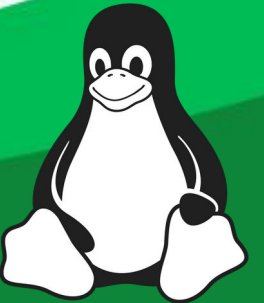


# Администрирование



# Понятие пользователя

---

Linux — система многопользовательская, а потому пользователь — ключевое понятие для организации всей системы доступа в Linux

В ОС Linux существует три типа пользователей:

Root

Системные пользователи

Обычные пользовател

# Свойства пользователя

---

Имя пользователя -в рамках системы имя должно быть уникальным

Идентификационный номер пользователя(UID) — уникальный идентификатором

Идентификационный номер группы (GID)

Пароль (password) - пароль пользователя в зашифрованном виде.

Полное имя (full name) - полное имя пользователя

Домашний каталог (home directory) - каталог, в который попадает пользователь после входа в систему

Начальная оболочка (login shell) - командная оболочка, запускается при входе в систему

# Создание пользователей

---

Наиболее типичный способ добавить пользователя — ввести команду `adduser`, принимающую в качестве аргумента имя нового пользователя, которого вы хотите создать

- `system`: добавить системного пользователя
- `home DIR`: использовать `DIR` в качестве домашнего каталога
- `shell SHELL`: используйте `SHELL` вместо значения по умолчанию.
- `ingroup GROUP`: установить первичную группу пользователя на `GROUP`
- `add_extra_groups`: добавить нового пользователя в дополнительную группу

# Изменение существующего пользователя

---

`passwd` — позволяет обычным пользователям изменять свой пароль

`chsh` (CHange Shell — сменить оболочку) — изменяет оболочку входа пользователя

`chage` (CHange AGE — сменить срок) — позволяет администратору изменять настройки

Отключение учетной записи: Вам может потребоваться отключить учетную запись. Это достигается с помощью команды `passwd -l пользователь` (`lock` — «заблокировать»).

Повторное включение

учетной записи выполняется аналогичным образом благодаря параметру `-u`

# Управление Unix-группами

---

Команды `addgroup` и `delgroup` добавляют или удаляют группу соответственно.

Команда `groupmod` изменяет информацию группы (ее `gid` или идентификатор).

Команда `grpasswd` группа изменяет пароль для группы, а команда `grpasswd -r` группа удаляет ее.

# Управление пакетами

Установка программного обеспечения в Linux осуществляется двумя основными

Способами:

- с помощью пакетов;
- из исходных кодов.

Два популярных типов пакетов

- RPM-пакеты (<http://rpmfind.net> и <http://rpm.pbone.net>;) )
- DEB-пакеты (<http://www.debian.org/distrib/packages> и <http://packages.ubuntu.com/>.)

# Установка из исходников

---

Для установки такой программы нужно распаковать архив с ее исходными кодами (желательно, в каталог `/usr/src`), затем перейти в каталог, содержащий файлы распакованного архива исходных кодов, и поочередно выполнить следующие команды:

```
./configure
```

```
make
```

```
make install
```



# Репозитории пакетов

---

Репозиторий — это хранилище пакетов. Используются Для централизованного управления обновлением пакетов. С помощью них можно быстро и удобно отслеживать обновления тех или иных пакетов. Причем это делает в автоматическом режиме сам менеджер пакетов, а вам остается лишь указать, какие обновления нужно загружать, а какие — нет

# Программы для управления пакетами

---

Rpm - Red Hat-совместимые дистрибутивы

Dpkg - на Debian (Ubuntu, Kubuntu и др.)

Apt -Debian, Ubuntu (и ее клоны), ALT Linux и др.

Yum - RedHat, CentOS, Fedora, Oracle Linux

Pacman - Arch-based

# Программа dpkg

Формат:

dpkg [ключи] действие, пример установки `sudo dpkg -i /<путь>/package.deb`

Последовательность действий при установке

1. Из пакета извлекаются управляющие файлы
2. если требуется, то обновление пакета выполняется автоматически
3. Выполняется сценарий `preinst`
4. Из пакета распаковываются остальные файлы
5. Если была установлена старая версия пакета, то из него выполняется сценарий `postrm`
6. Выполняется настройка пакета (распаковываются новые конфигурационные файлы)

Удалить пакет тоже просто:

`sudo dpkg -r <package>`

# Параметры dpkg

- I [образец] Выводит все установленные пакеты, имена которых соответствуют образцу
- L <имя пакета> Выводит имена файлов из указанного пакета (пакет должен быть установлен)
- p <имя пакета> Выводит информацию об установленном пакете
- s <имя пакета> Выводит информацию о статусе пакета
- unpack <имя пакета.deb> Распаковывает, но не устанавливает пакет

Для того чтобы узнать какому пакету принадлежит файл, dpkg имеет опцию -S, выведено будет имя пакета и адрес файла, например:

```
dpkg -S /usr/bin/cloud
```

# Программа apt-get

Программа apt-get используется не только в Debian/Ubuntu, но и в других дистрибутивах. В современных версиях вводить команду apt, а не apt-get

Установка пакета: `sudo apt-get install package`

Программа apt-get просматривает файл `/etc/apt/sources.list` — в этом файле перечислены источники (репозитории) DEB-пакетов, в качестве которых может выступать как компакт-диск, содержащий пакеты, так и сервер в Интернете

Общий формат вызова этой программы следующий:

`apt-get [опции] команды [пакет]`

# Программа rpm

Программа rpm — полноценный текстовый менеджер пакетов, позволяющий устанавливать, удалять пакеты, просматривать информацию об уже установленных и новых пакетах, обновлять пакеты

Чтобы установить пакет с помощью rpm, выполните команду:

```
# rpm -ihv <имя_пакета>
```

Удалить пакет так же просто:

```
# rpm -e <имя_пакета>
```

Для обновления пакета служит команда:

```
# rpm -U <имя_пакета>
```

# Программа rpm

Просмотреть, установлен ли тот или иной пакет, можно с помощью команды:

```
# rpm -qa | grep <имя_пакета>
```

Если вы хотите просмотреть информацию о пакете, то введите команду:

```
# rpm -qi <имя_пакета>
```

Просмотреть список файлов, входящих в состав пакета, можно командой:

```
# rpm -ql <имя_пакета>
```

Наконец, вывести все пакеты можно командой:

```
$ rpm -qa | grep more
```

Какому пакету принадлежит файл:

```
rpm -qf /bin/l
```

# Программа yum

---

`yum install пакет` Установить пакет из репозитория

`yum remove пакет` Удалить пакет

`yum update` Проверить наличие обновлений всех пакетов.

`yum update пакет` Проверить обновления конкретного пакета

`yum check-update` Только проверить наличие обновлений



# Программа yum

---

`yum check-update пакет` Проверить наличие обновлений конкретного пакета

`yum info пакет` Вывести информацию о пакете

`yum list` Вывести список пакетов установленных, и доступных для установки `yum list <...>`

Вывести список всех пакетов, соответствующих шаблону

`yum search строка` Найти все пакеты, в описании указанная строка

`yum groupinstall "группа"` Установить все пакеты из указанной группы

`yum grouplist` Вывести список групп пакетов

# Системные службы

---

Раньше, для управления службами использовались скрипты, каждая служба запускалась в фоновом режиме одна за другой. Использовалась обертка `service`

Запустить службу:

```
sudo service ufw start
```

Остановить службу:

```
sudo service ufw stop
```

Перезапустить службу:

```
sudo service ufw restart
```

Проверить статус службы:

```
sudo service ufw status
```

# Системные службы. Systemd

---

Но потом на смену этому методу пришла новая модель и система инициализации systemd.

Теперь появилась возможность параллельной инициализации, а также зависимостей между службами. Таким образом, теперь можно определить сложное дерево порядка запуска служб.

В Systemd есть специальный инструмент для управления службами в Linux — systemctl.

Синтаксис:

`systemctl` опции команда служба служба...

# Системные службы. Systemctl

---

Список всех запущенных юнитов: `# systemctl`

Информация о статусе юнита: `# systemctl status nginx`

Релоад конфигурации демона `# systemctl reload nginx`

Релоад конфигурации systemd (нужно при создании/редактировании юнит файлов) `# systemctl daemon-reload`

Запустить/остановить юнит `# systemctl [start | stop] nginx` Включить/выключить автозапуск юнита при загрузке системы: `# systemctl [enable | disable] nginx`

Список юнитов которые не запустились во время последней загрузки системы: `# systemctl --failed`

# SSH

---

SSH - (Secure Shell) - это протокол удаленного управления компьютером с ОС Linux

Синтаксис команды выглядит следующим образом: `$ ssh [опции] имя пользователя@сервер [команда]`

`f` - перевести ssh в фоновый режим;

`g` - разрешить удаленным машинам обращаться к локальным портам;

`I` - имя пользователя в системе;

`n` - перенаправить стандартный вывод в `/dev/null`;

`p` - порт ssh на удаленной машине;

`q` - не показывать сообщения об ошибках;

`v` - режим отладки;

`C` - включить сжатие.

# SSH

---

По умолчанию ssh работает на порту 22

Некоторые готовые рецепты по использованию ssh

Чтобы просто подключиться к серверу по SSH используйте такую команду:

```
ssh user@host
```

Чтобы выполнить команду на удаленном хосте: `ssh user@host ls`

Локальный скрипт выполнить на удаленной машину:

```
ssh user@host 'bash -s' < script.sh
```

# SSH. Аутентификация без пароля

---

Использование ssh пароля для входа на сервер не только неудобно но и небезопасно, потому что этот пароль в любой момент может быть подобран. Самый надежный и часто используемый способ аутентификации - с помощью пары ключей RSA. Секретный ключ хранится на компьютере, а публичный используется на сервере для удостоверения пользователя.

# SSH. Передача файлов

---

Можно копировать файлы по ssh. Для этого используется утилита scp. Просто укажите файл, который нужно передать, удаленный сервер и папку на сервере, вот:

```
$ scp /адрес/локального/файла пользователь@хост:адрес/папки
```

Например: scp ~/test.txt user@host:documents

Если вам нужно наоборот скопировать файлы с удаленного хоста на локальный, то ваш хост и удаленный меняются местами

```
scp root@191.162.0.2:/writing/articles/SCP.zip ~/dir
```

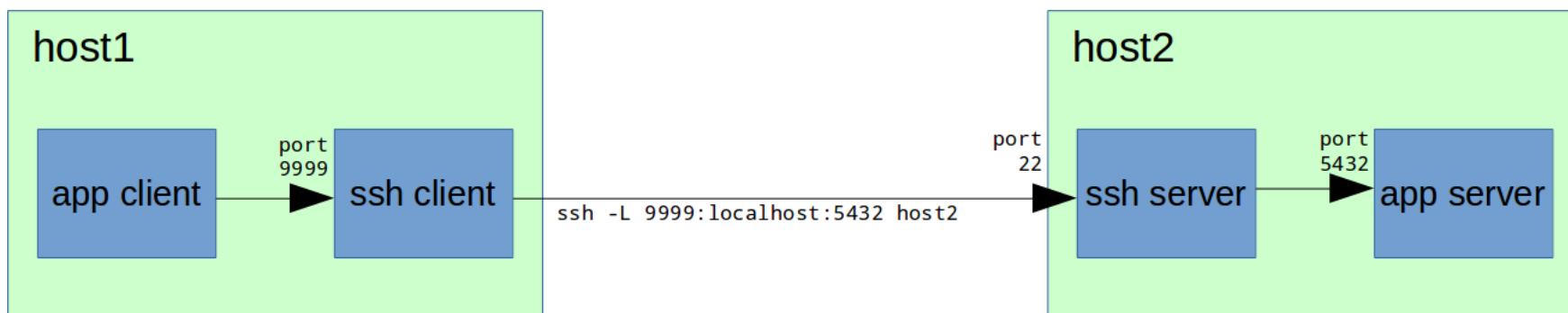


# Магия SSH. Local TCP forwarding

Имеем удаленный сервер «host2» с неким приложением Требуется подключиться с нашего рабочего места «host1» клиентским приложением к серверу PostgreSQL на «host2».

Для этого на «host1» в консоли набираем:

```
host1# ssh -L 9999:localhost:5432 host2
```

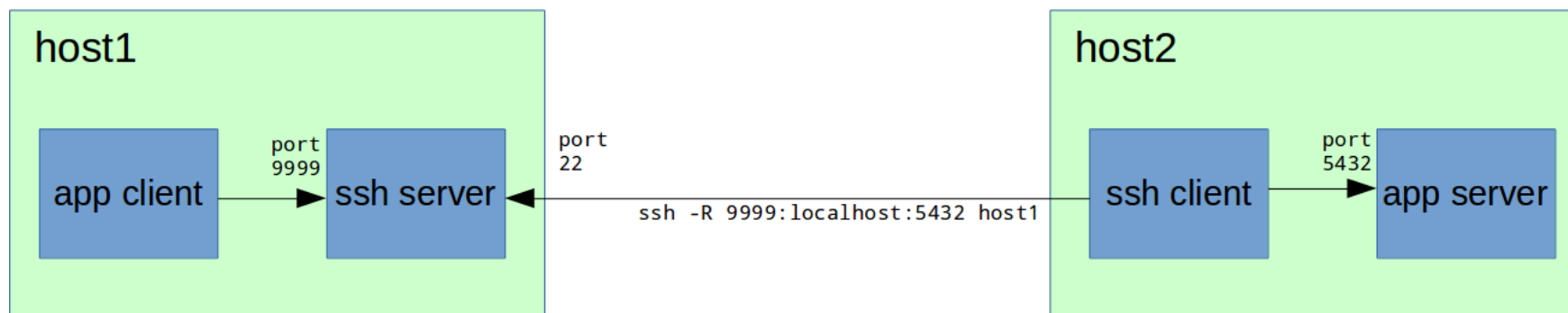


# Магия SSH. Remote TCP forwarding

Теперь нужно устанавливать ssh-соединение в обратном направлении — от «host2» к «host1».

Т.е. наша административная рабочая станция будет SSH-сервером и будет доступна по SSH с «host2», а на «host2» нужно будет выполнить подключение SSH-клиентом:

```
ssh -R 9999:localhost:5432 host1
```



**Спасибо за внимание!**

