Описание и пример настройки HA-кластера JumpServer

JumpServer (в том числе Community Edition) полностью поддерживает кластеризацию HA без каких-либо ограничений. В этой статье я покажу, как это работает.

Зачем нужен HA-кластер для JumpServer

HA (High Availability) кластер для JumpServer необходим для обеспечения высокой доступности системы и минимизации простоев. Он позволяет:

1. **Избежать простоев**: Если один из узлов кластера выходит из строя, другой продолжает обслуживать запросы пользователей.
2. **Обеспечить отказоустойчивость**: Кластеризация позволяет системе автоматически переключаться на доступные узлы при возникновении ошибок.
3. **Увеличить производительность**: Нагрузка распределяется между несколькими узлами, что улучшает отклик системы при одновременном использовании большого числа пользователей.
4. **Повысить надежность хранения данных**: Использование общих ресурсов, таких как MySQL и Redis, с поддержкой кластеризации минимизирует риск потери данных.
5. **Масштабируемость**: Кластер можно легко расширить, добавив дополнительные узлы для обслуживания большего числа пользователей и задач.

Эта архитектура особенно важна для организаций, где JumpServer используется как критически важная система доступа и контроля.

Компоненты кластера JumpServer.

Ноды\Узлы JumpServer - основные узлы кластера, сервера с установленным JumpServer, каждый сервер не содержит "полезных" данных, можно клонировать\копировать, удалять\добавлять узлы JumpServer.

База данных MySQL\PostreSQL - основная СУБД для хранения всех данных JumpServer: хранит настройки системы, параметры устройств, учетных записей, пароли к целевым системам. Также по умолчанию хранит текстовые логи сессий: команды SSH, SQL запросы, введенные команды с клавиатуры в RDP сессии.
По умолчанию JumpServer создает\использует PostreSQL внутри контейнера на том же сервере, где устанавливаете JumpServer.

База данных Redis - вспомогательная база данных для кэширования, может быть как единой для всего кластера так и отдельными базами для каждой ноды кластера.

По умолчанию JumpServer создает\использует Redis внутри контейнера на том же сервере, где устанавливаете JumpServer.

Хранилище видеозаписей - по умолчанию хранит записи сессий в папке с продуктом **$folder/core/data/media,** где **$folder** - папка указанная в основном конфиг-файле, по умолчанию **VOLUME_DIR=/data/jumpserver**. Через веб-интерфейс продукта можно настроить внешнее хранилище видеозаписей: **SFTP, S3, Ceph, MinIo и другие**

Хранилище логов команд - по умолчанию хранятся в основной БД, через веб интерфейс можно настроить хранение команд и запросов в **Elasticssearch**.

Балансировщик - обычно на базе HAProxy, но можно использовать любой.

Архитектура кластера.

Обычно кластер JumpServer состоит из 2 и более узлов кластера, которые:

- подключены к общей БД(или кластеру) MySQL\PostreSQL
- подключены к общему Redis(или каждый к своему)
- имеют общую СХД для хранения видеозаписей:
  - общая папка данными **$folder/core/data/ (реализуется обычно с помощью NFS сервера)**
  ИЛИ
  - общая внешняя СХД для хранения сессий: SFTP, S3, Ceph, MinIo и другие
  ИЛИ
  - отдельная СХД для каждой ноды
- имеют общую СХД для записей логов команд:
  - на базе общей БД(по умолчанию)
  ИЛИ
  - на базе общей внешней Elasticsrarch
  ИЛИ
  - отдельный Elasticsearch для каждой ноды кластера
- балансировщик

Пример создания кластера HA JumpServer из 2 узлов

Пример создания кластера JumpServer
- с единой общей БД MySQL
- с единой общей БД Redis
- с общей папкой для хранения видеозаписей **$folder/core/data/** с помощью NFS сервера Linux

Для этого нам потребуется:

1. **Сервер с NFS, MySQL, Redis**:
   - 4 CPU, 8 ГБ оперативной памяти.
2. **Узел JumpServer Node1**:
   - 4 CPU, 8 ГБ оперативной памяти, 100Гб свободного места на диске
3. **Узел JumpServer Node2**:
   - 4 CPU, 8 ГБ оперативной памяти, 100Гб свободного места на диске
4. **Сервер HAProxy** (или другой балансировщик нагрузки).

1. Подготовка сервера с NFS, MySQL и Redis

- **Сервер**: Ubuntu 22.04, IP: `10.10.50.10`

Установка и настройка NFS

Команды могут отличаться для других версий Linux, но в целом нужно создать общую папку:

```
sudo apt install nfs-kernel-server
sudo mkdir -p /data
sudo chown -R nobody:nogroup /data/
sudo chmod 777 /data/
sudo nano /etc/exports
```

Добавьте строку в файл `/etc/exports`:

```
/data 10.10.50.10/24(rw,sync,no_subtree_check)
```

Примените настройки и перезапустите NFS-сервис:

```
sudo exportfs -a
sudo systemctl restart nfs-kernel-server
```

Установка и настройка MySQL

Инструкции зависят от версии ОС. Для создания базы данных и пользователя выполните:

```
mysql -uroot
mysql> create database jumpserver default charset 'utf8';
mysql> set global validate_password_policy=LOW;
mysql> create user 'jumpserver'@'%' identified by 'KXOeyNgDeTdpeu9q';
mysql> grant all on jumpserver.* to 'jumpserver'@'%';
mysql> flush privileges;
mysql> exit;
```

Не забудьте настроить файрволл для открытия порта MySQL (`3306`).

Установка и настройка Redis

Инструкции зависят от версии ОС. После установки Redis выполните:

```
sed -i "s/bind 127.0.0.1/bind 0.0.0.0/g" /etc/redis.conf
sed -i "561i maxmemory-policy allkeys-lru" /etc/redis.conf
sed -i "481i requirepass KXOeyNgDeTdpeu9q" /etc/redis.conf
```

Это позволит доступ к Redis с паролем `KXOeyNgDeTdpeu9q`. Обязательно используйте уникальный пароль для вашего сервера. Откройте порт `6379` в файрволле.

2. Установка JumpServer

Установка первой ноды JumpServer

Монтирование каталога NFS

Установите клиент NFS, смонтируйте папку и настройте автоматическое монтирование при загрузке:

```
sudo apt install nfs-common
mkdir -p /opt/jumpserver/core/data
mount -t nfs 10.10.50.10:/data /opt/jumpserver/core/data
echo "10.10.50.10:/data /opt/jumpserver/core/data nfs defaults 0 0" >> /etc/fstab
```

Настройка конфигурации JumpServer

Редактируйте файл `config-example.txt` в папке установщика:

```
# Измените следующие параметры, остальные оставьте по умолчанию.
# ВАЖНО: SECRET_KEY должен совпадать на всех узлах JumpServer, иначе данные не будут расшифровываться.

VOLUME_DIR=/opt/jumpserver

SECRET_KEY=
BOOTSTRAP_TOKEN=
LOG_LEVEL=ERROR
SESSION_EXPIRE_AT_BROWSER_CLOSE=True

# MySQL

DB_HOST=10.10.50.10
DB_PORT=3306
DB_USER=jumpserver
DB_PASSWORD=KXOeyNgDeTdpeu9q
DB_NAME=jumpserver

# Redis

REDIS_HOST=10.10.50.10
REDIS_PORT=6379
REDIS_PASSWORD=KXOeyNgDeTdpeu9q

# KoKo Lion
SHARE_ROOM_TYPE=redis
REUSE_CONNECTION=False
```

Запустите установку:

```
./jmsctl.sh install
```

После завершения установки вы получите значения для:

```
SECRET_KEY=kWQdmdCQKjaWlHYpPhkNQDkfaRulM6YnHctsHLlSPs8287o2kW
BOOTSTRAP_TOKEN=KXOeyNgDeTdpeu9q
```

Установка второй ноды JumpServer

Установите клиент NFS, смонтируйте папку точно также как на первой ноде.

При редактировании файла конфигурации JumpServer **заполните значения** `SECRET_KEY` **и** `BOOTSTRAP_TOKEN` , полученные после установки первой ноды:

```
VOLUME_DIR=/opt/jumpserver

SECRET_KEY=kWQdmdCQKjaWlHYpPhkNQDkfaRulM6YnHctsHLlSPs8287o2kW
BOOTSTRAP_TOKEN=KXOeyNgDeTdpeu9q
LOG_LEVEL=ERROR
SESSION_EXPIRE_AT_BROWSER_CLOSE=True

# MySQL

DB_HOST=10.10.50.10
DB_PORT=3306
DB_USER=jumpserver
DB_PASSWORD=KXOeyNgDeTdpeu9q
DB_NAME=jumpserver

# Redis

REDIS_HOST=10.10.50.10
REDIS_PORT=6379
REDIS_PASSWORD=KXOeyNgDeTdpeu9q

# KoKo Lion
SHARE_ROOM_TYPE=redis
REUSE_CONNECTION=False
```

И запустите установку

```
./jmsctl.sh install
```

Результат

После завершения настройки вы получите два узла JumpServer, которые используют один MySQL/Redis сервер и хранилище NFS. Вы можете использовать любой из узлов для доступа к целевым устройствам или настроить HAProxy для автоматического перенаправления пользователей на активный узел.

Версия #3
Сергей Попцов создал 22 января 2025 11:25:11
Сергей Попцов обновил 23 января 2025 10:41:32