCIP};
```

10. По мере работы скрипта указываем параметры работы сервера. ВАЖНО! Указанный при работе скрипта новый root-пароль автоматически станет новым паролем доступа этого пользователя! **На этом начальная настройка Prod-сервера на уровне ОС будет полностью завершена!**

5. Инструкция по развертыванию ПО на локальной машине разработчика

Ввиду сжатых сроков разработки полноценной инструкции по развертыванию системы на локальной машине разработчика пока еще не подготовлено! Требуются усилия технического писателя. Тем не менее в сжатом виде памятка по развертыванию все же имеется.

1. С помощью гита клонируем с сервера, на котором это хранится, папку со скриптами развертывания рабочего окружения программиста вида ws_<box_type>_<this_prj>, например, ws_vbox_abirix_lc1. Последнее означает – «версия развертывания с помощью Virtualbox локально поколения 1»;
2. Инструкции по развертыванию хранятся в файлах **!notes/deploy_params.bklv** и **!notes/deploy_steps.bklv** (обычный текстовый файл). Последний файл с описаниями действий по развертыванию будет иметь примерно следующее содержание:

```
# 1. на винде  
bash /c/_bklv/data/ws_vbox_abirix_lc1/init.sh.bklv;
```

) Система организована так, что после прохождения первого этапа развертывания (успешного выполнения данного скрипта) подключаться к виртуальной машине можно будет без пароля с использованием беспарольного сертификата, имеющего имя файла и расположение в папке: **!notes/ssh/keys_used/vagrant20180307noPass.ppk*

```
# 2. в виртуалке под вагрантом  
sudo bash /vagrant/_system/_update.sh;
```

***) Система организована так, что после прохождения второго этапа развертывания (успешного выполнения данного скрипта внутри локальной машины) подключаться к БД MariaDB в виртуальной машине можно будет через SSH-тоннель с использованием беспарольного сертификата, имеющего имя файла и расположение: **!notes/ssh/keys_used/vagrant20180307noPass.ppk** (специальный формат для IDE DataGrid)*

6. Пример создания скриптов развертывания на локальной машине разработчика

Предполагается, что уже есть удаленный Init-сервер, который предварительно настроен с учетом локального развертывания. Далее нужно всего лишь «поднять» локально машину с ПО, идентичным серверному ПО на одном из нужных серверов пула с участием данного Init-сервера.

Памятка по развертыванию ПО на PC разработчика

1.1) ПАРАМЕТРЫ ВИРТУАЛИЗАЦИИ

(МЕНЯЮТСЯ РЕДКО, ИЗМЕНЕНИЯ => НОВАЯ ВЕРСИЯ - don't change!!!)

параметр:	значение:	
<<fix_deploy_localhost>>	localhost	- локально
<<fix_ssh_key>>	...	= ключ SSH
<<box_id>>	...	= версия ОС
<<box_version>>	...	= версия ОС
<<box_type>>	...	~ Virtualbox

1.2) ПАРАМЕТРЫ РАЗВЕРТЫВАНИЯ

(very careful - указаны в качестве примера)

параметр:	значение:	
<<ip_init>>	88.198....	= IP-адрес Init
<<ip_this>>	abirix...host1	= условный IP-адрес
<<port_80>>	80	~ проброс порта 80
<<port_443>>	...	~ проброс порта 443
<<version>>	abirix...	- текущая версия деплоя

1.3) ТЕХНИЧЕСКИЕ ПАРАМЕТРЫ

(поясняющие параметры - мало на что влияют)

параметр:	значение:
<<this_platform>>	Windows
<<this_prj>>	abirix...
<<this_hostname>>	abirix...
<<this_memory>>	4096
<<this_pass_root>>	...
<<this_pass_mysql>>	1234567mysql
<<this_pass_postgres>>	1234567postgres
<<this_git_user>>	V.Konakov
<<this_git_domain>>	abirix.com
<<date_created>>	16.05.2018
<<date_now>>	16.05.2018
<<person_created>>	Basil A Konakov
<<person_me>>	Basil A Konakov

2) ВСТАВКА ЗНАЧЕНИЙ ПАРАМЕТРОВ ВМЕСТО МАКРОСОВ :

2.1) В комментариях файлов:

init.sh.bklv
deploy\ws_script.make_deploy.sh.bklv
deploy\ws_script.install.sh.bklv
deploy\ws_script.vagrant_provision.sh.bklv
deploy\ws_script.common_params.sh.bklv

# @adopted	version:<<version>>
# @platform	<<platform>>
# @created	<<date_created>>
# @author	<<person_created>>
# @refactoredBy	<<who_am_i>>
# @updated	<<date_now>>

2.2) В комментариях файла init.sh.bklv:

```
# @localtion C:\... \<<box_type>> \<<this_prj>> ...  
## remove C:\... \<<this_prj>> ...  
## remove C:\... \<<box_type>> \<<this_prj>> \vagrant ...  
## C:\_bklv\... \box\<<this_hostname>> ...
```

2.3) В коде скрипта Vagrantfile:

```
config.vm.box = "<<box_id>>"  
config.vm.box_version = "<<box_version>>"  
config.vm.hostname = "<<this_hostname>>"  
config.vm.synced_folder "../sftp/<<this_prj>>" ...  
...  
v.memory = <<box_memory>>  
v.name = "<<this_hostname>>"  
...  
config.vm.network :forwarded_port, guest: 80, host: <<http_port_80>>  
config.vm.network :forwarded_port, guest: 443, host: <<http_port_443>>
```

2.4) В коде скрипта deploy\ws_script.common_params.sh.bklv:

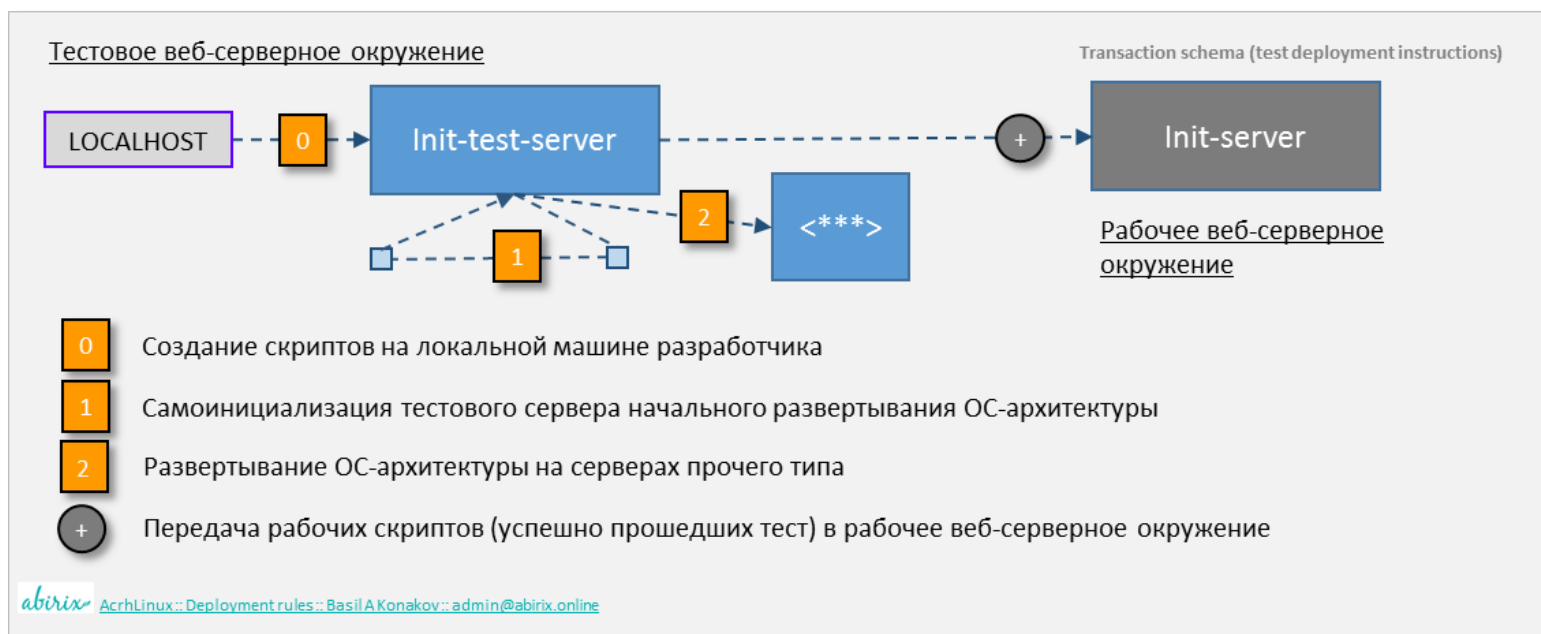
```
PRJ_VERSION=<<version>>;  
PRJ_MEMORY=<<this_memory>>;  
PRJ_TYPE=<<box_type>>;  
PRJ_CODE=<<fix_deploy_localhost>>;  
PRJ_NAME=<<this_prj>>;  
PRJ_KEY=<<fix_ssh_key>>;  
PRJ_HOSTNAME=<<this_hostname>>;  
PRJ_SFTP=<<this_prj>>;  
PRJ_SRCIP=<<ip_init>>;  
PRJ_ROOTPASS="<<this_pass_root>>";  
PRJ_GITUSER="<<ip_this>>git@<<this_git_domain>>";
```

7. Создание и тестирование скриптов инициализации для новых версий пула

Разработка и тестирование скриптов инициализации на рабочем сервере происходит только после тестирования скриптов на локальной машине разработчика и обязательном полноценном тестировании (на «голую систему») в пуле серверов тестового веб-серверного окружения Проекта.

В настоящее время для осуществления тестов, которых предполагается не много, и они будут не часто, выделено 2 сервера (Init-test-server и Prod-test-server), которых пока достаточно. Возможно увеличение числа серверов до 4 штук (для параллельного тестирования сразу нескольких версий), но это не в ближайшее время.

Общая схема работы DevOps по разработке нового ПО будет всегда осуществляться следующим образом (в четыре этапа или более простой вариант в 3 этапа, когда Init-test-server уже инициализирован - управление версия в таком случае уже предусмотрено через параметр <version>):



8. Транзакции развертывания рабочих серверов Проекта

Транзакции развертывания рабочего веб-серверного окружения для целей микро-сервисной архитектуры всей системы в целом (не только ВЕБ-СЕРВЕРА) представлены ниже.

Некоторые из указанных выше транзакций пока еще вообще не описаны, в то время как основные транзакции уже практически проработаны и в ряде случаев фактически осуществлены, в том числе (в квадратных скобках указан идентификатор транзакции на схеме):

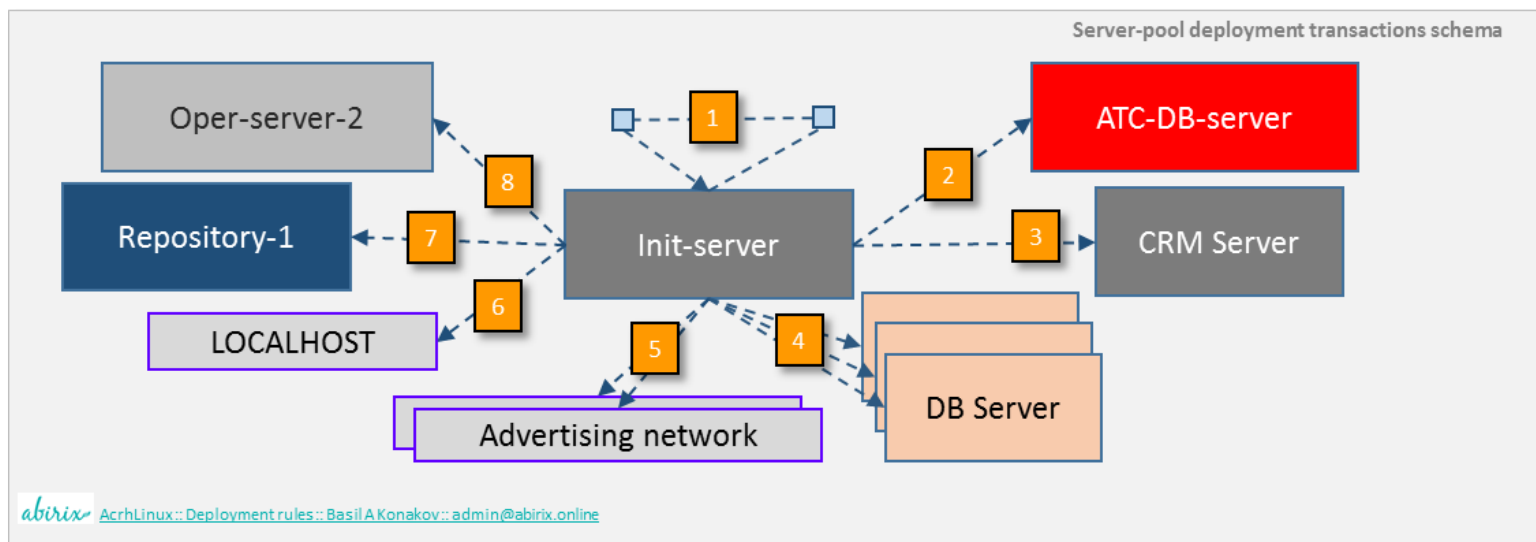
- 1) Транзакция самоинициализации рабочего сервера начального развертывания пула серверов на уровне ОС [1];
- 2) Транзакция развертывания рабочего веб-сервера с модулями CRM-системы [3];
- 3) Транзакция развертывания локального окружения на машине разработчика [6];
- 4) Транзакция развертывания сервера версионного контроля [7].

В ближайшее время будут завершены работы по тестированию, настройке и вводу в эксплуатацию:

- 5) Сервера интеграции с Sancom [2];
- 6) Сервера внутренних коммуникаций [8].

Остальные транзакции развертывания скорее всего будут завершены уже не в ближайшее время. Это такие транзакции, как:

- 7) Развертывание архитектуры распределенного хранения баз данных (с использованием репликации но пока без шардирования) [8];
- 8) Автоматизированного развертывание сети размещения рекламных материалов и привлечения трафика.



Server-pool deployment transactions schema

Рабочая / перспективная версия		
abirix_v001	0	Развертывание локального окружения на машине разработчика в точном соответствии с веб-серверным
abirix_v001	1	Самоинициализация рабочего сервера начального развертывания ОС-архитектуры
abirix_sancom1	2	Развертывание оболочки сервера интеграции с Sancom (данные SIP-АТС телефонии) – в процессе!
abirix_v001	3	Развертывание Веб-сервера с модулями CRM-системы
abirix_dbs1	4	Развертывание сервера для организации распределенного хранилища данных (в проекте 2й очереди)
abirix_lps1	5	Развертывание Веб-серверного окружения рекламных сайтов и веб-сервисов для целей маркетинга и продаж (в проекте 1й очереди)
abirix_repo1	6	Развертывание оболочки сервера версионного контроля скриптов веб-приложений
abirix_oper1	7	Развертывание оболочки сервера коммуникаций и информационного обмена внутри отдела разработчиков

abirix AcrhLinux::Deployment rules::Basil A Konakov::admin@abirix.online

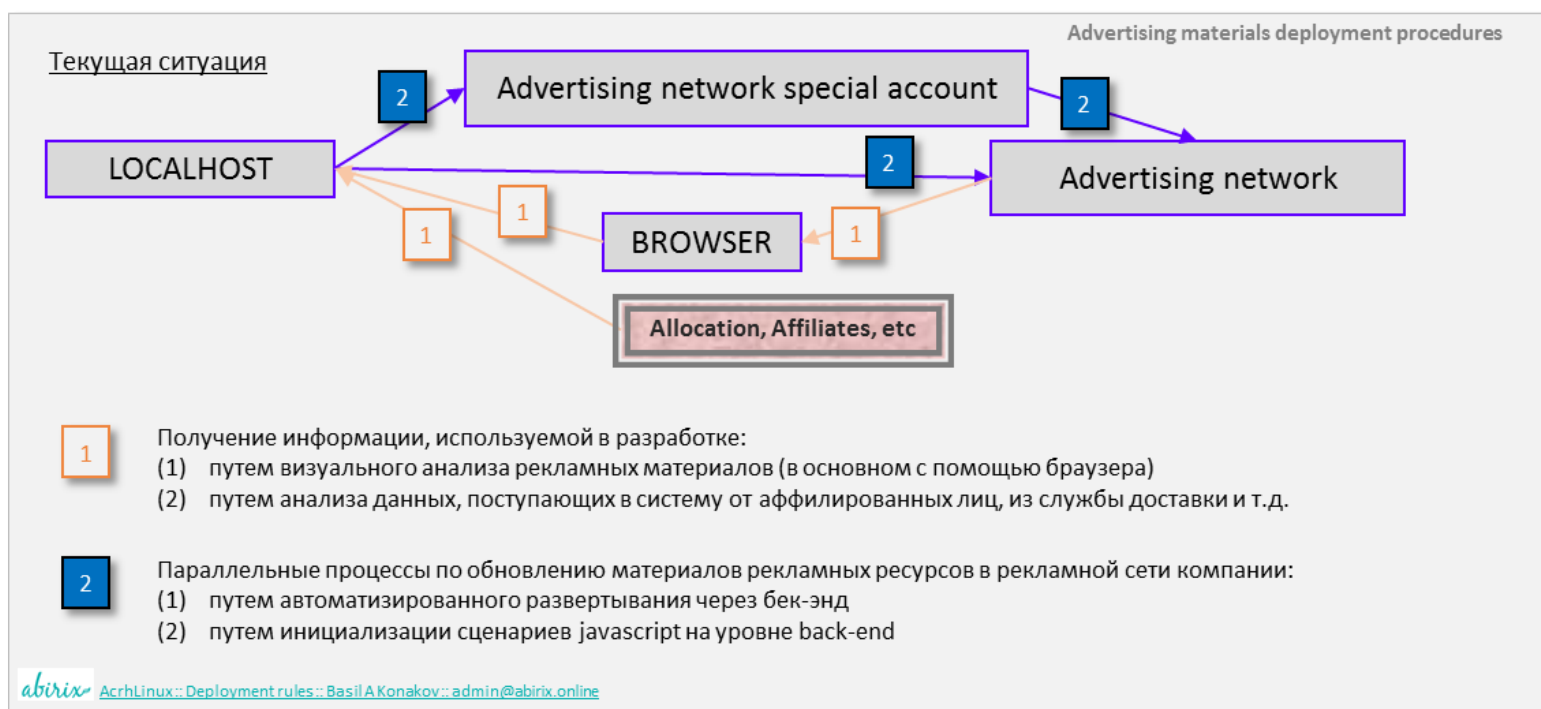
9. Системные транзакции, необходимые для комплексного развития системы

В настоящее время разработка приложений [LOCALHOST] в основном происходит ВРОДЕ КАК (информация получена из разрозненных источников, требует уточнения!) на базе информации, которая поступает к разработчикам после визуального анализа рекламных материалов статистами и аналитиками [из Adv.network с помощью браузера], а также из других систем (от аффилиатов, службы доставки и т.д.).

В итоге такая информация (как оно и должно быть) достаточно эффективно формируется в виде задач, которые позволяют так или иначе управлять ресурсами рекламных материалов. Однако, при этом все еще остается высокой вероятностью совершения ошибки «человеческого фактора» (вследствие невнимательности, утомленности, умысла или замешательства – такие параметры как цены, валюта и т.д. все же устанавливаются вручную), а также отсутствует возможность контролировать эффективность рекламных компаний на уровне отдельных продуктов и конкретного трафика.

Последнему обстоятельству во-многом способствует тот факт, что изменение контента используемых в настоящее время рекламных ресурсов осуществляется параллельно в несколько процессов:

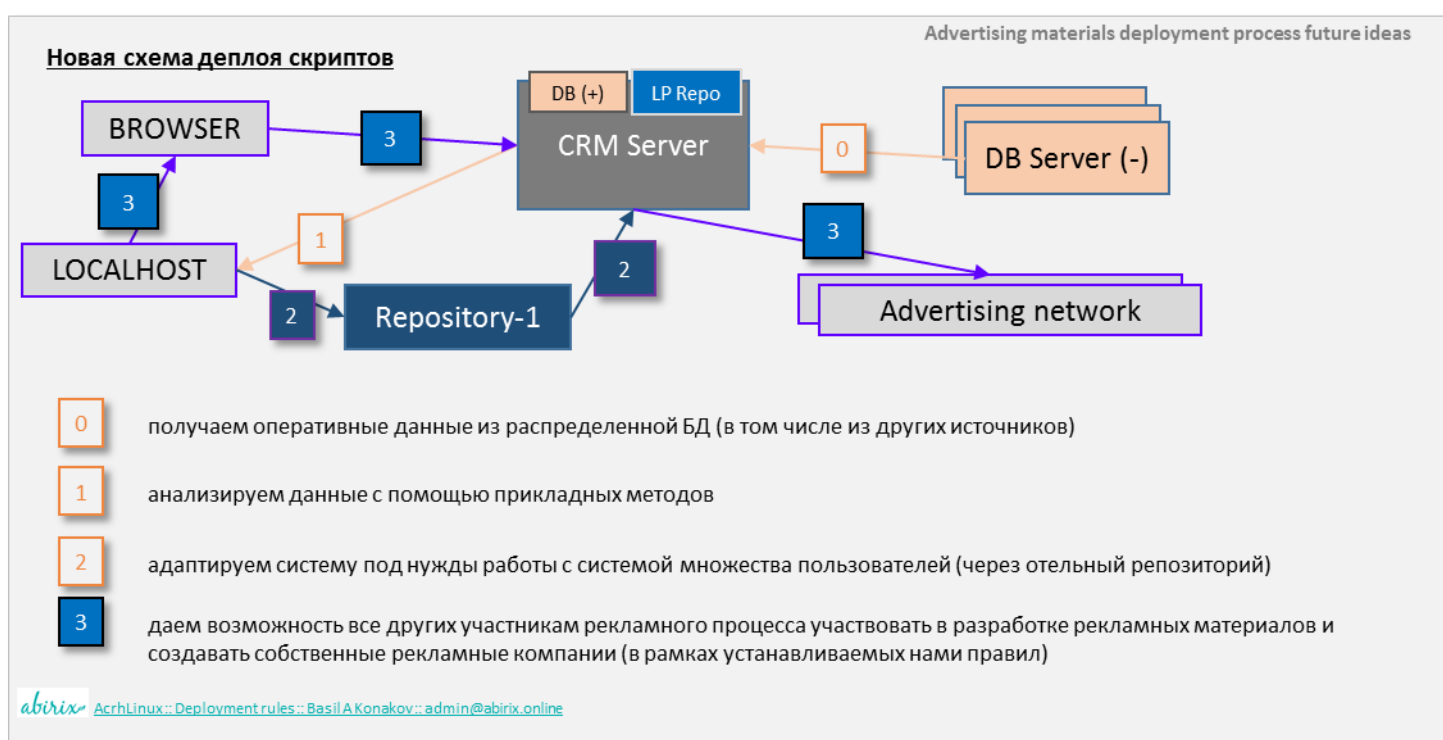
- 1) В первом случае информация поступает на сервер напрямую путем заливки данных через единый аккаунт развертывания системы, что в целом позволяет проводить тестирование, версионный контроль и управление через конфигурации, но при этом не учитывает принадлежность рекламных ресурсов той или иной рекламной компании, которая может быть не эффективной или не активной или иметь смещение приоритетов в сторону «других факторов» (например, с сейла на апсейл и т.п.), а значит уровень конверсии потенциально может быть какой угодно низкий, вплоть до нулевого уровня;
- 2) Дополнительно часть материалов на рекламных сайтах обновляется на лету (через фронт-энд), но при этом не позволяет осуществлять перенаправление рекламного трафика и/или интеграцию рекламных ресурсов в единую систему «presale-sale-upsale-sale/presale», что также нередко приводит к низкой конверсии.



Новая схема развертывания приложений позволяет избежать необходимости множественного контроля результатов развертывания, предоставляя возможность осуществления таких процедур непосредственным заказчикам / пользователям рекламных материалов.

Основных этапов новой схемы управления скриптами предполагается целых три:

- 1) Предполагается, что сама система будет разрабатываться с учетом множества анализируемых данных, которые будут поступать к разработчику в виде стандартизированной информации (процедуры 0 и 1, процедура 0 – на перспективу), а значит будут учитываться параметры рекламных компаний и рекламного трафика;
- 2) Деплой скриптов системы будет осуществляться также, как и сейчас, с помощью версионного контроля, но уже с помощью ssh-подключений к отдельному серверу - чтобы не было вероятности получения сбоя системы ввиду сбоев в работе других модулей, а также для озвученных ранее целей будет создан дополнительный модуль – CRM;
- 3) Деплой рекламных материалов будет осуществляться с помощью версионного контроля, но не организованного с помощью Git, что накладывало бы серьезные ограничения на возможности коллективной работы с такими материалами, а уже непосредственно внутри системы CRM.



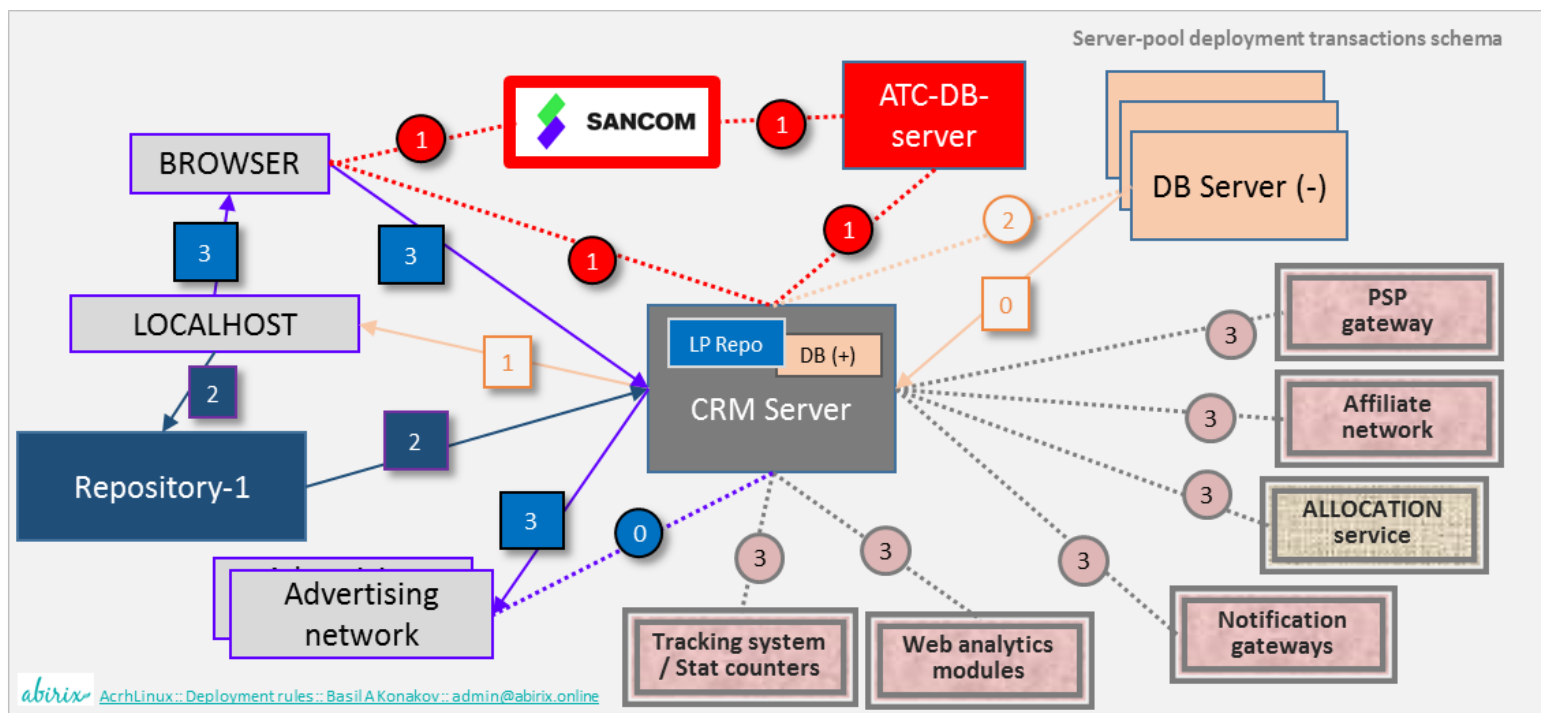
10. Планы по развитию системы в плане интеграции

Указанные здесь планы имеют краткосрочный и долгосрочный характер и указаны в общем. Детальное описание данного функционал представлено в соответствующем разделе «III. Advert CMI («Advertising system Content Management and Integration»).

В дальнейшем схема интеграции модулей системы между собой и с внешними модулями может быть следующей:

- 1) Интеграция системы CRM с функционалом ATC SANCOM (через браузер), а также через сервер получения информация от Sancom (1);
- 2) Интеграция системы CRM с функционалом распределенного хранилища данных (будет создан для целей highload), а пока это локальная база на сервере CRM;

- 3) Интеграция с внешними системами и веб-сервисами, в том числе: платежные системы, система логистики, система аффилиатов (рекламные партнеры), партнерские программы, системы веб-аналитики и трекеры:



11. Конфигурация всех учетных записей системы с учетом локальной разработки

