

1C, postgres, linux, proxmox

- [Инструкции по установке PostgreSQL 13 для 1c](#)
- [ftp на linux PVE](#)
- [Репозитории РФ](#)
- [ККТ занят порт Атол](#)
- [Советую про systemd](#)
- [Инфо о интерфейсе / скорость / ДНС](#)
- [Создание iso в linux](#)
- [Python HTTP server как служба](#)
- [fail2ban защита от перебора](#)
- [Борьба за iops ы](#)
- [scrcpy](#)
- [nopasswd](#)

Инструкции по установке PostgreSQL 13 для 1c

Вы получили это письмо, поскольку запрашивали инструкции по установке PostgreSQL для 1c на сайте 1c.postgres.ru/.

Используйте инструкции для установки PostgreSQL для 1c. Обратите внимание, что команды должны выполняться от пользователя с правами суперпользователя.

```
wget https://repo.postgrespro.ru/1c-13/keys/pgpro-repo-add.sh
sh pgpro-repo-add.sh
```

Если наш продукт единственный Postgres на вашей машине и вы хотите сразу получить готовую к употреблению базу:

```
apt-get install postgrespro-1c-13
```

Если у вас уже установлен другой Postgres и вы хотите чтобы он продолжал работать параллельно (в том числе и для апгрейда с более старой major-версии):

```
apt-get install postgrespro-1c-13-contrib
/opt/pgpro/1c-13/bin/pg-setup initdb
/opt/pgpro/1c-13/bin/pg-setup service enable
/opt/pgpro/1c-13/bin/pg-setup service start
```

ftp на linux PVE

Отличный выбор! Развертывание FTP-сервера в **LXC-контейнере** на Proxmox VE (PVE) — это легковесное и безопасное решение.

1. Создание LXC-контейнера

1. В веб-интерфейсе PVE

- Перейдите в `Create CT` (контейнер).
- Выберите **шаблон** (например, `Debian 12` или `Ubuntu 22.04`).
- Укажите **ID контейнера**, **пароль root** и **хранилище**.
- В `Network` задайте статический IP (например, `192.168.1.100`) или DHCP.

2. Через CLI (альтернатива)

```
```bash
pct create 100 local:vztmpl/debian-12-standard_12.0-1_amd64.tar.zst \
--storage local-lvm --rootfs 8 \
--net0 name=eth0,bridge=vbr0,ip=192.168.1.100/24,gw=192.168.1.1
pct start 100
```
```

2. Установка FTP-сервера (vsftpd)

1. Подключитесь к контейнеру

```
```bash
pct enter 100 # или ssh root@192.168.1.100
```
```

2. Обновите систему и установите vsftpd

```
```bash
apt update && apt upgrade -y
apt install vsftpd -y
```
```

3. Настройка vsftpd

1. Отредактируйте конфиг:

```
```bash
nano /etc/vsftpd.conf
```
```

2. Минимальная рабочая конфигурация:

```
```ini
listen=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
```
```

```
chroot_local_user=YES
allow_writeable_chroot=YES
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=40100
local_umask=022
...

```

3. ****Перезапустите vsftpd:****

```
```bash
systemctl restart vsftpd
systemctl enable vsftpd
...

```

### ### **\*\*4. Создание FTP-пользователя\*\***

#### 1. **\*\*Добавьте пользователя (например, `ftpuser`):\*\***

```
```bash
useradd -m -d /home/ftpuser -s /bin/bash ftpuser
passwd ftpuser
...

```

2. ****Если нужен доступ к конкретной папке:****

```
```bash
mkdir /mnt/storage
chown ftpuser:ftpuser /mnt/storage
usermod -d /mnt/storage ftpuser
...

```

### ### **\*\*5. Открытие портов в PVE (если контейнер за NAT)\*\***

Если у контейнера **\*\*NAT\*\*** (через `vmbro`), пробейте порты на хосте PVE:

```
```bash
iptables -t nat -A PREROUTING -p tcp --dport 21 -j DNAT --to 192.168.1.100:21
iptables -t nat -A PREROUTING -p tcp --dport 40000:40100 -j DNAT --to 192.168.1.100:40000-40100
...

```

Для сохранения правил:

```
```bash
apt install iptables-persistent -y
netfilter-persistent save
...

```

### ### **\*\*6. Проверка подключения\*\***

- Используйте **\*\*FileZilla\*\*** или `ftp` в терминале:

```
...
ftp 192.168.1.100
...

```

Логин: `ftpuser` и ваш пароль.

---

### ### **\*\*Дополнительные настройки\*\***

- **\*\*SFTP (повышенная безопасность):\*\*** Используйте OpenSSH (`sftp`).
- **\*\*Резервное копирование:\*\*** Настройте `rsync` через FTP.
- **\*\*Логирование:\*\*** Включите `xferlog\_enable=YES` в `/etc/vsftpd.conf`.

---

### ### **\*\*Итог\*\***

Вы настроили **\*\*FTP-сервер в LXC-контейнере\*\*** на PVE. Это изолирует сервис от хоста и упрощает управление.

Если нужен **\*\*безопасный доступ\*\***, рассмотрите **\*\*SFTP/SSH\*\*** или **\*\*WebDAV\*\*** (например, через Nextcloud).

# Репозитории РФ

## Ubuntu 24.04 LTS (Noble Numbat)

```
deb https://mirror.selectel.ru/ubuntu/ noble main restricted universe multiverse
deb https://mirror.selectel.ru/ubuntu/ noble-updates main restricted universe multiverse
deb https://mirror.selectel.ru/ubuntu/ noble-security main restricted universe multiverse
deb https://mirror.selectel.ru/ubuntu/ noble-backports main restricted universe multiverse
```

```
deb http://mirror.yandex.ru/ubuntu/ noble main restricted universe multiverse
deb http://mirror.yandex.ru/ubuntu/ noble-updates main restricted universe multiverse
deb http://mirror.yandex.ru/ubuntu/ noble-security main restricted universe multiverse
deb http://mirror.yandex.ru/ubuntu/ noble-backports main restricted universe multiverse
```

# ККТ занят порт Атол

<https://kkmspb.ru/kassovye-apparaty/knowledge-base/Atol/the-port-is-busy.php>

# Советую про systemd

<https://habr.com/ru/articles/942760/>

# Инфо о интерфейсе / скорость / ДНС

```
networkctl status enp4s0
```

```
ethtool enp4s0
```

```
cat /sys/class/net/enp4s0/speed // просто скорость
```

# Создание iso в linux

bash

```
Установка (если не установлен)
sudo apt install genisoimage # Debian/Ubuntu
sudo yum install genisoimage # CentOS/RHEL

Создание ISO образа
genisoimage -o my_files.iso -J -R -V "MY_FILES" file1.txt file2.doc directory/
```

Опции:

- `-o` - имя выходного файла
- `-J` - Joliet расширения для Windows совместимости
- `-R` - Rock Ridge расширения для Unix прав
- `-V` - метка тома

# Python HTTP server как служба

## 1. Создаем службу

bash

```
sudo nano /etc/systemd/system/python-fileserver.service
```

## 2. Конфиг службы

ini

```
[Unit]
Description=Python HTTP File Server
After=network.target

[Service]
Type=simple
User=your_username
Group=your_username
WorkingDirectory=/home/your_username/filehost
ExecStart=/usr/bin/python3 -m http.server 8000 --bind 0.0.0.0
Restart=always
RestartSec=5
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target
```

## 3. Заменяем в конфиге:

- `your_username` - ваш пользователь на VDS
- `/home/your_username/filehost` - путь к папке с файлами
- `8000` - нужный порт

## 4. Активируем службу

bash

```
Перечитываем конфиги
sudo systemctl daemon-reload

Включаем автозапуск
sudo systemctl enable python-fileserver

Запускаем службу
sudo systemctl start python-fileserver

Проверяем статус
sudo systemctl status python-fileserver
```

# fail2ban защита от перебора

## Использование fail2ban

```
bash
```

```
Установка
```

```
sudo apt update && sudo apt install fail2ban
```

```
Настройка для SSH
```

```
sudo nano /etc/fail2ban/jail.local
```

Добавьте конфигурацию:

```
[sshd]
```

```
enabled = true
```

```
port = ssh
```

```
filter = sshd
```

```
logpath = /var/log/auth.log
```

```
maxretry = 3
```

```
bantime = 3600
```

```
findtime = 600
```

# Борьба за iops ы

Тест производительности дисковой подсистемы в PROXMOX.

Изначально: zfs пул на nvme, ashift 12, включен кэш в настройках VM write back, ФС внутри VM ext4.

Подобраны параметры для fio - смешанная нагрузка 1C

```
fio --name=1c_mixed --filename=/tmp/mixed_workload --size=8G --rw=randrw --rwmixread=60 --bs=8k --ioengine=libaio --iodepth=16 --numjobs=8 --runtime=400 --time_based --group_reporting > /home/a30t/vk1c-ext4-ashift12-wcache.txt
```

## vk1c-ext4-ashift12-wcache

Run status group 0 (all jobs):

READ: bw=675MiB/s (708MB/s), 675MiB/s-675MiB/s (708MB/s-708MB/s), io=264GiB (283GB), run=400015-400015msec

WRITE: bw=450MiB/s (472MB/s), 450MiB/s-450MiB/s (472MB/s-472MB/s), io=176GiB (189GB), run=400015-400015msec

Disk stats (read/write):

dm-0: ios=4137380/3208203, merge=0/0, ticks=492888/2446976, in\_queue=2939864, util=76.95%, aggrios=4137380/3206699, aggrmerge=0/1507, aggrticks=514372/2050053, aggrin\_queue=2721125, aggrutil=80.08%

sda: ios=4137380/3206699, merge=0/1507, ticks=514372/2050053, in\_queue=2721125, util=80.08%

## vk1c-ext4-ashift12

Run status group 0 (all jobs):

READ: bw=516MiB/s (541MB/s), 516MiB/s-516MiB/s (541MB/s-541MB/s), io=202GiB (217GB), run=400001-400001msec

WRITE: bw=344MiB/s (361MB/s), 344MiB/s-344MiB/s (361MB/s-361MB/s), io=134GiB (144GB), run=400001-400001msec

Disk stats (read/write):

dm-0: ios=3574961/3125992, merge=0/0, ticks=742172/3416224, in\_queue=4158396, util=89.33%, aggrios=3575186/3102889, aggrmerge=222/25408, aggrticks=768781/3000911, aggrin\_queue=3856302, aggrutil=90.87%  
sda: ios=3575186/3102889, merge=222/25408, ticks=768781/3000911, in\_queue=3856302, util=90.87%

## vk1c-ext4-ashift13

Run status group 0 (all jobs):

READ: bw=511MiB/s (536MB/s), 511MiB/s-511MiB/s (536MB/s-536MB/s), io=200GiB (214GB), run=400001-400001msec  
WRITE: bw=341MiB/s (357MB/s), 341MiB/s-341MiB/s (357MB/s-357MB/s), io=133GiB (143GB), run=400001-400001msec

Disk stats (read/write):

dm-0: ios=3469061/3526092, merge=0/0, ticks=696228/3092284, in\_queue=3788512, util=87.32%, aggrios=3470006/3506293, aggrmerge=28/25951, aggrticks=716387/2628436, aggrin\_queue=3431389, aggrutil=87.60%  
sda: ios=3470006/3506293, merge=28/25951, ticks=716387/2628436, in\_queue=3431389, util=87.60%

# scrcpy

<https://github.com/Genymobile/scrcpy/blob/master/doc/linux.md>

First, you need to install the required packages:

```
for Debian/Ubuntu
sudo apt install ffmpeg libSDL2-2.0-0 adb wget \
 gcc git pkg-config meson ninja-build libSDL2-dev \
 libavcodec-dev libavdevice-dev libavformat-dev libavutil-dev \
 libswresample-dev libusb-1.0-0 libusb-1.0-0-dev
```

Then clone the repo and execute the installation script ([source](#)):

```
git clone https://github.com/Genymobile/scrcpy
cd scrcpy
./install_release.sh
```

When a new release is out, update the repo and reinstall:

```
git pull
./install_release.sh
```

To uninstall:

```
sudo ninja -Cbuild-auto uninstall
```

# nopasswd

sudo visudo

```
user ALL=(ALL) NOPASSWD: ALL
```