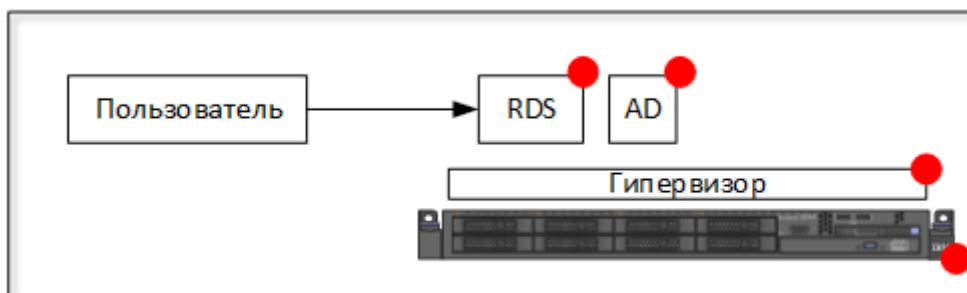


## План настройки:

1. **Различные варианты архитектуры решений MS RDS (эта статья)**
2. Описание MS RDS в Windows server 2012 и плана моей настройки (тут)
3. Настройка домена в Windows server 2012 R2: Active Directory, DNS, DHCP
4. Настройка дублирующих служб Active Directory, DNS, DHCP
5. Развертывание RDS и создание кластера NLB
6. Развертывание DFS для хранения данных пользователей
7. Настройка премощаемых профилей и перенаправленных папок

В своих статьях по настройке, я всегда пытаюсь показывать архитектуру, которую рекомендует производитель или разработчик программного обеспечения. ВСЕГДА у ВСЕХ существуют так называемые best practice или reference architecture, в которых показано, как лучшим образом построить масштабируемую и отказоустойчивую систему. Я написал у ВСЕХ? За исключением Microsoft... Может в этом есть большая политика или еще что-то, но для для служб RDS я не смог найти рекомендаций, только какие-то старые статьи и примеры простых настроек, которые не претендуют на полноценные гайды. Интересно написана вот эта статья, с виду все правильно, но для небольшой компании не подойдет, слишком громоздкая схема и дорогая в реализации в плане оборудования и лицензий (их автор не рассматривает). Я тоже подготовил свои вариации на тему терминального сервера от Microsoft, и вот как я мыслю, от простого к сложному.

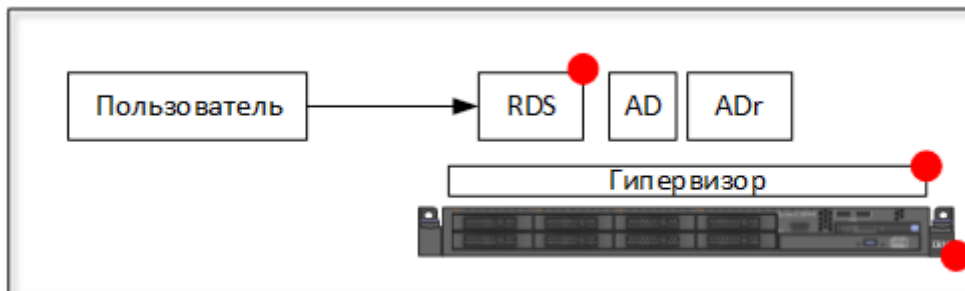
### Вариант 1



Самый простой вариант, известный очень давно, с ним спорить никто не станет, да и придаться не к чему, кроме сомнительной отказоустойчивости данного решения. Хотя есть в компании, где такие установки работают годами и ничего с ними не происходит. На сервер установлен гипервизор (любой), настроены две виртуальные машины: одна Active Directory, DNS, DHCP и вторая со службой удаленных рабочих столов (RDS) и сервером лицензий. Данная схема может успешно держать до 80-100 пользователей. Точки отказа помечены красными кружками. Из строя могут выйти:

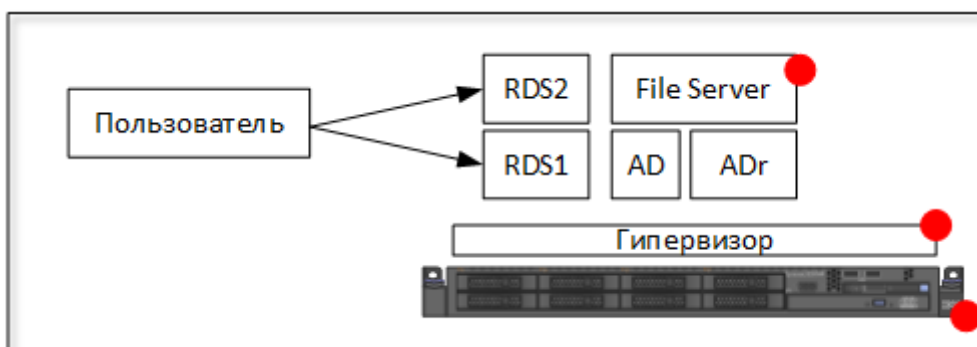
- сервер RDS (программная то) — все сессии рвутся, пользователи не работают
- AD, DNS, DHCP (программная то) — существующие сессии продолжают работать, все попытки новых подключения заканчиваются неудачей
- гипервизор (программная то) — все выключается никто не работает больше до восстановления
- сервер (аппаратная) — ничего не работает до починки сервера

## Вариант 2



Создаем еще одну виртуальную машину и резервируем важные для работы сервисы Active Directory, DNS, DHCP. Избавляемся от одной программной точки отказа.

## Вариант 3



Делаем еще один или несколько терминальных серверов (RDS), в основном для того, чтобы большему числу пользователей дать терминальный доступ, рекомендуется 40-50 человек на один терминальный сервер. Также придется добавить файловый сервер (программная то), где будут храниться профили пользователей и перенаправленные папки. Если пользователь подключается к одному из терминальных серверов, то на этот сервер загружается его профиль с файлового сервера и подключаются перенаправленные папки. Пользователей можно раскидывать между серверами с помощью DNS Round Robin или NLB cluster.

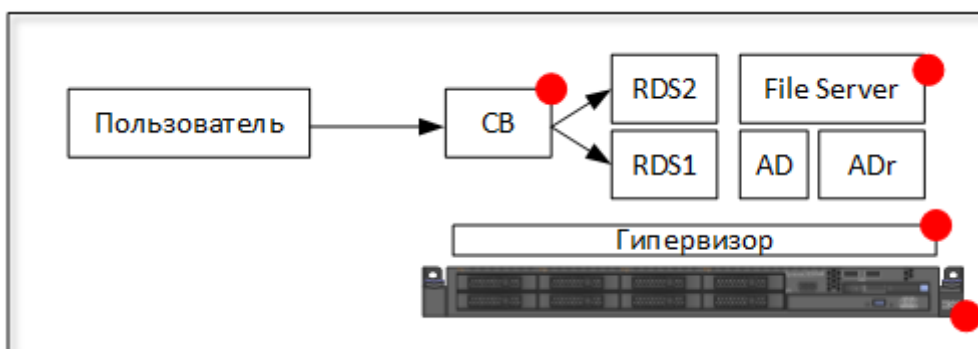
**DNS round robin** настраивается очень легко, в DNS создаются записи с одинаковыми именами и разными IP адресами. Пользователи обращаются на адрес и случайным образом получают один из IP адресов и попадают на один из терминальных серверов. Опасности DNS round robin:

- **нет распределения нагрузки** между серверами, может оказаться, что на одном из серверов больше пользователей чем на другом. Или один сервер будет загружен на 80%, а второй на 20%. *Данную проблему можно решить, нарастив количество терминальных серверов, тогда нагрузку будет распределять уже гипервизор, который умеет динамически распределять оперативную память и процессорную мощность.*
- **двойные сессии**, если пользователь отключился по каким-то причинам, а затем переподключается, то может легко попасть на другой сервер и откроется еще одна сессия. А в старой могут остаться открытые документы, программы, браузер и другая недоделанная работа. *Можно настроить автоматическое отключение неактивных сессий, это приучит пользователей чаще сохраняться и двойных сессий не будет.*

- **отказ в подключении** — DNS round robin не проверяет доступность IP адреса, который он выдает, и это самое неприятное в этом простом алгоритме. Поэтому, если один из терминальных серверов выйдет из строя, то при подключении пользователь может все равно получить его IP адрес и подключение не пройдет. Дальше два варианта, или пользователь не оставляет попытки подключения и в конце концов получает рабочий IP адрес или начинает звонить в тех поддержку. *Данную проблему решает только замена DNS round robin на NLB cluster.*

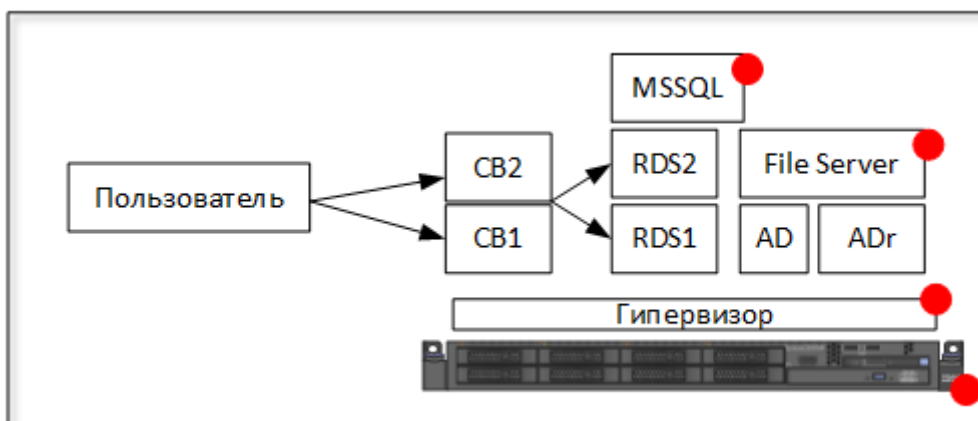
**NLB cluster** — встроенная фича Windows server, которая устанавливается через мастер добавления ролей сервера. В отличие от DNS round robin не перенаправляет пользователей на недоступные в данный момент серверы. Из минусов остаются невозможность распределять нагрузку и двойные сессии.

#### Вариант 4



Чтобы распределять нагрузку у Microsoft RDS есть специальная роль — Connection Broker. Он отслеживает нагрузку, количество пользователей на серверах, уже открытые сессии и перенаправляет новые подключения или переподключения на подходящие серверы RDS. Добавляя новый сервер с Connection Broker мы получаем новую программную точку отказа.

#### Вариант 5

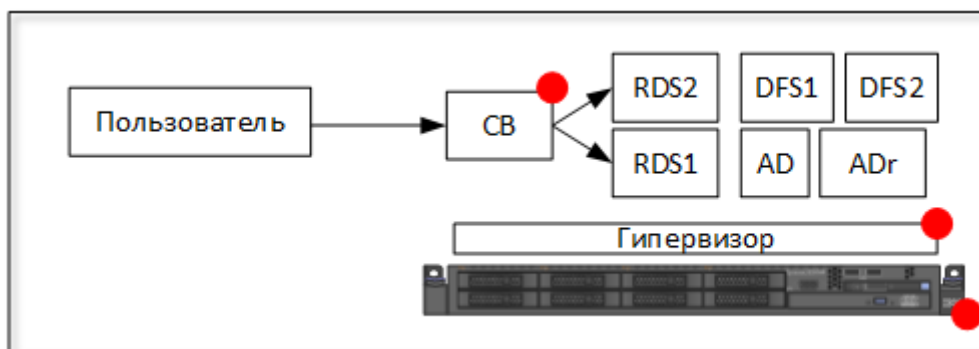


Добавляем дублирующий Connection Broker, но чтобы они работали вместе, обязательно нужен MSSQL сервер, который становится новой точкой отказа. Причем MSSQL отказоустойчивый кластер настроить и сложно и дорого. Чтобы сделать это

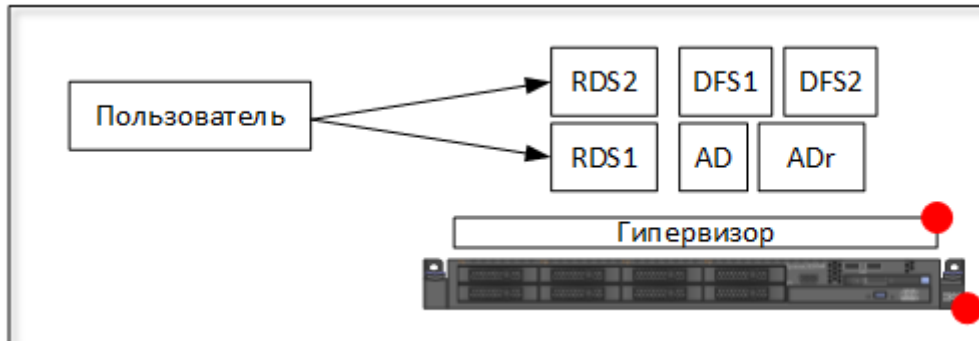
нормально понадобится общая папка на системе хранения данных и лицензия MS SQL server Standard, обойтись версией MSSQL Express не получится. На мой взгляд, в эту сторону имеет смысл двигаться только в больших инфраструктурах, где уже есть отказоустойчивый кластер MS SQL. В любом случае, сделать это на одном физическом сервере не получится.

Распределять пользователей между двумя и более Connection Broker также можно средствами DNS Round Robin или NLB Cluster.

### Вариант 6-7 (финальные для одного физического сервера)

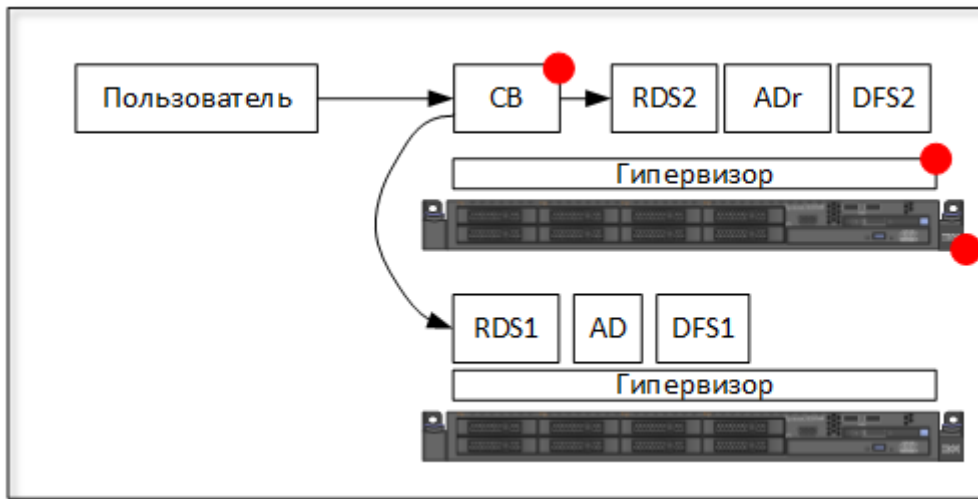


Возвращаемся к варианту с одним Connection Broker. От точки отказа файлового сервера избавляемся, заменяем связкой DFS серверов. Это будет финальный вариант для одного физического сервера.



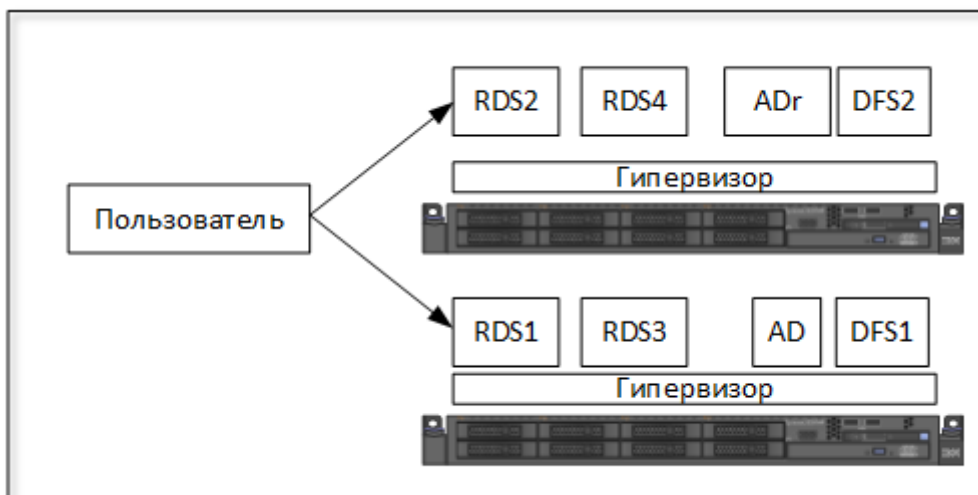
С одной стороны отсутствие Connection Broker лишает распределения нагрузки, с другой стороны убираем точку отказа. Рекомендуется использование NLB cluster на RDS серверах.

### Вариант 8



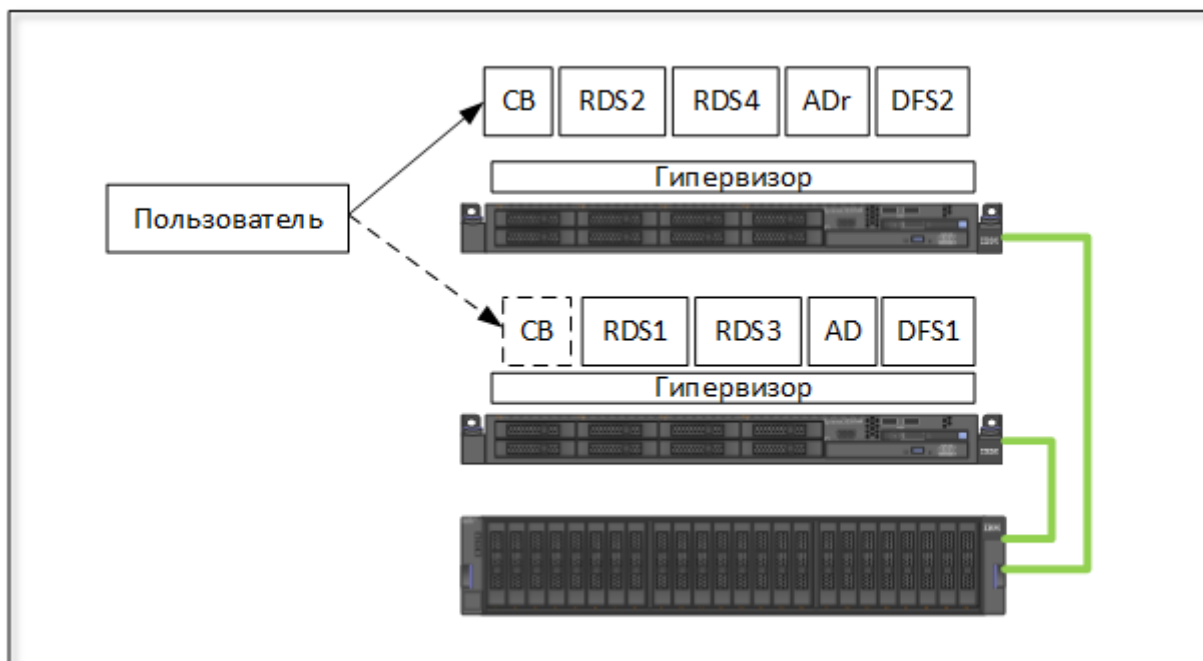
Добавляем второй физический сервер, что добавляет нам производительности. Остается точка отказа Connection Broker (программная), гипервизор (программная то) и сервер на котором находится CB (аппаратная то)

### Вариант 9



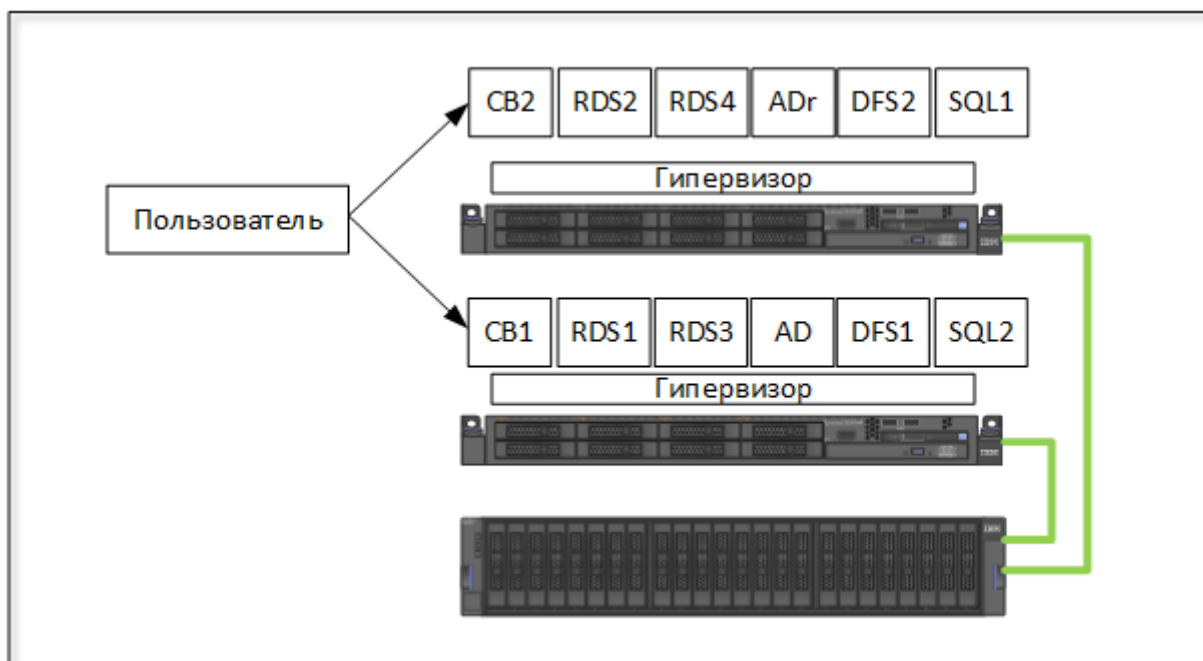
Явных точек отказа в данном варианте нет. Распределение нагрузки через NLB cluster. Масштабировать решение легко, никаких проблем с этим быть не должно.

### Вариант 10



Если добавить в решение систему хранения данных, то можно собрать отказоустойчивый кластер и защитить Connection Broker от аппаратного отказа сервера на котором он запущен. В High Availability кластере виртуальная машина с CB будет автоматически перезапущена на втором сервере. Напомню, что виртуальная машина занимается распределением нагрузки между терминальными серверами. Данный вариант не защищает от программного сбоя (BSOD например) в виртуальной машине CB.

### Вариант 11



По логике Microsoft эта схема самая правильная, и, конечно, самая дорогая. Connection broker работает в отказоустойчивом режиме, используя в свою очередь отказоустойчивый кластер MS SQL, работающий в режиме active-passive (лицензия MS SQL Standard) или в режиме active-active (лицензия MS SQL Enterprise)

# ВЫВОДЫ

Как вывод ко всем этим вариантам настройки терминальных ферм можно сказать следующее. Выбор того или иного варианта будет обусловлен: бюджетом на проект, возможностью кратковременного простоя. По опыту могу сказать, что чем меньше элементов в архитектуре решения, тем стабильнее оно работает.

0 Комментариев **ITsave**

 **Войти** ▾

 Рекомендовать 3

 Поделиться

Новое в начале ▾



Начать обсуждение...

Прокомментируйте первым.

ТАКЖЕ НА ITSAVE

## Сравнение серверов 2 процессора 4 ядра

3 комментария • 3 года назад



**halfcat** — Спасибо Вам огромное за проделанную работу. Когда пытаешься проанализировать ситуацию с

## windows install

1 комментарий • 2 года назад



**JeKa** — с 8 по 13 нет картинок :(

## vmware view

8 комментариев • 2 года назад



**Сергей Костюков** — Все сделал как было сказано выше и все получилось !!!, но вот странность удаленно(из локалки все

## 7915E4G

2 комментария • 3 года назад



**itsave** — Первое и самое верное, обновить прошивки сервера. Если ошибки не исчезают, можно обращаться в

 Подписаться  Добавить Disqus на свой сайт [Добавить Disqus](#) [Добавить](#)

ITsave.ru 2016

Zerif Lite создано с помощью WordPress