

8.

ОТКАЗОУСТОЙЧИВЫЙ КЛАСТЕР

Раздел «Отказоустойчивый кластер» включает следующие компоненты:

«Необходимое окружение и библиотеки»,

«Подготовительные этапы к установке кластера»,

«Действия в программе установке wiSLA»,

«Настройка скриптов для учёта кратковременных обрывов связи»,

«Действия по восстановлению работы кластера при выходе из строя одного из узлов ЦОД1»,

«Действия по восстановлению работы кластера при выходе из строя одного из узлов ЦОД2»,

«Действия по восстановлению работы кластера при выходе из строя третьей точки опоры».

- [Необходимое окружение и библиотеки](#)
- [Подготовительные этапы к установке кластера](#)
- [Действия в программе установки wiSLA](#)
- [Настройка скриптов для учёта кратковременных обрывов связи](#)
- [Действия по восстановлению работы кластера при выходе из строя одного из узлов ЦОД1](#)
- [Действия по восстановлению работы кластера при выходе из строя одного из узлов ЦОД2](#)
- [Действия по восстановлению работы кластера при выходе из строя третьей точки опоры](#)

Необходимое окружение и библиотеки

Необходимые пакеты для установщика на oracle linux 8

```
autogen-libopts-5.18.12-8.el8.x86_64.rpm      ntpdate-4.2.6p5-29.el7.centos.2.x86_64.rpm
compat-openssl10-1.0.2o-4.el8.x86_64.rpm     pv-1.6.6-7.el8.x86_64.rpm
dialog-1.3-13.20171209.el8.x86_64.rpm        python3-bcrypt-3.1.6-2.el8.1.x86_64.rpm
glibc-langpack-ru-2.28-225.0.4.el8_8.6.x86_64.rpm python3-paramiko-2.12.0-1.el8.noarch.rpm
libsodium-1.0.18-2.el8.x86_64.rpm            python3-pynacl-1.3.0-5.el8.x86_64.rpm
ntp-4.2.6p5-29.el7.centos.2.x86_64.rpm       uuid-1.6.2-43.el8.x86_64.rpm
```

- libnsl-2.28-225.0.4.el8_8.6.x86_64.rpm - пакет который ставился на 1-й и 4-й сервера
- compat-openssl10-1.0.2o-4.el8.x86_64.rpm - для pgpool
- uuid-1.6.2-43.el8.x86_64.rpm glibc-langpack-ru-2.28-225.0.4.el8_8.6.x86_64.rpm -Postgresql требует пакеты glibc-langpack-ru и uuid

На узлах 2,5(hadoop) должна быть организована "общая папка" , при создании файла в примонтированном gluster на одном узле, он должен появляться на другом

Скрипт для чистки всего(кроме pgpool), следует запускать перед установкой и после удаления из установщика.

Установщик не чистит папки в home директории. И установщик может не убрать процессы java, postgres, pgpool, если возникают проблемы при установке, запуске, остановки - проверить наличие процессов

```
for i in $(seq 1 7);
do
ssh 0001wislatest0$i sudo killall java
ssh 0001wislatest0$i sudo killall postgres
ssh 0001wislatest0$i 'rm -rf /opt/wisla5/* /home/wisla/{hadoop,zookeeper,hbase,postgresql}'
done
ssh 0001wislatest02 rm -rf /mnt/glusterVol/* #где glusterVol - куда примонтировали gluster.
```

На самом деле установщик при установке видит что папки в home есть и на glusterVol есть папка hadoop, установщик предлагает на каждую папку запрос об удалении. Однако лучшим вариантом будет "почистить скриптом", т.к. это упростит работу.

Подготовительные этапы к установке кластера

Преимущества кластера

Настройка отказоустойчивого кластера wiSLA позволяет решить 2 задачи:

- в случае отказа одного из ЦОД система сохраняет работоспособность;
- в кластере работает балансировка нагрузки, что позволяет более эффективно использовать аппаратные ресурсы серверов.

В примере будет показана установка системы wiSLA на отказоустойчивый контур, который включает в себя семь серверов, распределённых между двумя ЦОД (по три в каждом) и одним дополнительным сервером – «третьей точкой опоры» (см. рисунок 66). Для взаимодействия между серверами выделена подсеть «межсерверного взаимодействия».

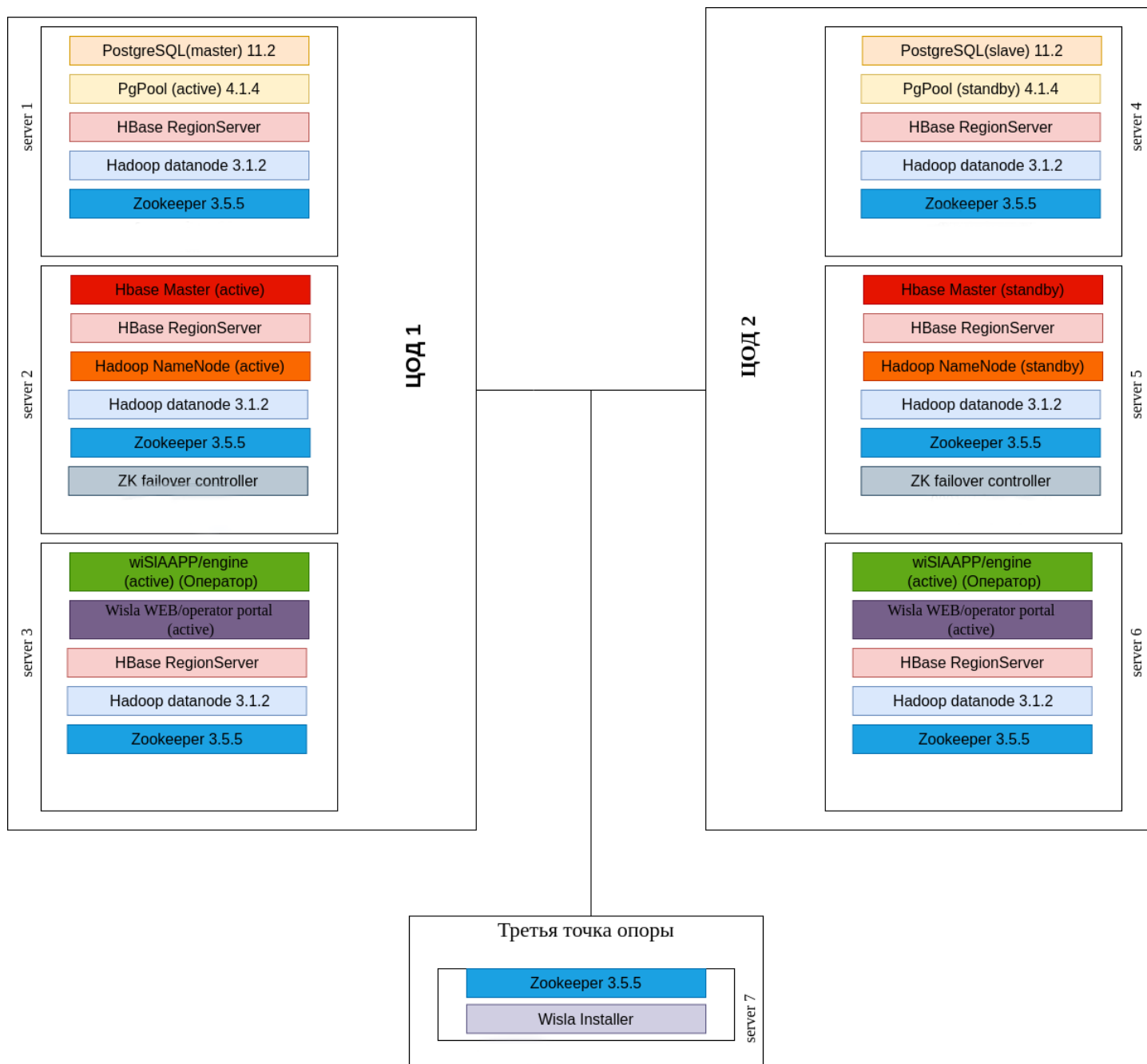


Рис. 66 Логическая группировка серверов в отказоустойчивом контуре wiSLA

Дополнительно могут быть развернуты wisla contactor portals (но не на 3 и 6 ноде, где уже есть портал оператора)

Отказоустойчивый контур должен состоять минимум из двух блоков (ЦОД), которые включают в себя сервера с основными компонентами, и дополнительного сервера (третьей точки опоры), с помощью которого контролируется доступность основных ЦОД и целостность кластера. Всего для отказоустойчивого контура должно быть выделено не менее пяти отдельных серверов. Желательно, чтобы аппаратная конфигурация серверов была одинаковой. В этом случае можно производить установку и обновление системы с помощью программы установки без ручного изменения параметров распределения оперативной памяти по компонентам.

В таблице приведён пример настройки отказоустойчивого контура, который содержит два ЦОД (по три сервера на каждом) и один дополнительный сервер (третья точки опоры).

Таблица 8 – Топология кластера (пример).

ЦОД / сервер	Имя сервера (hostname) / IP-адрес	Компоненты
--------------	-----------------------------------	------------

ЦОД1	wislaserver01 / 192.168.1.101	PostgreSQL: Master Pgpool HBase: HRegionServer Hadoop: DataNode Zookeeper
	wislaserver02 / 192.168.1.102	HBase: HMaster, HRegionServer Hadoop: NameNode, DataNode Zookeeper
	wislaserver03 / 192.168.1.103	APP-server WEB-server HBase: HRegionServer Hadoop: DataNode Zookeeper
ЦОД2	wislaserver04 / 192.168.1.104	PostgreSQL: Slave Pgpool HBase: HRegionServer Hadoop: DataNode Zookeeper
	wislaserver05 / 192.168.1.105	HBase: HMaster, HRegionServer Hadoop: NameNode, DataNode Zookeeper
	wislaserver06 / 192.168.1.106	APP-server WEB-server HBase: HRegionServer Hadoop: DataNode Zookeeper

Дополнительный сервер	wislaserver07 / 192.168.1.107	Zookeeper, wiSLA-Installer
-----------------------	-------------------------------	----------------------------


Предварительно для каждого сервера контура требуется базовая настройка, которая описана в разделе «Подготовка операционной системы к запуску программы установки». Для серверов с Hadoop NameNode (wislaserver02 и wislaserver05) потребуется дополнительно примонтировать блочное устройство, объем диска на обоих серверах должен быть одинаковым. Для серверов с Pgroup (wislaserver01 и wislaserver04) должен быть выделен IP-адрес в подсети межсерверного взаимодействия, который будет использовать Pgroup. Далее в примерах настройки используется «192.168.1.110».


На серверах с Pgroup (wislaserver01 и wislaserver04) также требуется убедиться в отсутствии alias-интерфейса

«eth0:10», где eth0 – корневой интерфейс, на котором настроена подсеть межсерверного взаимодействия, фактически может отличаться.

Настройка беспарольного доступа по SSH для межсерверного взаимодействия

Для корректной установки системы wiSLA и взаимодействия серверов контура должен быть организован беспарольный доступ по SSH под пользователем wisla по принципу «каждый с каждым». Также требуется беспарольный доступ под пользователем root между серверами с Pgroup и третьей точкой опоры. При этом для доступа с сервера на сервер используется hostname. Для этого требуется выполнить следующие шаги:

 !Внимание, возможно требуется после копирования длинных команд - заменять пробелы на пробелы

 !Внимание, при изменении топологии, возможно корректирование предлагаемых команд

1. На 7-й сервере добавить хосты в /etc/hosts

```
mkdir installDir; pushd installDir
host_prefix=0001wislatest0
echo 192.168.1.101 "$host_prefix"1 >> hostsForALL
echo 192.168.1.102 "$host_prefix"2>> hostsForALL
echo 192.168.1.103 "$host_prefix"3>> hostsForALL
echo 192.168.1.104 "$host_prefix"4>> hostsForALL
echo 192.168.1.105 "$host_prefix"5>> hostsForALL
echo 192.168.1.106 "$host_prefix"6>> hostsForALL
echo 192.168.1.107 "$host_prefix"7>> hostsForALL
cat hostsForALL | sudo tee -a /etc/hosts

hostnames="0001wislatest01 0001wislatest02 0001wislatest03 0001wislatest04 0001wislatest05 0001wislatest06 0001wislatest07"
hostnames_1_6="0001wislatest01 0001wislatest02 0001wislatest03 0001wislatest04 0001wislatest05 0001wislatest06"
hostnames_1_4_7="0001wislatest01 0001wislatest04 0001wislatest07"
```

2. На каждом сервере контура под пользователем wisla выполнить генерацию ключей, пароль оставить пустым (можно вообще 1 ключ создать и скопировать его везде):

```
su -l wisla
cd /home/wisla
ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa # или ssh-keygen -t rsa
```

Если не известен пароль, то необходимо:

```
зайти на хост на который нужно подключиться
скопировать содержимое ~/.ssh/*.pub #ssh-keygen -t rsa если нет файла
добавить в конец файла ~/.ssh/authorized_keys
```

4. Скопировать /etc/hosts на каждый хост

```
for i in $hostnames_1_6; do cat hostsForALL | ssh "$i" 'sudo tee -a /etc/hosts';done
for i in $hostnames_1_6; do ssh "$i" "sudo hostnamectl set-hostname $i";done
```

5. С каждого сервера контура под пользователем wisla выполнить копирование ключа на все остальные сервера для пользователя wisla и для пользователя root для части серверов(На серверах с Pgroup и дополнительном сервере под пользователем wisla выполнить копирование ключа на другие сервера с Pgroup или дополнительный сервер для пользователя root):

```
host_keys="$(for i in $hostnames;do ssh \"$i\" 'cat /home/wisla/.ssh/*.pub'; done)"

for i in $hostnames;do echo "$host_keys" | ssh "$i" tee -a /home/wisla/.ssh/authorized_keys; done

host_keys="$(for i in $hostnames_1_4_7;do ssh "$i" 'cat /home/wisla/.ssh/*.pub'; done)"

for i in $hostnames_1_4_7;do echo "$host_keys"| ssh "$i" 'sudo mkdir -p /root/.ssh; sudo tee -a /root/.ssh/authorized_keys'; done
```

6. Выполнить разовое подключение по SSH под пользователем wisla с каждого сервера на другие сервера, чтобы подтвердить добавление ключей удалённых серверов (по умолчанию отпечаток ключа будет храниться в файле /home/wisla/.ssh/known_hosts). При этом на вопрос «Are you sure you want to continue connecting (yes/no)?» требуется отвечать «yes». При подключении пароль удалённого сервера запрашиваться не должен.

Выполнить команду на каждом сервере под пользователем wisla:

```
for i in $hostnames; do ssh "$i" exit; done
```

Выполнить команду на 1 4 7 серверах под пользователем wisla:

```
for i in $hostnames_1_4_7; do ssh root@"$i" exit; done
```

В некоторых случаях доступ по root из ssh невозможен: следует проверить файл /etc/security/access.conf(ROOT:ALL) и файл /etc/ssh/sshd_config (PermitRootLogin yes)

Установка GlusterFS

GlusterFS – распределённая, параллельная, линейно масштабируемая файловая система с возможностью защиты от сбоев. Предварительно на серверах wislaserver02 и wislaserver05 должны быть созданы lvm-разделы абсолютно одинакового размера. Далее в примере используется «/dev/sdb1», фактически название раздела может отличаться.

Действия по установке и настройке GlusterFS выполняются под пользователем root.

Для установки GlusterFS на серверах wislaserver02 и wislaserver05 требуется выполнить следующие шаги:

1. Загрузить установочные пакеты GlusterFS по ссылке <ftp://ftp.wellink.ru/Deploies/Centos/cluster/glusterfs.tar.gz> ;
2. Подключиться к серверу по SSH под пользователем root;

3. Перейти в каталог с архивом, распаковать архив, установить необходимые пакеты glusterfs, glusterfs-server, glusterfs-cli,:

```
tar zxvf glusterfs.tar.gz
yum install glusterfs-3.5.2-1.el6.x86_64.rpm glusterfs-api-3.5.2-1.el6.x86_64.rpm
glusterfs-cli-3.5.2-1.el6.x86_64.rpm glusterfs-fuse-3.5.2-1.el6.x86_64.rpm glusterfs-georeplication-3.5.2-1.el6.x86_64.rpm glusterfs-
libs-3.5.2-1.el6.x86_64.rpm glusterfsserver-3.5.2-1.el6.x86_64.rpm
```

Настройка GlusterFS

Для настройки работы GlusterFS требуется выполнить следующие шаги:

1. Рекомендовано на серверах wislaserver02 и wislaserver05 отформатировать раздел для glusterfs в файловую систему xfs:

```
mkfs.xfs /dev/sdb1
```

2. Смонтировать раздел, добавить в автозапуск и запустить glusterfs на wislaserver02 и wislaserver05:

```
mkdir -p /mnt/gluster && mount /dev/sdb1 /mnt/gluster && mkdir -p /mnt/gluster/brick
echo "/dev/sdb1 /mnt/gluster xfs defaults 0 0" >> /etc/fstab
systemctl start glusterd # /etc/init.d/glusterd start
```

3. На сервере wislaserver02 выполнить команды:

```
gluster peer probe "$host_prefix"5
gluster volume create namenodevol rep 2 transport tcp "$host_prefix"2:/mnt/gluster/brick
"$host_prefix"5:/mnt/gluster/brick
gluster volume start namenodevol
mkdir /mnt/glusterVol; mount -t glusterfs "$host_prefix"2:/namenodevol /mnt/glusterVol/
chown wisla:wisla -R /mnt/glusterVol
```

4. На сервере wislaserver05 выполнить команды:

```
mkdir /mnt/glusterVol; mount -t glusterfs "$host_prefix"5:/namenodevol /mnt/glusterVol/
chown wisla:wisla -R /mnt/glusterVol
```

Внимание, следует протестировать что файл созданный на одном узле в /mnt/glusterVol ,должен появиться в том же месте на другом узле

5. На сервере wislaserver02 выполнить команды:

```
touch /mnt/glusterVol/test_file.txt
```

6. На сервере wislaserver05 проверить наличие файла и затем файл можно удалить

```
ls /mnt/glusterVol/test_file.txt
rm -f /mnt/glusterVol/test_file.txt
```

Действия в программе установки wiSLA

Шаги для установки wiSLA описаны в разделе «Работа с программой установки». Но есть ряд отличий в процессе, они будут приведены ниже.

Программа установки предварительно должна быть скопирована на сервер wislaserver07 в домашний каталог пользователя wisla. Права на файл должны быть у пользователя wisla, и файл должен быть исполняемым. Далее требуется выполнить действия, которые перечислены в блоке «Перечень действий для установки wiSLA», с учётом отличий в настройках, которые описаны ниже:

После запуска установщика можно подложить файл топологии и конфигурации по привычному пути, но это "на свой страх и риск".

1. Запуск программы установки.
2. Запуск установки системы.
3. Топология. Потребуется указать топологию, описанную выше в таблице 8:

```
Application servers: wislaserver03 wislaserver06
Operator Web servers: wislaserver03 wislaserver06
Contractor Web servers: wislaserver03 wislaserver06
Postgres main (single server): wislaserver01
Postgres slaves: wislaserver04
Pgpool servers: wislaserver01 wislaserver04
Zookeeper quorum: wislaserver01 wislaserver02 wislaserver03 wislaserver04 wislaserver05 wislaserver06
wislaserver07
Hadoop/HBase masters: wislaserver02 wislaserver05
Hadoop/HBase workers: wislaserver01 wislaserver02 wislaserver03 wislaserver04 wislaserver05 wislaserver06
```

4. Ожидание инициализации модулей .
5. Оценка параметров сервера и подбор оптимальных значений по распределению памяти.
Нужно учитывать, что программа установки делает оценку физических параметров того сервера, на котором была запущена. Поэтому, если аппаратная конфигурация серверов контура отличается, то следует рассчитать распределение памяти и вручную изменять предлагаемые настройки модулей системы по ходу установки.
6. Выбор версии и архитектуры Java Runtime Environment.
7. Настройка компонента Zookeeper.
8. Настройка компонента Hadoop:

```
Name directory: /mnt/glusterVol
Zookeeper quorum: wislaserver01:2181 wislaserver02:2181 wislaserver03:2181 wislaserver04:2181
wislaserver05:2181 wislaserver06:2181 wislaserver07:2181
Replication count: 4
```

Для настройки «Replication count» используется число, которое высчитывается по формуле: $(\text{HadoopWorkersCount} / 2) + 1$, где HadoopWorkersCount – количество серверов, заданное в топологии в строке «Hadoop/HBase workers».

Также стоит отметить, что путь hdfs://wisla. Надо в hosts прописать на 1,3,4,6 сервере 127.0.0.1 wisla. Что-то где-то прибито гвоздями и порт должен быть 8020. wisla:8020 или - wisla не имеет разницы, так как это порт по умолчанию.

9. Настройка компонента HBase, стоит проверить путь к хадупу [hdfs://wisl.a](https://wisl.a).

10. Настройка компонента PostgreSQL (без изменений). Возможно стоит взять более широкую маску сети(/24) причем раз это сеть, можно адрес сети написать
После шага 10 «Настройка компонента PostgreSQL» появится дополнительный шаг для конфигурирования Pgrool, на котором потребуется задать настройки:

```
Trust host or network: 192.168.1.0/24
Specifies the virtual IP address: 192.168.1.110
Specifies the netmask for virtual IP address: 255.255.255.0
Virtual interface: eth0
```

- Trust host or network – подсеть межсерверного взаимодействия.
- Specifies the virtual IP address – IP-адрес, который выделен для работы pgrool в подсети межсерверного взаимодействия.
- Specifies the netmask for virtual IP address – маска подсети межсерверного взаимодействия.
- Virtual interface – корневой интерфейс, на котором настроена подсеть межсерверного взаимодействия на серверах wislaserver01 и wislaserver04.

Указываем сеть в которой работает СУЩЕСТВУЮЩИЙ интерфейс, и указываем IP который НЕ используется. Также указываем СУЩЕСТВУЮЩИЙ интерфейс. pgrool на него навесит ip (причем ip уже должен быть назначен на всех хостах которые подключены в сеть, куда смотрит интерфейс). возможно стоит взять более широкую маску сети(/24)

- a) для корректной работы pgrool желателен отдельный сетевой интерфейс. К примеру он может быть вланом.
- b) pgrool назначает на интерфейс ip, а :10 это номер псевдонима.
- ⚠ c) в сети, на которую смотрит интерфейс должны быть назначены ip в той же подсети.
- d) активный pgrool назначает на этот интерфейс ip.
- e) если вы обращаетесь к PostgreSQL то должны быть подключены к сети, в которой есть pgrool

11. Настройка компонента WildFly:

- app host for portal ... - пишем соответствующие ip для данных хостов

12. Настройка топологии wiSLA.

13. Настройки модуля сбора данных:

- wiProbe destination - указываем, куда будут программные или аппаратные зонды стучаться

14. Настройки интеграции LDAP.

15. Настройки дополнительных ресурсов wiSLA.

16. Настройка рассылки уведомлений.

17. Настройка оператора портала.

⚠ Обращаем ваше внимание, если вы получаете доступ к portalу с помощью проброса портов или через прокси сервер, то вам необходимо отредактировать пункт HOST и в Whitelisted domains установить необходимые IP-адреса.

18. Подтверждение настроек.
19. Автоматический запуск после установки.
20. Реиндексация wildfly (не бд).
21. Начало работы с порталом.

Настройка скриптов для учёта кратковременных обрывов связи

Для настройки учёта кратковременных обрывов связи необходимо выполнить следующие шаги:

1. Предварительно следует создать каталог `/opt/wisla5/scripts/` на серверах ЦОД1 и ЦОД2. Владелец каталогов должен быть пользователь `wisla`. На серверах `wislaserver03` и `wislaserver06` эти каталоги будут созданы автоматически после установки `wiSLA`. Далее приведена команда для создания каталога:

```
mkdir -p /opt/wisla5/scripts
```

Архив со скриптами можно загрузить по ссылке ftp://ftp.wellink.ru/Deploies/Centos/cluster/failover_scripts.tar.gz, скопировать на любой из серверов кластера для пользователя `wisla` и распаковать с помощью команды:

```
tar zxvf failover_scripts.tar.gz
```

2. Владелец скриптов и конфигурационных файлов должен быть пользователь `wisla`. На сервере `wislaserver04` в каталоге `/opt/wisla5/pgpool/current/sbin/` переименовать файл `ifconfig` на `ifconfig1` с помощью команды:

```
sudo mv /opt/wisla5/pgpool/current/sbin/ifconfig /opt/wisla5/pgpool/current/sbin/ifconfig1
```

3. Скопировать скрипты на сервера контура.

Для копирования рекомендуется использовать консольную утилиту `scp`. Пример команды копирования:

```
scp TRUSTED_SERVERS wisla@wislaserver01:/opt/wisla5/scripts/
```

В примере администратор находится в каталоге с распакованными скриптами, копирует файл

`TRUSTED_SERVERS` на сервер `wislaserver01` в каталог `/opt/wisla5/scripts/`.

Таблица 7 – Размещение скриптов для учёта кратковременных обрывов связи.

Сервер	Файл	Путь
wislaserver01	TRUSTED_SERVERS	/opt/wisla5/scripts/
	isolation_test.sh	/opt/wisla5/scripts/
	pgpool_connect.sh	/opt/wisla5/scripts/
	DATA_PROCESSING_CENTER_SERVERS	/opt/wisla5/pgpool/current/
	TRUSTED_SERVERS	/opt/wisla5/pgpool/current/
	failover.sh	/opt/wisla5/pgpool/current/
wislaserver02	TRUSTED_SERVERS	/opt/wisla5/scripts/

	isolation_test.sh	/opt/wisla5/scripts/
wislaserver03	TRUSTED_SERVERS	/opt/wisla5/scripts/
	isolation_test.sh	/opt/wisla5/scripts/
wislaserver04	TRUSTED_SERVERS	/opt/wisla5/scripts/
	isolation_test.sh	/opt/wisla5/scripts/
	pgpool_connect.sh	/opt/wisla5/scripts/
	DATA_PROCESSING_CENTER_SERVERS	/opt/wisla5/pgpool/current/
	TRUSTED_SERVERS	/opt/wisla5/pgpool/current/
	failover.sh	/opt/wisla5/pgpool/current/
	ifconfig	/opt/wisla5/pgpool/current/sbin/
wislaserver05	TRUSTED_SERVERS	/opt/wisla5/scripts/
	isolation_test.sh	/opt/wisla5/scripts/
wislaserver06	TRUSTED_SERVERS	/opt/wisla5/scripts/
	isolation_test.sh	/opt/wisla5/scripts/

4. Сконфигурировать файлы со списком серверов и скрипты. Файл должен завершаться пустой строкой. Редакторы vi (vim) и nano делают это автоматически при сохранении файла.

wislaserver01:

В файле /opt/wisla5/scripts/TRUSTED_SERVERS указать список серверов ЦОД2 и третью точку опоры:

```
wislaserver04
wislaserver05
wislaserver06
wislaserver07
```

В файле /opt/wisla5/pgpool/current/DATA_PROCESSING_CENTER_SERVERS указать список серверов ЦОД1:

```
wislaserver01
wislaserver02
wislaserver03
```

В файле /opt/wisla5/pgpool/current/TRUSTED_SERVERS указать третью точку опоры:

```
wislaserver07
```

В конфигурационном файле /opt/wisla5/pgpool/current/etc/pgpool.conf в опции trusted_servers (по умолчанию строка 442) указать сервер с pgpool ЦОД2 и третью

точки опоры:

```
trusted_servers = 'wislaserver04,wislaserver07'
```

wislaserver02:

В файле /opt/wisla5/scripts/TRUSTED_SERVERS указать список серверов ЦОД2 и третью точку опоры:

```
wislaserver04  
wislaserver05  
wislaserver06  
wislaserver07
```

wislaserver03:

В файле /opt/wisla5/scripts/TRUSTED_SERVERS указать список серверов ЦОД2 и третью точку опоры:

```
wislaserver04  
wislaserver05  
wislaserver06  
wislaserver07
```

wislaserver04:

В файле /opt/wisla5/scripts/TRUSTED_SERVERS указать список серверов ЦОД1 и третью точку опоры:

```
wislaserver01  
wislaserver02  
wislaserver03  
wislaserver07
```

В файле /opt/wisla5/pgpool/current/DATA_PROCESSING_CENTER_SERVERS указать список серверов ЦОД2:

```
wislaserver04  
wislaserver05  
wislaserver06
```

В файле /opt/wisla5/pgpool/current/TRUSTED_SERVERS указать третью точку опоры:

```
wislaserver07
```

В скрипте /opt/wisla5/pgpool/current/sbin/ifconfig указать (отредактировать) дополнительный IP адрес, который был зарезервирован для работы pgpool:

```
PGPOOL_ALIAS="192.168.1.110"
```

В конфигурационном файле /opt/wisla5/pgpool/current/etc/pgpool.conf в опции trusted_servers (строка 442) указать сервер с pgpool ЦОД1 и третью точку опоры:

```
trusted_servers = 'wislaserver01,wislaserver07'
```

wislaserver05:

В файле /opt/wisla5/scripts/TRUSTED_SERVERS указать список серверов ЦОД1 и третью точку опоры:

```
wislaserver01  
wislaserver02  
wislaserver03  
wislaserver07
```

wislaserver06:

В файле /opt/wisla5/scripts/TRUSTED_SERVERS указать список серверов ЦОД1 и третью точку опоры:

```
wislaserver01  
wislaserver02  
wislaserver03  
wislaserver07
```

Добавить в crontab задачи на запуск скриптов. Для этого под пользователем wisla нужно выполнить команду:

```
crontab -e
```

Открывается конфигурация cron для пользователя wisla в редакторе vim.

wislaserver01:

```
*/1 * * * * /opt/wisla5/scripts/isolation_test.sh  
*/1 * * * * /opt/wisla5/scripts/pgpool_connect.sh
```

wislaserver02:

```
*/1 * * * * /opt/wisla5/scripts/isolation_test.sh
```

wislaserver03:

```
*/1 * * * * /opt/wisla5/scripts/isolation_test.sh
```

wislaserver04:

```
*/1 * * * * /opt/wisla5/scripts/isolation_test.sh  
*/1 * * * * /opt/wisla5/scripts/pgpool_connect.sh
```

wislaserver05:

```
*/1 * * * * /opt/wisla5/scripts/isolation_test.sh
```

wislaserver06:

```
*/1 * * * * /opt/wisla5/scripts/isolation_test.sh
```

Перезапустить систему, используя программу установки wiSLA – в разделе Maintenance вначале выполнить «Stop all», затем «Start all».

Действия по восстановлению работы кластера при выходе из строя одного из узлов ЦОД1

Отказоустойчивый кластер фактически рассчитан только на один отказ одного из узлов кластера, после которого требуется восстановление его работы. В случае выхода из строя двух узлов система перестаёт функционировать.

при статусе:

```
psql -p 19999 -c "show pool_nodes"
```

```
node_id | hostname | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
0 | wislaserver01 | 5432 | 3 | 0.500000 | standby
1 | wislaserver04 | 5432 | 2 | 0.500000 | primary
```

1)стопнуть вторую ноду из инсталлера

2)запустить вторую ноду из инсталлера

```
/opt/wisla5/pgpool/current/bin/pcp_recovery_node -p 9898 -h 10.198.2.17 -U wisla 1
#Password: wisla
```

Дождаться выполнения -> все ок

Выход из строя ЦОД1

При выходе из строя ЦОД1 все компоненты переходят в активный режим на ЦОД2:

- Pgpool на сервере wislaserver04 переводит PostgreSQL в режим master и активирует alias на интерфейс на сервере;
- Nadoor и HBase на сервере wislaserver05 переходят в режим active;
- wiSLA на сервере wislaserver06 забирает себе все задачи, которые ранее были распределены между двумя серверами.

В данном случае требуется восстановить согласованность данных и перенести активные модули на ЦОД1. Потребуется полная остановка и запуск системы на обоих ЦОД. Для этого нужно выполнить следующие действия:

1. Восстановить связь между узлами кластера.
2. В отдельной сессии SSH на сервере wislaserver07 под пользователем wisla запустить программу установки.
3. В разделе «Maintenance» -> «wiSLA management» остановить wiSLA на всех серверах:

```
Stop_all
```

4. В разделе «Maintenance» -> «Pgpool management» запустить Pgpool на сервере wislaserver01:

```
Start on wislaserver01
```

5. Убедиться в том, что Pgpool на сервере wislaserver01 активирован. Для этого в разделе «Statuses» выбрать «Pgpool status».

6. В отдельной сессии SSH открыть shell сервера wislaserver04 под пользователем wisla.
7. На сервере wislaserver04 выполнить команду для просмотра состояния узлов PostgreSQL:

```
psql -p 19999 -c "show pool_nodes"
```

8. Убедиться в том, что сервер wislaserver04 имеет status «2» (активен) и role «primary», а wislaserver01 имеет status «3» (неактивен) и role «standby»:

```
node_id | hostname | port | status | lb_weight | role  
-----+-----+-----+-----+-----+-----  
0 | wislaserver01 | 5432 | 3 | 0.500000 | standby  
1 | wislaserver04 | 5432 | 2 | 0.500000 | primary
```

9. На сервере wislaserver04 выполнить команду:

```
ip a
```

и убедиться в **наличии** alias «eth0:10» на интерфейсе «eth0» .

10. В отдельной сессии SSH открыть командную строку сервера wislaserver01 под пользователем wisla.

11. На сервере wislaserver01 выполнить команду:

```
ip a
```

и убедиться в **отсутствии** alias «eth0:10» на интерфейсе «eth0».

12. На сервере wislaserver04 выполнить команду:

```
LD_LIBRARY_PATH=/opt/wisla5/pgpool/current/lib/ pcp_recovery_node -d 0 127.0.0.1 9898 wisla wisla 0
```

13. Дождаться выполнения команды, должно снова появиться приглашение для ввода.

14. Снова выполнить команду для просмотра состояния узлов PostgreSQL и убедиться в том, что сервер wislaserver01 уже имеет status «2» (активен) и role «primary», а wislaserver04 сохранил status «2» (активен), но изменил role на «standby».

```
[wisla@wislaserver04 ~]$ psql -p 19999 -c "show pool_nodes"  
node_id | hostname | port | status | lb_weight | role  
-----+-----+-----+-----+-----+-----  
0 | wislaserver01 | 5432 | 2 | 0.500000 | primary  
1 | wislaserver04 | 5432 | 2 | 0.500000 | standby
```

15. В открытой программе установки остановить Pgpool на сервере wislaserver04. Для этого в разделе «Maintenance» -> «Pgpool management» выполнить:

```
Stop on wislaserver04
```

16. В открытой программе установки остановить PostgreSQL на сервере wislaserver04. Для этого в разделе «Maintenance» -> «Postgresql management» выполнить:

```
Stop on wislaserver04
```

17. В открытой программе установки запустить Pgpool на сервере wislaserver04. Для этого в разделе «Maintenance» -> «Pgpool management» выполнить:

```
Start on wislaserver04
```

18. Убедиться в том, что Pgpool на сервере wislaserver04 активирован. Для этого в разделе «Statuses» выбрать «Pgpool status».

19. На сервере wislaserver04 выполнить команду:

```
ip a
```

и убедиться в **отсутствии** alias «eth0:10» на интерфейсе «eth0».

20. На сервере wislaserver01 выполнить команду:

```
ip a
```

и убедиться в **наличии** alias «eth0:10» на интерфейсе «eth0».

21. На сервере wislaserver01 выполнить команду для просмотра состояния узлов PostgreSQL и убедиться в том, что сервер wislaserver01 имеет status «2» (активен) и role «primary», а wislaserver04 имеет status «3» (неактивен) и role «standby».

```
[wisla@wislaserver01 ~]$ psql -p 19999 -c "show pool_nodes"
node_id | hostname | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
0 | wislaserver01 | 5432 | 2 | 0.500000 | primary
1 | wislaserver04 | 5432 | 3 | 0.500000 | standby
```

22. На сервере wislaserver01 выполнить команду (обратить внимание на замену числа 0 на 1 в конце команды):

```
LD_LIBRARY_PATH=/opt/wisla5/pgpool/current/lib/ pcp_recovery_node -d 0 127.0.0.1 9898 wisla wisla 1
```

23. Дождаться выполнения команды, должно снова появиться приглашение для ввода.

24. На сервере wislaserver01 повторно выполнить команду для просмотра состояния узлов PostgreSQL и убедиться в том, что сервер wislaserver01 сохранил status «2» (активен) и role «primary», а wislaserver04 изменил status на «2» (активен), role осталась «standby».

```
[wisla@wislaserver01 ~]$ psql -p 19999 -c "show pool_nodes"
node_id | hostname | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
0 | wislaserver01 | 5432 | 2 | 0.500000 | primary
1 | wislaserver04 | 5432 | 2 | 0.500000 | standby
```

25. На сервере wislaserver04 выполнить команду:

```
tail -n 10 /home/wisla/postgresql/postgres.log
```

и убедиться в наличии строки:

```
database system is ready to accept read only connections
```

26. В отдельной сессии SSH открыть командную строку сервера wislaserver02 под пользователем root.

27. На сервере wislaserver02 выполнить команды:

```
systemctl start glusterd #/etc/init.d/glusterd start
mount -t glusterfs wislaserver05:/namenodevol /mnt/glusterVol/
```

28. На сервере wislaserver02 убедиться в наличии непустого каталога /mnt/glusterVol/current/

```
ls /mnt/glusterVol/current/
```

29. В открытой программе установки в разделе «Maintenance» -> «HBase management» выполнить остановку HBase на всех серверах:

```
Stop_all
```

30. В открытой программе установки в разделе «Maintenance» -> «Hadoop management» выполнить остановку Hadoop на всех серверах:

Stop_all

31. В открытой программе установки в разделе «Maintenance» -> «Zookeeper management» выполнить остановку Zookeeper на всех серверах:

Stop_all

32. В открытой программе установки в разделе «Maintenance» -> «Zookeeper management» выполнить запуск Zookeeper на всех серверах:

Start_all

33. В открытой программе установки в разделе «Maintenance» -> «Hadoop management» выполнить запуск Hadoop на всех серверах:

Start_all

34. В открытой программе установки в разделе «Maintenance» -> «HBase management» выполнить запуск HBase на всех серверах:

Start_all

35. В открытой программе установки в разделе «Maintenance» -> «wiSLA management» выполнить запуск wiSLA на всех серверах:

Start_all

36. После запуска wiSLA проверить состояние компонентов системы в разделе «Statuses» -> «All statuses» и убедиться в работоспособности портала wiSLA.

Действия по восстановлению работы кластера при выходе из строя одного из узлов ЦОД2

Отказоустойчивый кластер фактически рассчитан только на один отказ одного из узлов кластера, после которого требуется восстановление его работы. В случае выхода из строя двух узлов система перестаёт функционировать.

Выход из строя ЦОД2

При выходе из строя ЦОД2 все компоненты остаются в активном режиме работы на ЦОД1. В данном случае требуется восстановить согласованность данных и восстановить работу системы на ЦОД2. Потребуется полная остановка и запуск системы на обоих ЦОД. Для этого нужно выполнить следующие действия:

1. Восстановить связь между узлами кластера.
2. В отдельной сессии SSH на сервере wislaserver07 под пользователем wisla запустить программу установки.
3. В разделе «Maintenance» -> «wiSLA management» остановить wiSLA на всех серверах:

```
Stop_all
```

4. В разделе «Maintenance» -> «Pgpool management» запустить Pgpool на сервере wislaserver04:

```
Start on wislaserver04
```

5. Убедиться в том, что Pgpool на сервере wislaserver04 активирован. Для этого в разделе «Statuses» выбрать «Pgpool status».
6. В отдельной сессии SSH открыть командную строку сервера wislaserver01 под пользователем wisla.
7. На сервере wislaserver01 выполнить команду для просмотра состояния узлов PostgreSQL:

```
psql -p 19999 -c "show pool_nodes"
```

8. Убедиться в том, что сервер wislaserver01 имеет status «2» (активен) и role «primary», а wislaserver04 имеет status «3» (неактивен) и role «standby»:

```
node_id | hostname | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
0 | wislaserver01 | 5432 | 2 | 0.500000 | primary
1 | wislaserver04 | 5432 | 3 | 0.500000 | standby
```

9. На сервере wislaserver01 выполнить команду:

```
ip a
```

и убедиться в **наличии** alias «eth0:10» на интерфейсе «eth0» .

10. В отдельной сессии SSH открыть командную строку сервера wislaserver04 под пользователем wisla.

11. На сервере wislaserver04 выполнить команду:

```
ip a
```

и убедиться в **отсутствии** alias «eth0:10» на интерфейсе «eth0».

12. На сервере wislaserver01 выполнить команду:

```
LD_LIBRARY_PATH=/opt/wisla5/pgpool/current/lib/ pcp_recovery_node -d 0 127.0.0.1 9898 wisla wisla 1
```

13. Дождаться выполнения команды, должно снова появиться приглашение для ввода.

14. На сервере wislaserver01 снова выполнить команду для просмотра состояния узлов PostgreSQL и убедиться в том, что сервер wislaserver01 сохранил status «2» (активен) и role «primary», а wislaserver04 изменил status на «2» (активен), role осталась «standby».

```
[wisla@wislaserver01 ~]$ psql -p 19999 -c "show pool_nodes"
node_id | hostname | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
0 | wislaserver01 | 5432 | 2 | 0.500000 | primary
1 | wislaserver04 | 5432 | 2 | 0.500000 | standby
```

15. На сервере wislaserver04 выполнить команду:

```
ip a
```

и убедиться в **отсутствии** alias «eth0:10» на интерфейсе «eth0».

16. На сервере wislaserver01 выполнить команду:

```
ip a
```

и убедиться в **наличии** alias «eth0:10» на интерфейсе «eth0» .

17. На сервере wislaserver04 выполнить команду:

```
tail -n 10 /home/wisla/postgresql/postgres.log
```

и убедиться в наличии строки:

```
database system is ready to accept read only connections
```

18. В отдельной сессии SSH открыть командную строку сервера wislaserver05 под пользователем root.

19. На сервере wislaserver05 выполнить команды:

```
systemctl start glusterd #/etc/init.d/glusterd start
mount -t glusterfs wislaserver02:/namenodevol /mnt/glusterVol/
```

20. На сервере wislaserver05 убедиться в наличии непустого каталога /mnt/glusterVol/current/

```
ls /mnt/glusterVol/current/
```

21. В открытой программе установки в разделе «Maintenance» -> «HBase management» выполнить остановку HBase на всех серверах:

```
Stop_all
```

22. В открытой программе установки в разделе «Maintenance» -> «Hadoop management» выполнить остановку Hadoop на всех серверах:

```
Stop_all
```

23. В открытой программе установки в разделе «Maintenance» -> «Zookeeper management» выполнить остановку Zookeeper на всех серверах:

```
Stop_all
```

24. В открытой программе установки в разделе «Maintenance» -> «Zookeeper management» выполнить запуск Zookeeper на всех серверах:

Start_all

25. В открытой программе установки в разделе «Maintenance» -> «Hadoop management» выполнить запуск Hadoop на всех серверах:

Start_all

26. В открытой программе установки в разделе «Maintenance» -> «HBase management» выполнить запуск HBase на всех серверах:

Start_all

27. В открытой программе установки в разделе «Maintenance» -> «wiSLA management» выполнить запуск wiSLA на всех серверах:

Start_all

28. После запуска wiSLA проверить состояние компонентов системы в разделе «Statuses» -> «All statuses» и убедиться в работоспособности портала wiSLA.

Действия по восстановлению работы кластера при выходе из строя третьей точки опоры

Данная ситуация не влияет на работоспособность контура, но грозит отказом системы в случае выхода из строя одного из ЦОД. Для восстановления работы третьей точки опоры требуется выполнить следующие действия:

1. Восстановить связь между узлами кластера.
2. На сервере wislaserver07 запустить программу установки.
3. В разделе «Maintenance» -> «Zookeeper management» запустить Zookeeper на сервере wislaserver07:

Start on wislaserver07

4. Проверить состояние компонентов системы в разделе «Statuses» -> «All statuses» и работоспособность wiSLA.