

Свой репозиторий РОСА

Материал из ROSA Wiki

Эта статья описывает создание своего репозитория пакетов для операционных систем ROSA Fresh/Chrome версии 12 и новее.

Содержание

- 1 Введение и терминология
- 2 Зеркалирование репозиториев
 - 2.1 О зеркалировании
 - 2.2 Зеркалирование через rsync
 - 2.2.1 Установка rsync
 - 2.2.2 Использование rsync для зеркалирования
 - 2.2.3 Доступ к репозиториям РОСА Хром через rsync
 - 2.3 Зеркалирование через dnf reposync
- 3 Создание своего репозитория
- 4 Локальный репозиторий
- 5 Доступ к репозиторию по сети
 - 5.1 Доступ к репозиторию по HTTP
 - 5.2 Доступ к репозиторию по NFS

Введение и терминология

Репозиторий пакетов — хранилище RPM-файлов с дополнительными файлами-метаданными.

Создать свой репозиторий может быть нужно для, например:

- размещения используемых сторонних пакетов для их установки и обновления из единого места,
- создания зеркала (точной копии) официальных репозиториев внутри локальной сети предприятия.

Примечание

Приведенные команды следует выполнять от непривилегированного пользователя (не root); команды, которые необходимо выполнять с правами root, написаны через sudo.

Подразумевается, что сервером, на котором размещается репозиторий, является система на базе ROSA Fresh/Chrome 12 (rosa2021.1) и новее, однако его можно разместить и на других ОС.

Репозитории Росы 12 называются rosa2021.1, репозитории Росы 13 — rosa13.

Зеркалирование репозиториев

О зеркалировании

Процесс зеркалирования представляет из себя создание и поддержание в актуальном состоянии точной копии репозиториев, размещенных на официальных серверах ROSA. В данном разделе описано, как сделать такую копию репозитория. Приведен наиболее простой способ, не

обеспечивающий консистентность зеркала для внешнего потребителя в момент его обновления.

Зеркалирование через rsync

Установка rsync

Для зеркалирования предлагается использовать утилиту rsync. Если она не установлена, выполните:

```
sudo dnf install rsync
```

Использование rsync для зеркалирования

Создайте каталоги для размещения копий репозиториев:

```
sudo mkdir -p /srv/mirror/x86_64 /srv/mirror/i686
```

Запустите синхронизацию. В приведенном ниже примере исключаются каталоги большого веса с debuginfo и SRPM, которые в большинстве случаев не нужны.

```
sudo rsync \
  -av --progress \
  --exclude SRPMS \
  --exclude 'debug_*' \
  --exclude __REMOVED \
  --exclude '*-rpm-backup' \
  rsync://mirror.rosa.ru/rosa/rosa2021.1/repository/{x86_64,i686} /srv/mirror/
```

Создание регулярно выполняемой задачи описано в соответствующем разделе.

Доступ к репозиториям РОСА Хром через rsync

Доступ к репозиториям Росы Хром осуществляется по ключу.

Пример для репозиториев РОСА Хром 12:

```
RSYNC_PASSWORD= rsync <...> rsync://ваш_ключ@mirror.rosa.ru/chrome-rosa2021.1/ <...>
```

Пример для репозиториев сертифицированной ФСТЭК версии РОСА Хром 12F:

```
RSYNC_PASSWORD= rsync <...> rsync://ваш_ключ@mirror.rosa.ru/rosa2021.15/ <...>
```

Для доступа к репозиториям rosa-rosa2021.1 и rosa2021.15 используются разные ключи. Ключ тот же, что и для доступа к репозиториям по HTTP, который прописывается в `/etc/dnf/vars/token`.

Ключ является логином, а пароль пустой.

Зеркалирование через dnf reposync

Для зеркалирования по HTTP можно использовать `dnf reposync`:

```
dnf reposync \
  --repofrompath main64,http://mirror.rosa.ru/rosa/rosa2021.1/repository/x86_64/main/release \
  --download-metadata --delete --norepopath --remote-time \
```

```
--repopid main64 \  
--download-path /srv/rosa2021.1/x86_64/main/release
```

Для полной копии репозитория следует зазеркалить компоненты репозитория main, contrib и non-free, main следует зазеркалить не только для x86_64, но и для i686, поскольку в 64-битных ОС используются 32-битные пакеты.

Для репозитория РОСА Хром с доступом по ключу URL репозитория будут следующими:

- для несертифицированной версии Росы Хром 12: `http://ваш_ключ:@mirror.rosa.ru/rosa/chrome-rosa2021.1/repository/x86_64/main/release/`
- для актуального репозитория сертифицированной ФСТЭК версии Росы Хром 12F (обновления для 12F.*): `http://ваш_ключ:@mirror.rosa.ru/rosa2021.15/repository/x86_64/main/release/`
- для онлайн-копии сертифицированного ФСТЭК репозитория Росы Хром 12F (копия второго диска из комплекта поставки): `http://ваш_ключ:@mirror.rosa.ru/cert-repo/12F.1/x86_64/`

Если зайти по этим ссылкам через браузер, то ключ будет логином, а пароль пустой.

Адрес `cert-repo.rosalinux.ru` эквивалентен `mirror.rosa.ru`.

Создание своего репозитория

Если имеются сторонние RPM-пакеты, то из них можно сделать репозиторий. Необходимо установить утилиту для создания метаданных репозитория:

```
sudo dnf install createrepo_c
```

Создайте каталог для репозитория:

```
sudo mkdir -p /srv/repo/x86_64
```

Положите RPM-пакеты в каталог `/srv/repo/x86_64`.

Создайте/обновите метаданные репозитория:

```
sudo createrepo_c /srv/repo/x86_64
```

Для обновления репозитория удалите и/или добавьте RPM-пакеты в каталоге `/srv/repo/x86_64` и повторно запустите команду создания (обновления) метаданных.

Для проверки можно вывести список пакетов в этом репозитории пакетным менеджером:

```
dnf --repofrompath local,/srv/repo/x86_64 --disablerepo '*' --enablerepo local list
```

Локальный репозиторий

Подключить каталог с репозиторием можно локально. Для этого создайте в папке `/etc/yum.repos.d/` файл аналогичный тем что там уже имеются. Приведите строку с `baseurl` к такому виду:

```
baseurl=/home/user/my-repo
```

Должно получиться что-то вроде:

```
[myrepo]
name=My repo
baseurl=/path/to/my/repo
gpgcheck=0
enabled=1
```

Доступ к репозиторию по сети

Доступ к репозиторию по HTTP

Настройка HTTP-сервера

Для доступа к репозиторию по сети достаточно запустить любой веб-сервер, умеющий отдавать статические файлы. В качестве примера приведем настройку `angie` (наследник `nginx`).

Установите веб-сервер `angie`:

```
sudo dnf install angie
```

Откройте конфигурационный файл на редактирование:

```
sudo nano /etc/angie/angie.conf
```

Приведете его к необходимому виду. Ниже приведен минимальный пример на основе конфигурационного файла по умолчанию.

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name localhost;
        location / {
            root /srv;
            autoindex on;
            index index.html index.htm;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

Для выхода из консольного редактора `nano` нажмите `Ctrl+X`, `Enter`, `Ctrl+O`.

Подробнее про настройку `angie` можно прочитать в документации.

Запустите `angie` и добавьте его в автозапуск:

```
sudo systemctl enable --now angie
```

Попробуйте зайти по адресу `http://localhost:80` (`http://localhost:80`) (или иной IP-адрес), должно отобразиться содержимое каталога `/srv`.

Можно проще

Если репозиторий не будет иметь особых нагрузок или нужен вам для тестирования, вместо полноценного веб сервера достаточно запустить в каталоге с репозиторием:

```
python3 -m http.server
```

По умолчанию такой сервер запускается на порте 8000, эту цифру необходимо добавить к адресу через двоеточие.

```
baseurl=http://192.168.1.100:8000
```

Настройка HTTP-клиента

В файлах `/etc/yum.repos.d/*` пропишите свой сервер в строки `baseurl`. Пример:

было:

```
baseurl=http://mirror.rosa.ru/rosa/rosa$releasever/repository/x86_64/main/release/,http://mirror.yandex.ru/rosa/rosa$releasever/repository/x86_64/main/release/
```

стало:

```
baseurl=http://192.168.1.100/repo/x86_64/rosa$releasever/repository/x86_64/main/release/
```

(где 192.168.1.100 — адрес сервера)

Через запятую можно указать резервный репозиторий, который будет использоваться в случае недоступности первого.

Проверьте работу репозитория:

```
sudo dnf makecache
```

Доступ к репозиторию по NFS

Настройка NFS-сервера

Установите пакет утилит для NFS:

```
sudo dnf install nfs-utils
```

Откройте на редактирование файл с настройкой NFS-сервера:

```
sudo nano /etc/exports
```

Ниже приведен пример, в котором разрешается доступ только на чтение с IP-адресов 192.168.1.0-192.168.1.255 (задана маска подсети).

```
/srv 192.168.1.1/255.255.255.0(ro,insecure,nohide,all_squash,anonuid=1000,anongid=1000,no_subtree_check)
```

Запустите и добавьте в автозапуск службу NFS-сервера:

```
sudo systemctl enable --now nfs-server
```

Настройка NFS-клиента

Создайте каталог для монтирования NFS:

```
sudo mkdir -p /mnt/nfs
```

Откройте на редактирование файл /etc/fstab:

```
sudo nano /etc/fstab
```

Пропишите строку монтирования NFS-раздела:

```
192.168.1.100:/srv /mnt/nfs nfs4 auto,_netdev,bg,nofail
```

(где вместо 192.168.1.100 укажите адрес сервера).

Выполните монтирование, что также проверит корректность записи в /etc/fstab:

```
sudo mount /mnt/nfs
```

В файлах /etc/yum.repos.d/* пропишите свой сервер в строки baseurl. Пример:

было:

```
baseurl=http://mirror.rosa.ru/rosa/rosa$releasever/repository/x86_64/main/release/,http://mirror.yandex.ru/rosa/rosa$releasever/repository/x86_64/main/release/
```

стало:

```
baseurl=/mnt/nfs/srv/repo/x86_64/rosa$releasever/repository/x86_64/main/release/
```

Через запятую можно указать резервный репозиторий, который будет использоваться в случае недоступности первого.

Проверьте работу репозитория:

```
sudo dnf makecache
```

Источник — http://wiki.rosa.ru/index.php?title=Свой_репозиторий_РОСА&oldid=619

