

Оглавление

1	Задачи	2
1.1	Постановка задач	2
1.2	Дополнительные задачи	2
2	Выполнение задач – на VM	3
2.1	Задача №1 – Развёртывание VM Oracle Linux 8	3
2.2	Задача №2 – Настройка дисковой подсистемы с помощью менеджера томов LVM	4
2.2.1	создать LVM-том, добавить и смонтировать дополнительный диск для PostgreSQL в /var/lib/pgsql;	4
2.2.2	Увеличить корневой раздел	7
2.2.3	Поработать со snapshot-ами томов (создать, внести изменения, откатить);	8
2.3	Задача №3 – Организация доступа	12
2.4	Задача №4 – Работа с пакетным менеджером DNF	13
2.5	Задача №5 – Инициализация и настройка PostgreSQL	14
2.6	Задача №6 – Развёртывание Демо-базы и настройка доступа	15
2.7	Задача №7 – Установка и работа в PgAdmin	16
3	Выполнение задач – Использование скриптов	17
3.1	Задача №1 – Развёртывание VM Oracle Linux 8	17
3.2	Задача №2 – Настройка дисковой подсистемы с помощью менеджера томов LVM	18
3.3	Задача №3 – Организация доступа	19
3.4	Задача №4 – Работа с пакетным менеджером DNF	20
3.5	Задача №5 – Инициализация и настройка PostgreSQL	21
3.6	Задача №6 – Развёртывание Демо-базы и настройка доступа	22
3.7	Задача №7 – Установка и работа в PgAdmin	23

1 Задачи

1.1 Постановка задач

1. Развернуть виртуальную машину с ОС Oracle Linux 8 любым удобным способом:
 - через портал «облака»;
 - вручную через консоль/CLI системы виртуализации;
 - автоматически с помощью Terraform/Ansible/Vagrant/API виртуализации;
2. Настройка дисковой подсистемы с помощью менеджера томов LVM:
 - создать LVM-тома, добавить и смонтировать дополнительный диск для PostgreSQL в `/var/lib/pgsql`;
 - увеличить корневой раздел;
 - поработать со snapshot-ами томов (создать, внести изменения, откатить);
3. Организовать доступ:
 - создание непривилегированных пользователей;
 - настройка `sudo` для выполнения привилегированных команд;
4. Работа с пакетным менеджером DNF:
 - Установка основных утилит из стандартных репозиториях Linux:
 - Редакторы и управление сессиями: `vim`, `nano`, `mc`, `screen`;
 - Сетевые клиенты и диагностика: `wget`, `curl`, `telnet`, `nmap-ncat`, `tcpdump`, `net-tools`, `bind-utils`;
 - Файловые системы и хранение: `autofs`, `nfs-utils`, `cloud-utils-growpart`, `lsfd`, `sysfsutils`, `sg3utils`;
 - Системный мониторинг и диагностика: `sysstat`;
 - Утилиты общего назначения: `pwgen`, `bc`, `unzip`, `glibc-langpack-ru`;
 - Разработка и контроль версий: `git`;
 - Установка модулей PostgreSQL версии 15;
5. Инициализация и настройка PostgreSQL:
 - создать базу данных PostgreSQL;
 - инициализировать кластер с включёнными контрольными суммами;
 - настроить удалённое подключение (правки `postgresql.conf`, `pg_hba.conf`);
6. Развёртывание Демо-базы и настройка доступа:
 - скачать Демо-базу, например, с:
<https://postgrespro.ru/education/courses/DBA1>
<https://postgrespro.ru/docs/postgrespro/16/demodb-bookings-installation>;
 - импортировать Демо-базу через SQL-дамп;
 - создать пользователя и выдать привилегии на созданные базы;
7. Установка и работа в PgAdmin:
 - установить PgAdmin Desktop;
 - настроить подключение к базе;
 - выполнить простые и сложные SQL-запросы (`WHERE`, `GROUP BY`, `ORDER BY`, `JOIN`);
 - проанализировать планы запросов через `EXPLAIN PLAN`;

1.2 Дополнительные задачи

1. Проекты вести в GitLab/GitHub - код, инструкции и т.п
2. Автоматизировать как можно больше шагов/этап с помощью скриптов `bash`, `Ansible`

2 Выполнение задач – на VM

2.1 Задача №1 – Развёртывание VM Oracle Linux 8

2.2 Задача №2 – Настройка дисковой подсистемы с помощью менеджера томов LVM

2.2.1 создать LVM-том, добавить и смонтировать дополнительный диск для PostgreSQL в /var/lib/pgsql;

Команды 1–5

```
# 1.1 инициализируем диск как Physical Volume
sudo pvcreate /dev/sda

# 1.2 добавляем его в существующую группу VG (например, ol_vbox)
sudo vgextend ol /dev/sda

# 1.3 создаём логический том 8 ГБ
sudo lvcreate -L 8G -n pglv ol

# 1.4 форматируем в XFS (или ext4)
sudo mkfs.xfs /dev/ol_vbox/pglv

# 1.5 монтируем и делаем постоянным
sudo mkdir -p /var/lib/pgsql
sudo echo '/dev/ol/pglv /var/lib/pgsql xfs defaults 0 2' > /etc/fstab
sudo mount -a
```

```
hin@vbox:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0   8G  0 disk
sdb                                  8:16     0  20G  0 disk
├─sdb1                               8:17     0   1M  0 part
├─sdb2                               8:18     0    1G  0 part /boot
└─sdb3                               8:19     0   19G  0 part
   ├─ol_vbox-root                    252:0     0   17G  0 lvm  /
   └─ol_vbox-swap                    252:1     0    2G  0 lvm  [SWAP]
sr0                                  11:0     1 1024M  0 rom
```

Рис. 1: Посмотрим список доступных дисков (lsblk)

1.1 Делаем новый диск PV

```
sudo pvcreate /dev/sda
```

```
hin@vbox:~$ sudo pvcreate /dev/sda
Physical volume "/dev/sda" successfully created.
```

Рис. 2: Результат pvcreate

1.2 Включаем его в VG – Volume Group

```
sudo vgextend ol_vbox /dev/sda
```

```
hin@vbox:~$ sudo vgextend ol_vbox /dev/sda
Volume group "ol_vbox" successfully extended
```

Рис. 3: Расширенная VG ol_vbox

1.3 Создаём логический том pglv на 5 ГБ

```
sudo lvcreate -L 5G -n pglv ol_vlox
```

```
hin@vbox:~$ sudo lvcreate -L 5G -n pglv ol_vlox
Logical volume "pglv" created.
hin@vbox:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0   8G  0 disk
└─ol_vbox-pglv                      252:2    0    5G  0 lvm
sdb                                  8:16     0  20G  0 disk
├─sdb1                              8:17     0    1M  0 part
├─sdb2                              8:18     0    1G  0 part /boot
├─sdb3                              8:19     0   19G  0 part
└─ol_vbox-root                     252:0    0   17G  0 lvm /
   └─ol_vbox-swap                   252:1    0    2G  0 lvm [SWAP]
sr0                                  11:0     1 1024M  0 rom
```

Рис. 4: Проверяем изменения после lvcreate

1.4 Форматируем в XFS

```
# Сначала посмотрим, какой тип у диска (Рис. 5)
lsblk -o NAME,FSTYPE,SIZE
# Затем отформатируем логический диск (Рис. 6)
sudo mkfs.xfs /dev/ol_vbox/pglv
```

```
mkfs.xfs /dev/ol_vbox/pglv
```

- mkfs.xfs — утилита для создания файловой системы XFS на заданном блочном устройстве.
- /dev/ol_vlox/pglv — LVM-том (логический том) с именем pglv, находящийся в группе томов ol_vlox.

При выполнении этой команды:

- Утилита сотрёт все данные на томе /dev/ol_vbox/pglv.
- Запишет метаданные XFS (супблок, журналы, таблицы аллокации и т.д.) в начало и в конец тома.
- Подготовит том так, чтобы ядро Linux могло смонтировать его как XFS-раздел.
- После окончания получен раздел, готовый к использованию под XFS.

```
hin@vbox:~$ lsblk -o NAME,FSTYPE,SIZE
NAME                                FSTYPE      SIZE
sda                                  LVM2_member  8G
└─ol_vbox-pglv                      5G
sdb                                  20G
├─sdb1                              1M
├─sdb2                              xfs          1G
├─sdb3                              LVM2_member  19G
└─ol_vbox-root                     xfs          17G
   └─ol_vbox-swap                   swap         2G
sr0                                  1024M
```

Рис. 5: Информация о диске перед форматированием

```

hin@vbox:~$ sudo mkfs.xfs /dev/ol_vbox/pglv
[sudo] пароль для hin:
meta-data=/dev/ol_vbox/pglv      isize=512    agcount=4, agsize=327680 blks
       =                       sectsz=512    attr=2, projid32bit=1
       =                       crc=1        finobt=1, sparse=1, rmapbt=1
       =                       reflink=1    bigtime=1 inobtcount=1 nrext64=1
data      =                       bsize=4096   blocks=1310720, imaxpct=25
       =                       sunit=0      swidth=0 blks
naming    =version 2             bsize=4096   ascii-ci=0, ftype=1, parent=0
log       =internal log         bsize=4096   blocks=16384, version=2
       =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                 extsz=4096   blocks=0, rtextents=0

```

Рис. 6: Форматирование тома в XFS (mkfs.xfs)

1.5 Монтируем и делаем постоянным

```

# создаём точку монтирования
sudo mkdir -p /var/lib/pgsql

# добавляем строку в /etc/fstab для автомонтирования
echo '/dev/ol/pglv /var/lib/pgsql xfs defaults 0 2' | sudo tee -a /etc/fstab

# перезагружаем таблицу юнитов
systemctl daemon-reload

# монтируем всё из fstab
sudo mount -a

```

```

hin@vbox:/dev$ echo 'dev/ol_vlox/pglv /var/lib/pgsql xfs defaults 0 2' | sudo tee -a /etc/fstab
dev/ol_vlox/pglv /var/lib/pgsql xfs defaults 0 2

```

Рис. 7: Успешное добавление записи и монтирование тома

```

hin@vbox:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0   8G  0 disk
└─ol_vbox-pglv                      252:2    0    5G  0 lvm  /var/lib/pgsql
sdb                                  8:16     0  20G  0 disk
├─sdb1                              8:17     0    1M  0 part
├─sdb2                              8:18     0    1G  0 part /boot
├─sdb3                              8:19     0   19G  0 part
└─ol_vbox-root                     252:0    0   17G  0 lvm  /
   └─ol_vbox-swap                   252:1    0    2G  0 lvm  [SWAP]
sr0                                  11:0     1 1024M  0 rom

```

Рис. 8: Результат: Создание LVM-тома под /var/lib/pgsql

2.2.2 Увеличить корневой раздел

2.1. Посмотрим на ситуацию до увеличения

```
# Посмотрим текущий размер LV
lsblk -f | grep root

# Посмотрим использование диска для корневой точки /
df -h /
```

```
hin@vbox:~$ lsblk -f | grep root
└─ol_vbox-root xfs                    528b8452-f5e5-48a1-a3f9-14f225ec6f47    12,5G    26% /
hin@vbox:~$ sudo df -h /
Файловая система    Размер  Использовано  Дст  Использовано%  Смонтировано в
/dev/mapper/ol_vbox-root  17G      4,5G        13G      27% /
```

Рис. 9: Статус до расширения

2.2. Посмотрим сколько есть свободного места

```
sudo vgs
# команда vgs выводит сводную таблицу по всему группам томов
# VFree – доступно, т.е. не распределено ни одному LV.
```

```
hin@vbox:~$ sudo vgs
[sudo] пароль для hin:
VG      #PV #LV #SN Attr   VSize  VFree
ol_vbox  2   3   0 wz--n- 26,99g <3,00g
```

Рис. 10: Видим, что доступно меньше 3g

2.3. Увеличим логический том root LV

```
sudo lvextend -L +2G -r /dev/ol_vbox/root
```

- `lvextend` – расширяет существующий LV
- `-L +2G` – добавляет +2 ГБ к текущему размеру тома
- `-r` – автоматически увеличивает файловую систему, находящуюся внутри LV
- `/dev/ol_vbox/root` – путь к корневому логическому тому

```
hin@vbox:~$ sudo lvextend -L +2G -r /dev/ol_vbox/root
[sudo] пароль для hin:
File system xfs found on ol_vbox/root mounted at /.
Size of logical volume ol_vbox/root changed from <17,00 GiB (4351 extents) to <19,00 GiB (4863 extents).
Extending file system xfs to <19,00 GiB (20396900352 bytes) on ol_vbox/root...
xfs_growfs /dev/ol_vbox/root
meta-data=/dev/mapper/ol_vbox-root isize=512    agcount=4, agsize=1113856 blks
=                               sectsz=512    attr=2, projid32bit=1
=                               crc=1        finobt=1, sparse=1, rmapbt=1
=                               reflink=1     bigtime=1 inobtcount=1 nrext64=1
=                               exchange=0
data      =                       bsize=4096    blocks=4455424, imaxpct=25
=                               sunit=0       swidth=0 blks
naming    =version 2              bsize=4096    ascii-ci=0, ftype=1, parent=0
log       =internal log          bsize=4096    blocks=16384, version=2
=                               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                  extsz=4096    blocks=0, rtextents=0
data blocks changed from 4455424 to 4979712
xfs_growfs done
Extended file system xfs on ol_vbox/root.
Logical volume ol_vbox/root successfully resized.
```

Рис. 11: Результат увеличения логического тома root

2.4. Посмотрим на ситуацию после увеличения

```
# Посмотрим текущий размер LV
lsblk -f | grep root

# Посмотрим использование диска для корневой точки /
df -h /
```

```
hin@vbox:~$ lsblk -f | grep root
└─ol_vbox-root xfs                    528b8452-f5e5-48a1-a3f9-14f225ec6f47    14,4G    24% /
└─ol_vbox-root xfs                    528b8452-f5e5-48a1-a3f9-14f225ec6f47    14,4G    24% /
hin@vbox:~$ df -h /
Файловая система    Размер  Использовано  Дрст  Использовано%  Смонтировано в
/dev/mapper/ol_vbox-root  19G      4,6G        15G      24% /
```

Рис. 12: Результат увеличения логического тома root

2.2.3 Поработать со snapshot-ами томов (создать, внести изменения, откатить);

Опр. Snapshot

Снапшот (snapshot) — это «моментальный снимок» состояния логического тома в момент его создания. LVM создаёт «копию-только-для-чтения» оригинального тома, но при этом не дублирует все данные, а хранит только изменения («копия-при-записи» или CoW).

В момент создания снапшота резервируется (например) 1 ГиБ CoW-области. Снапшот и исходный LV делят одни и те же блоки — место почти не расходуется.

В момент записи в какой-то блок исходного LV, происходят изменения. Старая версия изменяемого блока копируется в CoW область снапшота, а новая версия пишется в исходный LV.

Расход места. Допустим, мы изменяем 4MiB памяти на LV (как самая минимальная ячейка для XFS или Ext4). Тогда мы тратим 4MiB из CoW памяти.

Пока сумма изменений будет меньше объема CoW (≤ 1 ГиБ) — снапшот будет работать корректно. Если изменений будет больше, CoW область переполнится: снапшот перейдет в состояние Invalid (т.е. потерянные данные).

Снапшот желательно хранить в той же VG, что и основной том, чтобы LVM мог эффективно управлять пространством.

Устройство LVM-пирога

Диски/разделы → PV → VG → LV → Файловая система → каталог / приложение

Уровень	Что это	Главная роль
PV (Physical Volume)	Физический носитель: диск <code>/dev/sdb</code> , раздел <code>/dev/sda3</code> , loop-файл и т.д.	«Кирпичи» пула; содержит физические экстенды LVM.
VG (Volume Group)	Объединение одного или нескольких PV в общий «пул» (пример: <code>ol_vbox</code>).	Даёт единый резерв пространства; можно расширять, добавляя новые PV.
LV (Logical Volume)	Логический том, «вырезанный» из VG (<code>root</code> , <code>home</code> , <code>pglv</code>).	Ведёт себя как раздел: форматируется, монтируется; легко расширяется, снапшотится.

Таблица 1: Структура LVM: PV → VG → LV

Желаемое место для Snapshot'a

Размер snapshot должен перекрывать объем ожидаемых изменений за время жизни снапшота, т.к. если память CoW области переполнится, снимок помечается как invalid и откат становится невозможным.

Устройство LVM-пирога

Диски/разделы → PV → VG → LV → Файловая система → каталог / приложение

3.1. Проверяем свободное место в группе томов (VG) ol_vbox

```
sudo vgs
```

```
hin@vbox:~$ sudo vgs
[sudo] пароль для hin:
VG      #PV #LV #SN Attr   VSize  VFree
ol_vbox  2   3   0 wz--n- 26,99g 1020,00m
```

Рис. 13: Проверка свободного места до создания снимка

3.2. Создаём снимок корневого LV на 1 ГиБ

```
# Создание снимка
sudo lvcreate -L 1G -s -n rootsnap /dev/ol_vbox/root

# убеждаемся, что rootsnap создан
sudo lvs -o lv_name,lv_size,data_percent,origin
```

```
hin@vbox:~$ sudo lvcreate -L 1G -s -n rootsnap /dev/ol_vbox/root
Volume group "ol_vbox" has insufficient free space (255 extents): 256 required.
hin@vbox:~$ sudo lvcreate -L 0.8G -s -n rootsnap /dev/ol_vbox/root
Rounding up size to full physical extent 820,00 MiB
Logical volume "rootsnap" created.
```

Рис. 14: Создание снимка

3.3. Смотрим свободное место после создания снимка

```
sudo vgs
```

```
hin@vbox:~$ sudo vgs
VG      #PV #LV #SN Attr   VSize  VFree
ol_vbox  2   4   1 wz--n- 26,99g 200,00m
```

Рис. 15: Проверяем изменение свободного места после создания снимка

3.4. Монтируем снимок «только-чтение»

```
# Создаём точку монтирования
# - опция -p выдаст ошибку, если каталог уже существует
sudo mkdir -p /mnt/rootsnap

# Монтирует LVM-снимок rootsnap только для чтения (ro) по адресу /mnt/rootsnap.
# - путь /dev/ol_vbox/rootsnap — устройство-снимок, созданное ранее.
sudo mount -o ro /dev/ol_vbox/rootsnap /mnt/rootsnap

Отмонтируем снимок, когда просмотр/бэкап завершены.
sudo umount /mnt/rootsnap
```

```

hin@vbox:~$ sudo mkdir -p /mnt/rootsnap
hin@vbox:~$ sudo mount -o ro /dev/ol_vbox/rootsnap /mnt/rootsnap
mount: /mnt/rootsnap: wrong fs type, bad option, bad superblock on /dev/mapper/ol_vb
ox-rootsnap, missing codepage or helper program, or other error.
        dmesg(1) may have more information after failed mount system call.
hin@vbox:~$ sudo mount -o ro,nouuid /dev/ol_vbox/rootsnap /mnt/rootsnap
hin@vbox:~$ sudo umount /mnt/rootsnap
hin@vbox:~$ lsblk

```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	8:0	0	20G	0	disk	
├─sda1	8:1	0	1M	0	part	
├─sda2	8:2	0	1G	0	part	/boot
├─sda3	8:3	0	19G	0	part	
│ └─ol_vbox-swap	252:1	0	2G	0	lvm	[SWAP]
│ └─ol_vbox-root-real	252:3	0	19G	0	lvm	
│ └─ol_vbox-root	252:0	0	19G	0	lvm	/
│ └─ol_vbox-rootsnap	252:5	0	19G	0	lvm	
└─sdb	8:16	0	8G	0	disk	
└─ol_vbox-pglv	252:2	0	5G	0	lvm	/var/lib/pgsql
└─ol_vbox-root-real	252:3	0	19G	0	lvm	
└─ol_vbox-root	252:0	0	19G	0	lvm	/
└─ol_vbox-rootsnap	252:5	0	19G	0	lvm	
└─ol_vbox-rootsnap-cow	252:4	0	820M	0	lvm	
└─ol_vbox-rootsnap	252:5	0	19G	0	lvm	
sr0	11:0	1	1024M	0	rom	

Рис. 16: Создание снапшота

3.5. Генерируем изменения (500 МБ «мусора» во /var/tmp)

```

sudo dd if=/dev/zero of=/var/tmp/bigfile bs=1M count=500 status=progress
sync

```

```

hin@vbox:~$ sudo dd if=/dev/zero of=/var/tmp/bigfile bs=1M count=500 status=progress
441450496 bytes (441 MB, 421 MiB) copied, 2 s, 206 MB/s
500+0 records in
500+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 2.66966 s, 196 MB/s

```

Рис. 17: Генерация мусора на 500 мб

3.6. Смотрим сколько места занято в COW-области снапшота

```

# Поле Data% покажет заполняемость
sudo lvs -o lv_name,lv_size,data_percent,origin /dev/ol_vbox/rootsnap
sync

```

```

hin@vbox:~$ sudo lvs -o lv_name,lv_size,data_percent,origin /dev/ol_vbox/rootsnap
LV      LSize  Data%  Origin
rootsnap 820,00m 61,56  root

```

Рис. 18: Snapshot занят на 61.56%

3.7. Откат к моменту создания снимка

```
sudo lvconvert --merge /dev/ol_vbox/rootsnap
# после перезапуска root вернётся к «снимку» sudo reboot
```

```
hin@vbox:~$ sudo lvconvert --merge /dev/ol_vbox/rootsnap
Delaying merge since origin is open.
Merging of snapshot ol_vbox/rootsnap will occur on next activation of ol_vbox/root.
hin@vbox:~$ sudo vgs
VG      #PV #LV #SN Attr   VSize  VFree
ol_vbox  2   3   1 wz--n- 26,99g 200,00m
hin@vbox:~$ sudo lvs -o lv_name,lv_size,data_percent,origin /dev/ol_vbox/rootsnap
LV      LSize  Data%  Origin
[rootsnap] 820,00m 61,58  root
```

Рис. 19: Откат снимка. Как видим: память ещё не высвободилась

3.8. Проверяем что место вернулось и снимка нет

```
# VFree опять +1G
sudo vgs

# ничего не выведет — снимок удалён
sudo lvs | grep rootsnap
```

```
hin@vbox:~$ sudo vgs
[sudo] пароль для hin:
VG      #PV #LV #SN Attr   VSize  VFree
ol_vbox  2   3   0 wz--n- 26,99g 1020,00m
hin@vbox:~$ sudo lvs | grep rootsnap
```

Рис. 20: Результат после перезагрузки.

2.3 Задача №3 – Организация доступа

2.4 Задача №4 – Работа с пакетным менеджером DNF

2.5 Задача №5 – Инициализация и настройка PostgreSQL

2.6 Задача №6 – Развёртывание Демо-базы и настройка доступа

2.7 Задача №7 – Установка и работа в PgAdmin

3 Выполнение задач – Использование скриптов

3.1 Задача №1 – Развёртывание VM Oracle Linux 8

3.2 Задача №2 – Настройка дисковой подсистемы с помощью менеджера томов LVM

3.3 Задача №3 – Организация доступа

3.4 Задача №4 – Работа с пакетным менеджером DNF

3.5 Задача №5 – Инициализация и настройка PostgreSQL

3.6 Задача №6 – Развёртывание Демо-базы и настройка доступа

3.7 Задача №7 – Установка и работа в PgAdmin