

# Серверная инфраструктура

@pvavilin

3 сентября 2022 г.

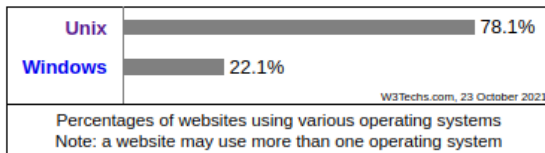
# Outline

# Почему Linux?

W3Techs

How to read the diagram:

Unix is used by 78.1% of all the websites whose operating system we know.



# Дистрибутивы Linux

их очень много!



# Какой же ставить на сервер?!

## Debian-based

- deb-пакеты для установки программ
- aptitude/apt — пакетный менеджер

## RedHat-based

- rpm-пакеты для установки программ
- yum — пакетный менеджер

# Debian-based

## Debian

Основная идея: он должен быть очень стабильный. В связи с этим версии пакетов будут довольно старые.

## Ubuntu

Задуман как user-friendly дистрибутив, но кто-то даже ставит его на продакшен-сервера. Не так заморочены за стабильность.

# RedHat-based

## RedHat

Коммерческий дистрибутив. Много дополнительных плюшек для стабильности и безопасности за платную подписку.

## CentOS

Бесплатная версия RedHat OS. Стремятся быть стабильными, в этом похожи на Debian.

## Fedora

User-friendly версия RedHat для рабочих станций.

# Другие

## Gentoo

Дистрибутив, где все (почти все) программы компилируются из исходных кодов с целью полной оптимизации под тот процессор, на котором программы будут работать.

## Arch Linux

Дистрибутив для опытных пользователей. Стремятся устанавливать только самые последние версии программ и библиотек.



# Полезные команды

## Обновление системы

- deb: apt-get update && apt-get upgrade
- rpm: yum update

## Поиск пакета

- deb: apt-cache search <request>
- rpm: yum search <request>

## Установка пакета

- deb: apt-get install <package>
- rpm: yum install <package>

# initd, systemd

## статья про systemd

|          |   |
|----------|---|
| BIOS     | Базовая система ввода/вывода<br>Загружает MBR                   |
| MBR      | Начальная загрузочная запись<br>Загружает GRUB                  |
| GRUB     | GRand Unified Bootloader<br>Загружает ядро                      |
| Kernel   | Ядро<br>Загружает /sbin/init                                    |
| Init     | Init<br>Загружает программы<br>уровней выполнения               |
| Runlevel | Программы уровней выполнения<br>загружаются из /etc/rc.d/rc*.d/ |

# BSD

FreeBSD/OpenBSD/NetBSD



# FreeBSD



## Доп. информация по Linux

- Stepik: Введение в Linux
- Брайан У. Керниган, Роб Пайк «Unix. Программное окружение»

# Работа с «удалённым» сервером. SSH

```
ssh -p 22 resu@host  
scp -P 22 resu@host:/path/to/file ~/  
scp -P 2220 \  
~/file resu@host:/path/to/file
```

# SSH Ключи

- Сгенерировать ключи `~/.ssh/id_rsa` и `~/.ssh/id_rsa.pub` для своего хоста (пользователя)  
`ssh-keygen`
- Отправить свой публичный ключ на удалённый сервер (добавит в `~/.ssh/authorized_keys`)  
`ssh-copy-id user@host`

**Никогда не публикуйте свой приватный ключ!**

`~/.ssh/id_rsa`

# Добавление алиасов для SSH-хостов

```
Host foo
  Hostname 192.168.250.23
  User root
Host bar
  Hostname 192.168.250.34
  User username
  Port 2221
```

В файл `~/.ssh/config` можно прописать алиасы для часто используемых хостов и сильно облегчить себе жизнь. Пароль прописать нельзя.

```
ssh bar
```

# Python + SSH

## Paramiko

```
from paramiko.client import SSHClient
client = SSHClient()
client.load_system_host_keys()
client.connect(
    "ssh.pvavilin.ru", username="root"
)
stdin, stdout, stderr = (
    client.exec_command("ls -l")
)
print(stdout.read().decode('utf-8'))
```



# Ansible

## Официальная страница

```
---  
- name: My task  
  hosts: all  
  tasks:  
    - name: Leaving a mark  
      command: \  
        "touch /tmp/ansible_was_here"
```

Запускаем список задач:

```
ansible-playbook mytask.yaml
```

# Salt

## Официальная страница

В отличие от Ansible, все сервера-клиенты слушают мастер-сервер и ждут от него указаний. Это позволяет, например, ограничить на клиентах набор допустимых команд.

# Rsync

Позволяет ускорить передачу данных с одного сервера на другой за счёт подсчёта контрольных сумм уже существующих данных. То есть, если на сервере есть данные за вчерашний день, то мы можем "докачать" блоки данных которые поменялись за сегодня.

```
rsync -av source destination
```

Можно использовать и на локальной машине.

## screen / tmux

При работе с удалённым сервером часто нужно запустить несколько вкладок терминала. screen/tmux позволяют открыть несколько сессий командной строки в рамках одного подключения (можно и на локальной машине). Плюс если у вас отвалится SSH-подключение, то при этом ваш сеанс работы с screen/tmux не будет потерян в отличие от работы в Bash напрямую. Благодаря последней возможности можно запускать фоновые сервисы без написания специальных скриптов.

screen старый и вездесущий.

Tmux более user-friendly.

# Проверка состояния сервера

- `uptime` покажет сколько времени сервер живёт с момента последней перезагрузки. Покажет загрузку процессоров задачами (Load Average)
- `w` покажет запущенные сеансы пользователей
- `top` покажет статистику по процессам
- `htop` user-friendly *top* (install)
- `iotop` покажет загрузку ввода/вывода (install)
- `lsof` показывает информацию об открытых «файлах»
- `df -h` покажет состояние примонтированных файловых систем
- `free -m` покажет состояние оперативной памяти и файла подкачки
- `date` покажет текущее время на сервере

## Другие оболочки командной строки

*Bash* — это лишь одна из множества оболочек командной строки, но давно стал стандартом де-факто.

Официальным стандартом является очень простая оболочка *sh*.

Кроме *Bash* пользуется популярностью оболочка *zsh* (в MacOS стандартная оболочка).

*Bash* и *zsh* совместимы с *sh* — это значит что базовый синтаксис у них соответствует синтаксису *sh*.

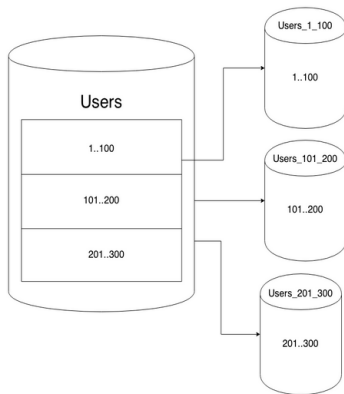
- PostgreSQL и MySQL (MariaDB) обе профессиональные RDBMS. PostgreSQL может быть чуть сложнее в настройке из-за бóльшего набора возможностей.
- SQLite — подходит только для очень маленьких задач или для изучения стандарта SQL.
- NoSQL огромное множество. Выбор делается в соответствии с теоремой CAP

# Репликация





# Шардирование



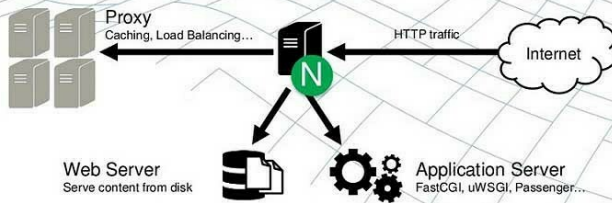
# Nginx

В интернете очень много статики!



# Nginx

## What is NGINX?



# Cron

```

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin

# Example of job definition:
# .----- minute (0-59)
# | .----- hour (0-23)
# | | .----- day of month (1-31)
# | | | .----- month (1-12) OR jan,feb,mar,apr
# | | | | .---- day of week (0-6) (Sunday=0 or 7)
# | | | | | OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# m h dom mon dow user  command
17 * * * * root    <command>
25 6 * * *   root    <command>
47 6 * * 7   root    <command>
52 6 1 * *   root    <command>

```

man 5 crontab

crontab -e

# Bacula

- Полный бэкап (например, раз в месяц)
- Инкрементальный бэкап (например, каждый день)

# Logrotate

Документация

Подчищает старые лог-файлы.

## Дополнительная литература

Если вам стал интересен какой-то из пунктов, то пишите запрос на доп. литературу в #XX\_offtopic или #XX\_library 😊

# Безопасность

- Генерируйте пароли при помощи *pwgen* или *makepasswd*
- Проверяйте пароли на стойкость с помощью *cracklib-check*
- Никогда не работайте под **root**-ом, используйте *sudo*
- Не пишите файлы в *shared* директории типа */tmp*
- Лучше использовать нестандартный порт для SSH
- Процессы лучше заворачивать в *docker*
- Безопасность в Linux очень обширная тема. Специалисты по Linux-безопасности крайне востребованы.



# Вопросы-ответы



designed by freepik.com