

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**Гордієнко Ю.Г., Таран В.І.**

# **ПОБУДОВА CLOUD-СИСТЕМ**

## **Лабораторний практикум**

Навчальний посібник  
для здобувачів ступеня магістра  
за освітньою програмою «Інженерія програмного забезпечення комп'ютерних систем»  
спеціальності 121 «Інженерія програмного забезпечення»  
за освітньою програмою «Комп'ютерні системи та мережі»  
спеціальності 123 «Комп'ютерна інженерія»  
за освітньою програмою «Інформаційні управляючі системи та технології»  
спеціальності 126 «Інформаційні системи та технології»

Електронне мережне навчальне видання

**ЗАТВЕРДЖЕНО**  
на засіданні кафедри обчислювальної техніки  
протокол № 10 від 25.05.2022

2022

## **Зміст**

Лабораторна робота №1 Архітектури розподілених систем — балансування навантаження веб-серверів.....	3
Лабораторна робота №2 Архітектури розподілених систем — кластер бази даних.....	12
Лабораторна робота №3 Віртуалізація сховища даних.....	21

# Лабораторна робота №1

## Архітектури розподілених систем — балансування навантаження веб-серверів

Мета роботи: ознайомитися із багаторівневими архітектурами на прикладі веб серверів та бази даних.

Завдання роботи:

- створити та налаштувати просту клієнт-серверну систему;
- створити та налаштувати 2 та 3 рівневу систему.

### Клієнт-серверна архітектура

Клієнт-сервер — архітектура (рис. 1.1) в якій клієнти звертаються до серверу, щоб отримати з нього дані, а потім відобразити ці дані користувачу. Зміни від клієнта у вигляді даних відправляються назад на сервер, який їх зберігає. Такий процес обміну даними характерний для веб-серверів.

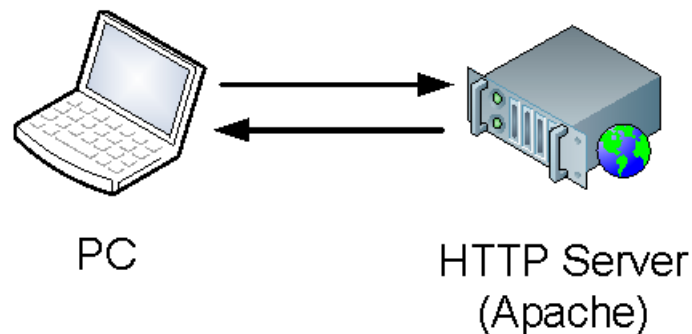


Рис. 1.1. Клієнт-серверна архітектура.

Щоб встановити веб-сервер *Apache*, необхідно виконати наступну команду. Приклад нижче показує встановлення для *Cent OS* із менеджером пакетів *yum* (рис. 1.2). Для *Debian/Ubuntu OS* менеджером пакетів буде *apt*. Назва пакету із веб-сервером може різнитись для різних систем.

```
root@vm121:~  
[root@vm121 ~]# yum install httpd  
Loaded plugins: fastestmirror  
Loading mirror speeds from cached hostfile  
* base: mirror.omnilance.com  
* extras: mirror.omnilance.com
```

Рис. 1.2. Встановлення веб-серверу *Apache* на *Cent OS*.

Далі може знадобитись підтвердження для встановлення необхідних пакетів (рис. 1.3).

```
=====
Package                Arch          Version      Repository    Size
=====
Installing:
httpd                   x86_64       2.4.6-80.el7.centos  base          2.7 M
Installing for dependencies:
apr                     x86_64       1.4.8-3.el7_4.1     base          103 k
apr-util                x86_64       1.5.2-6.el7         base          92 k
httpd-tools             x86_64       2.4.6-80.el7.centos  base          89 k
mailcap                 noarch       2.1.41-2.el7        base          31 k

Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 3.0 M
Installed size: 10 M
Is this ok [y/d/N]: y
```

Рис. 1.3. Підтвердження для встановлення веб-серверу *Apache*.

Щоб запустити та налаштувати автоматичний запуск веб-серверу після перезавантаження системи, необхідно виконати наступні команди (рис. 1.4).

```
root@vm121:~# systemctl start httpd.service
root@vm121:~# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
root@vm121:~#
```

Рис. 1.4. Налаштування запуску веб-серверу.

Перевірка, що сервер працює і прослуховується на відповідних портах (рис. 1.5). У даному прикладі, це порт 80).

```
root@vm121:~# netstat -tln | grep 80
tcp6      0      0 :::80          :::*           LISTEN
1099/httpd
root@vm121:~#
```

Рис. 1.5. Перевірка, що веб-сервер працює.

Також модно відкрити веб-браузер, ввести *IP* адресу веб-серверу і перевірити чи відкривається веб-сторінка на вказаному сервері (рис. 1.6).

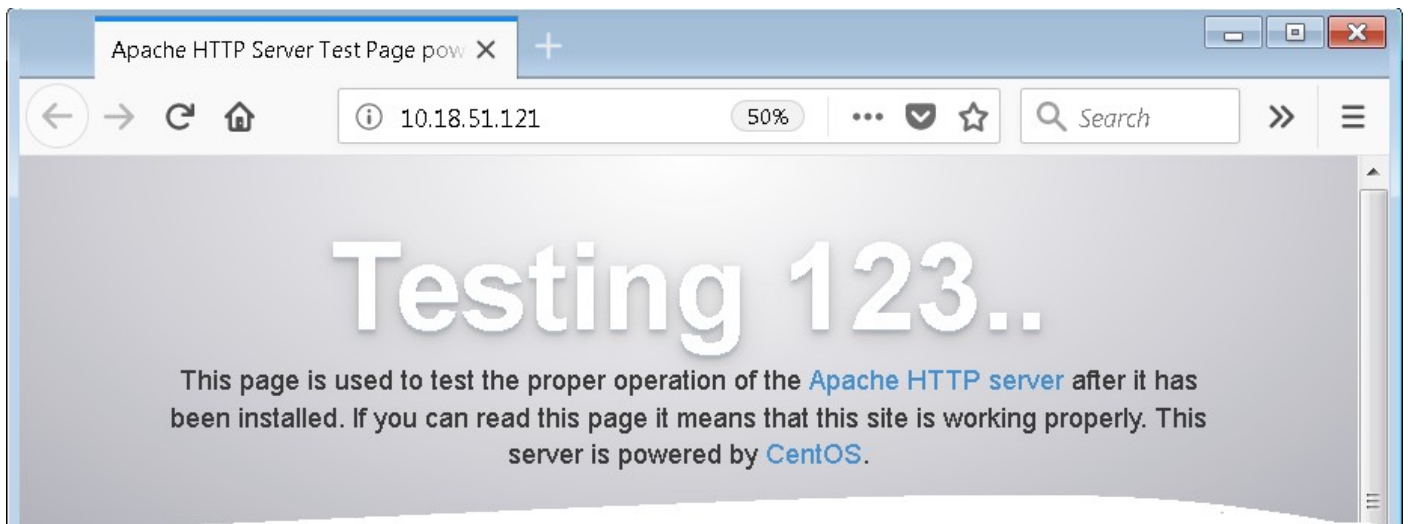


Рис. 1.6. Веб-сторінка встановленого серверу.

За замовчуванням всі файли веб-серверу зберігаються у директорії `/var/www/html`. Для прикладу, помістимо туди власний файл `employee.html` і, додавши його ім'я до IP адреси веб-серверу, перевіримо, що нова веб-сторінка відкривається (рис. 1.7).

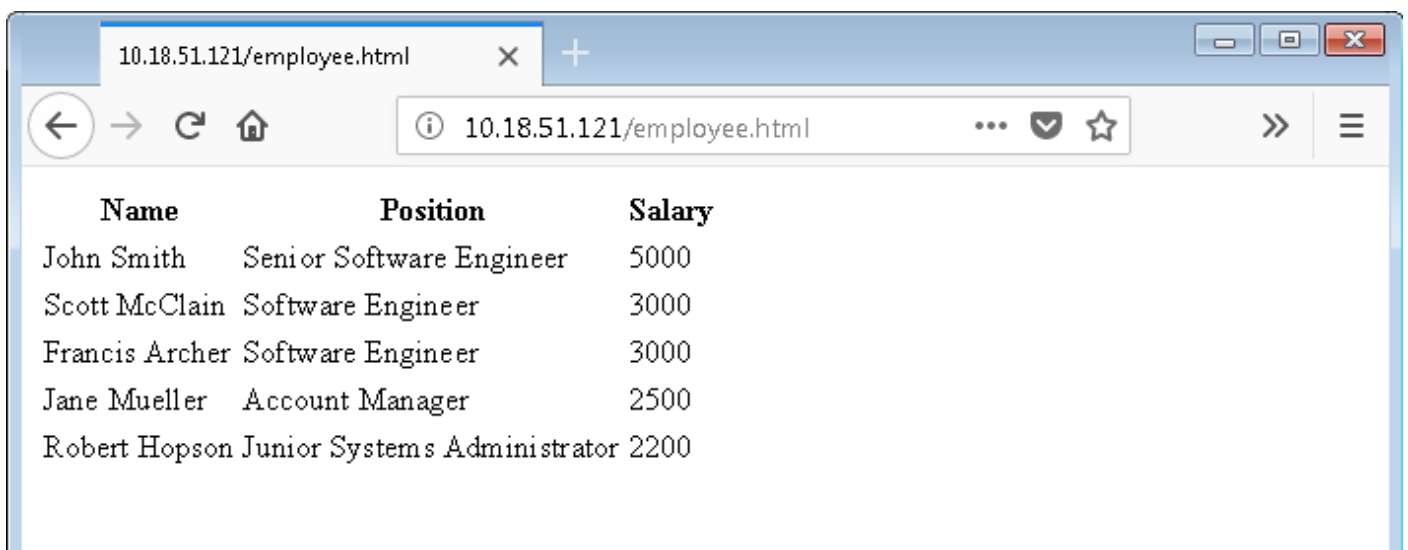


Рис. 1.7. Власна веб-сторінка на сервері.

### Налаштування: клієнт — веб-сервер — сервер бази даних

Для початку необхідно встановити сервер бази даних. В даному прикладі буде використовуватись *MySQL Community Server*. Для цього потрібно виконати наступні команди:

```
#yum install https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm  
#yum install mysql-server
```

Показані команди можуть відрізнитись у разі використання іншої версії пакету або іншої системи. Запуск та перевірка серверу бази даних показано на рис. 1.8.



```
root@vm122:~  
[root@vm122 ~]# systemctl enable mysqld  
[root@vm122 ~]# systemctl start mysqld  
[root@vm122 ~]# netstat -tln | grep 3306  
tcp6      0      0 :::33060          :::*              LISTEN  
1457/mysqld  
tcp6      0      0 :::3306           :::*              LISTEN  
1457/mysqld  
[root@vm122 ~]#
```

Рис. 1.8. Запуск та перевірка серверу бази даних.

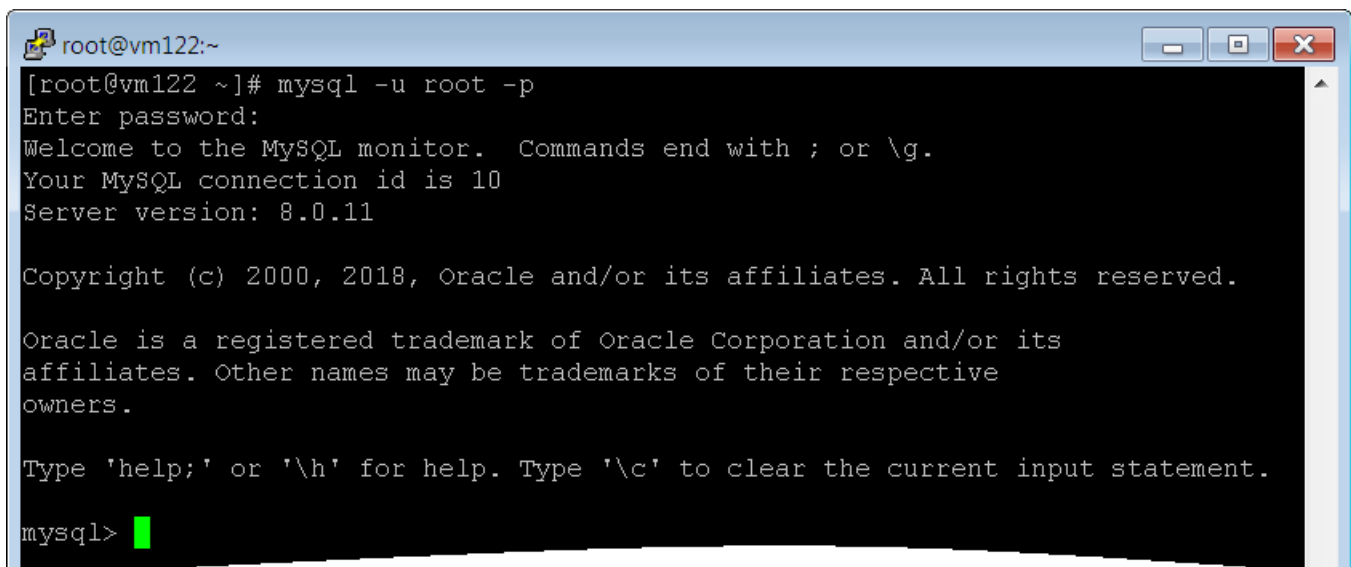
Під час встановлення серверу автоматично генерується пароль для користувача *root* (це адміністратор серверу бази даних, не системний *root* користувач). Пароль можна переглянути у файлі */var/log/mysqld.log* (рис. 1.9).



```
root@vm122:~  
[root@vm122 ~]# grep password /var/log/mysqld.log  
2018-06-07T18:05:35.955744Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: f0Tit#s8yC!x  
[root@vm122 ~]#
```

Рис. 1.9. Перегляд згенерованого паролю адміністратора.

Щоб підключитись до серверу бази даних можна використати вбудований клієнт *mysql*, де вказати параметри *-u* — ім'я користувача, та *-p* — прапор, що для входу використовується пароль (рис. 1.10). Пароль використовується той, що було отримано на попередньому кроці.



```
root@vm122:~  
[root@vm122 ~]# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 10  
Server version: 8.0.11  
  
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

Рис. 1.10. Вхід на сервер БД.

Щоб змінити пароль користувача, наприклад користувача *root* (рис. 1.11).

```
root@vm122:~
mysql> alter user root@localhost identified by 'c10udCourses2018!';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Рис. 1.11. Зміна паролю для *root* на *c10udCourses2018!* .

Далі можна створити нову базу даних або завантажити її з дампа файлу. У прикладі нижче показано команду для завантаження бази даних з дампа файлу:

```
mysql> source /root/testdb.sql
```

Перегляд таблиці у базі даних виконується наступними командами (рис. 1.12).

```
root@vm122:~
mysql> use testdb;
Database changed
mysql> select * from `employee`;
+----+-----+-----+-----+-----+
| id | firstname | lastname | position | salary |
+----+-----+-----+-----+-----+
| 1  | John      | Smith    | Senior Software Engineer | 5000.00 |
| 2  | Scott     | McClain  | Software Engineer         | 3000.00 |
| 3  | Francis   | Archer   | Software Engineer         | 3000.00 |
| 4  | Jane      | Mueller  | Account Manager           | 2500.00 |
| 5  | Robert    | Hopson   | Junior Systems Administrator | 2200.00 |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Рис. 1.12. Перегляд вмісту таблиці у базі даних.

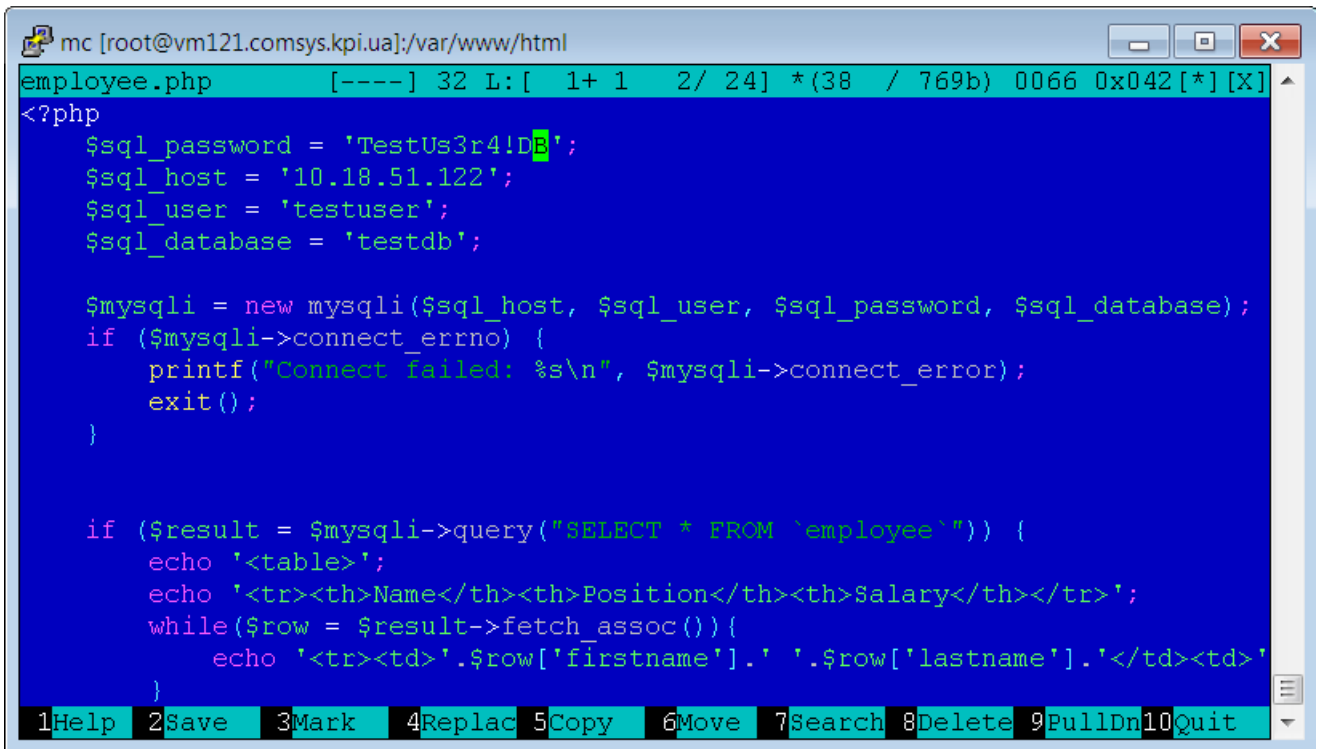
Також для зручності можна створити окремого користувача для доступу до певної бази даних. Для цього використовуються команди, показані нижче:

```
mysql> CREATE USER 'testuser'@'%' IDENTIFIED BY 'TestUs3r4!DB';
mysql> GRANT USAGE ON *.* TO 'testuser'@'%';
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON `testdb`.* TO 'testuser'@'%';
mysql> FLUSH PRIVILEGES;
```

Щоб переглядати вміст бази даних через окрему веб-сторінку, на веб-сервері необхідно додатково встановити розширення *PHP*. Приклад команди наведено нижче:

```
#yum install php php-mysqlnd
```

На веб-сервері створюється нова сторінка з кодом *PHP* (наприклад *employee.php*) у якому вказується параметри доступу до бази даних (*IP* адреса, ім'я користувача, пароль, ім'я бази даних, тощо), а також описуються операції, необхідні для завантаження та відображення даних. Приклад такої сторінки *PHP* показано на рис. 1.13.



```
mc [root@vm121.comsys.kpi.ua]:/var/www/html
employee.php [----] 32 L: [ 1+ 1 2/ 24] *(38 / 769b) 0066 0x042 [*] [X]
<?php
    $sql_password = 'TestUs3r4!D@';
    $sql_host = '10.18.51.122';
    $sql_user = 'testuser';
    $sql_database = 'testdb';

    $mysqli = new mysqli($sql_host, $sql_user, $sql_password, $sql_database);
    if ($mysqli->connect_errno) {
        printf("Connect failed: %s\n", $mysqli->connect_error);
        exit();
    }

    if ($result = $mysqli->query("SELECT * FROM `employee`")) {
        echo '<table>';
        echo '<tr><th>Name</th><th>Position</th><th>Salary</th></tr>';
        while($row = $result->fetch_assoc()) {
            echo '<tr><td>'.$row['firstname'].' '.$row['lastname'].'</td><td>';
        }
    }
}
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Рис. 1.13. Типова сторінки *PHP* для відображення вмісту таблиці з бази даних.

Перевірити створену сторінку можна через веб-браузер (рис. 1.14).

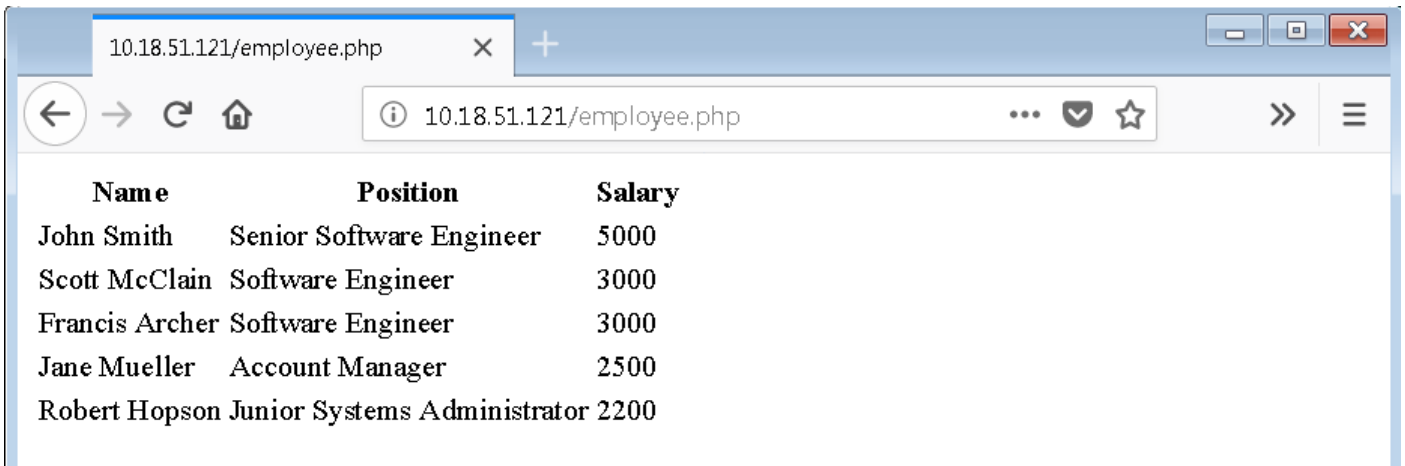


Рис. 1.14. Веб-сторінка із таблицею з бази даних.

### Налаштування: клієнт — балансувальник навантаження — веб-сервери — сервер бази даних

В якості балансувальника навантаження пропонується використати сервер *nginx*. Приклад такої інфраструктури показано на рис. 1.15.



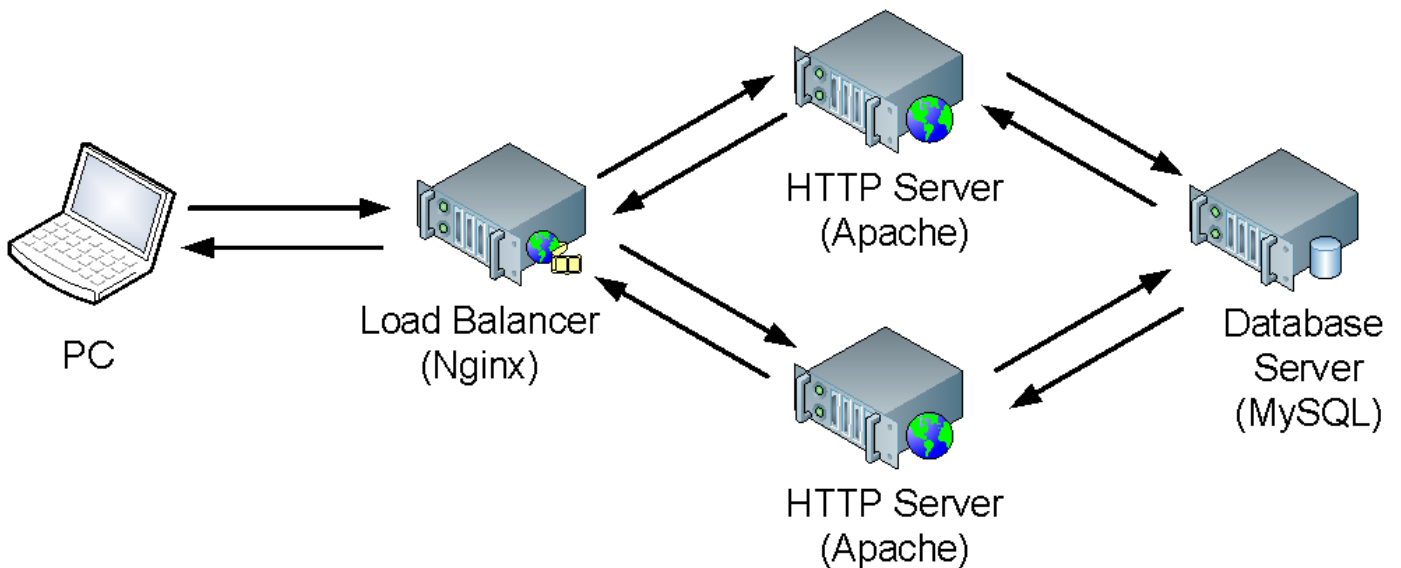


Рис. 1.15. Інфраструктура клієнт — балансувальник навантаження — веб-сервер — сервер бази даних.

Приклад встановлення та запуск балансувальника *nginx* відбувається командами, показаними нижче:

```
#yum install nginx
#systemctl enable nginx
#systemctl start nginx
```

Перевірити, що сервер балансування працює, можна шляхом вводу його *IP* адреси у браузер (рис. 1.16).

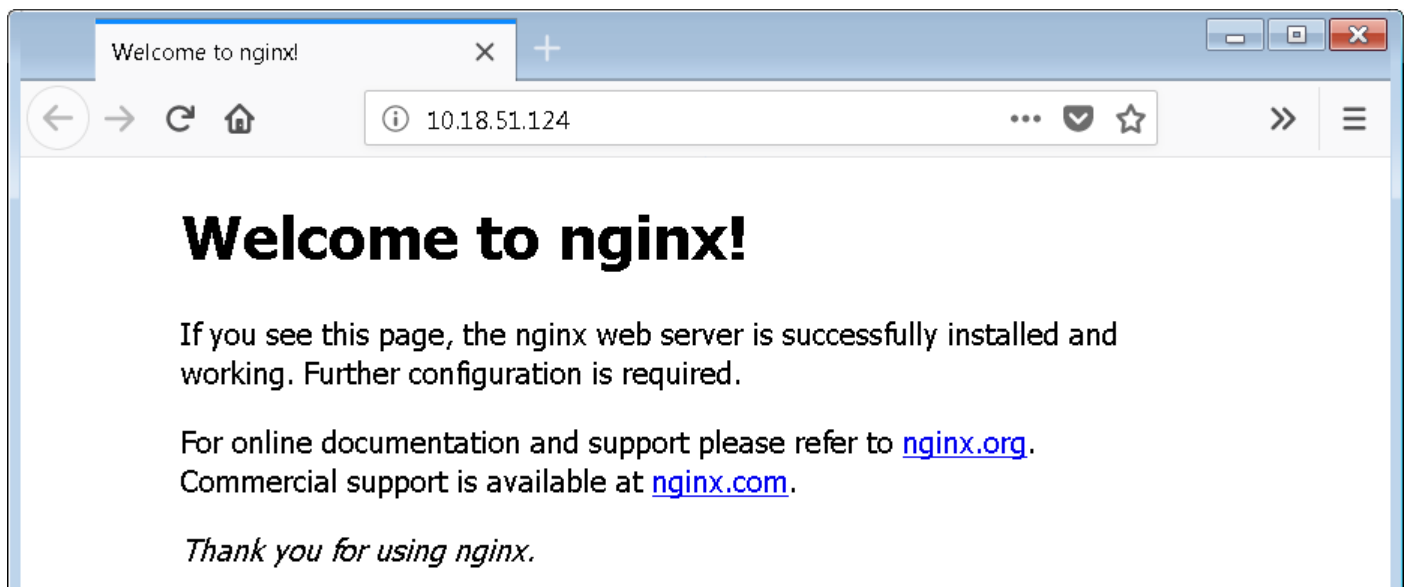
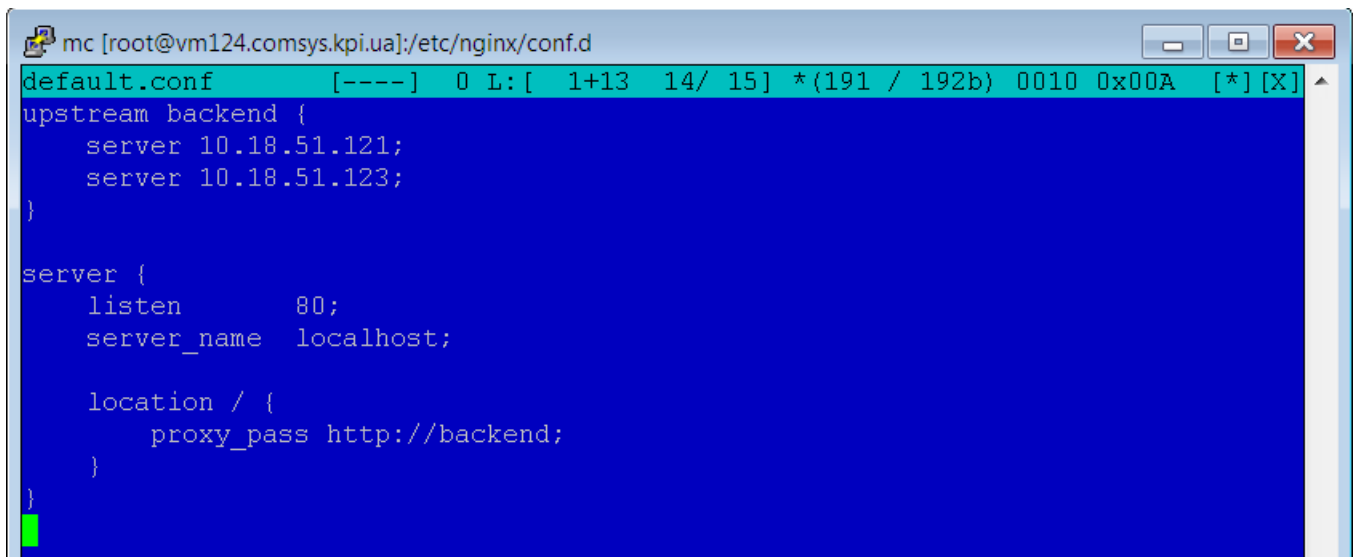


Рис. 1.16. Приклад сторінки серверу *nginx*.

Далі потрібно створити або відредагувати файл */etc/nginx/conf.d/default.conf* (фактичний шлях у різних системах може відрізнятись) і додати у блок *upstream backend* перелік *IP* адрес серверів між якими потрібно балансувати навантаження (рис. 1.17).



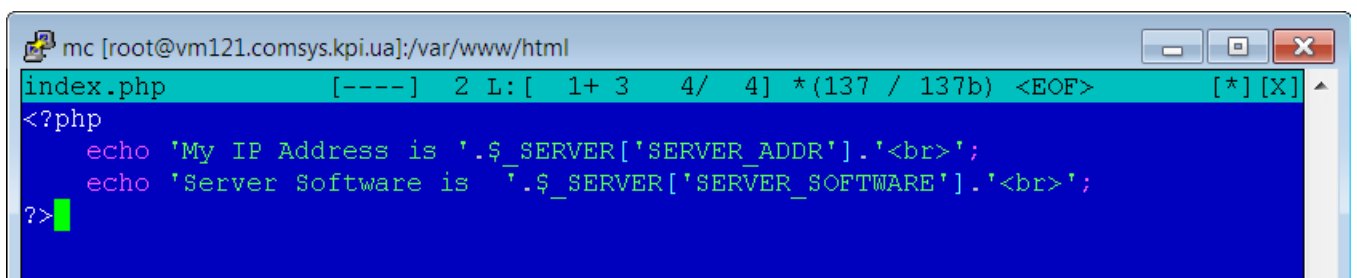
```
mc [root@vm124.comsys.kpi.ua]:/etc/nginx/conf.d
default.conf [----] 0 L: [ 1+13 14/ 15] *(191 / 192b) 0010 0x00A [*] [X] ^
upstream backend {
    server 10.18.51.121;
    server 10.18.51.123;
}

server {
    listen      80;
    server_name localhost;

    location / {
        proxy_pass http://backend;
    }
}
```

Рис. 1.17. Приклад файлу `/etc/nginx/conf.d/default.conf`.

Після внесення змін необхідно перезавантажити сервер *nginx*. Також додатково можна відредагувати, створену раніше сторінку *PHP*, і додати в неї код для відображення *IP* адрес серверу (рис. 1.18). В цьому випадку може бути легко побачити з якого серверу було отримано відповідь.



```
mc [root@vm121.comsys.kpi.ua]:/var/www/html
index.php [----] 2 L: [ 1+ 3 4/ 4] *(137 / 137b) <EOF> [*] [X] ^
<?php
    echo 'My IP Address is ' . $_SERVER['SERVER_ADDR'] . '<br>';
    echo 'Server Software is ' . $_SERVER['SERVER_SOFTWARE'] . '<br>';
?>
```

Рис. 1.18. Приклад модифікації сторінки *PHP* для відображення *IP* адрес серверу.

Звертаючись до *IP* адреси серверу балансування навантаження, запити будуть по чергово перенаправлятися на вказані у налаштування веб-сервери (рис. 1.19, 1.20).

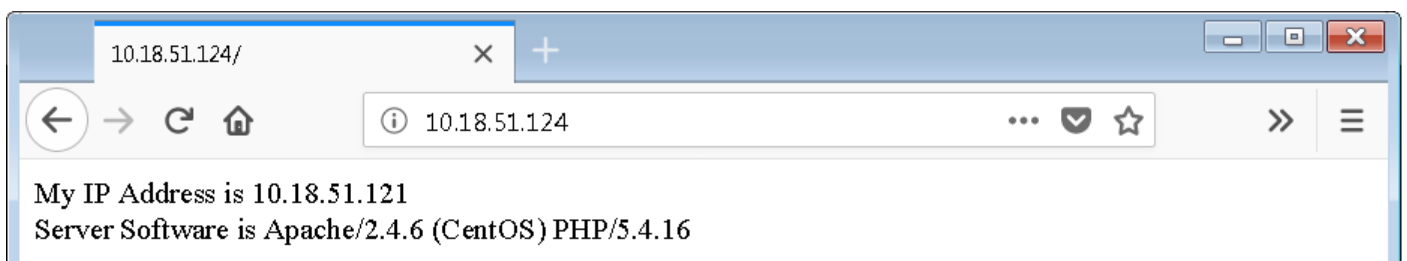


Рис. 1.19. Відповідь від першого веб-серверу.

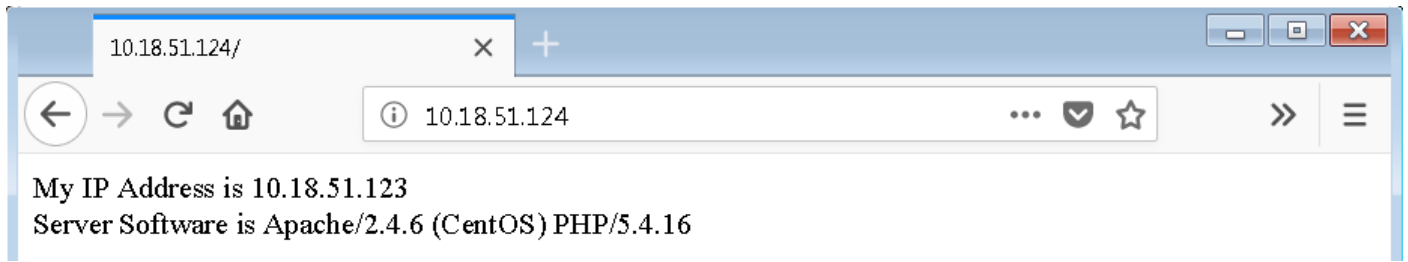


Рис. 1.20. Відповідь від другого веб-серверу.

Для зміни поведінки балансування навантаження можна додатково використовувати інші алгоритми балансування, такі як: *round-robin*, *least-connected*, *ip-hash*, тощо.

## Лабораторна робота №2

### Архітектури розподілених систем — кластер бази даних

Мета роботи: ознайомитися із багаторівневими архітектурами на прикладі кластеру бази даних.

Завдання роботи:

- масштабувати горизонтально окремі компоненти системи;
- реалізувати відмовостійкість системи.

**Налаштування: клієнт — балансувальник навантаження — веб-сервери — кластер бази даних**

Продовжуючи розширення, створеної у попередній роботі інфраструктури, необхідно налаштувати кластер *MySQL InnoDB*. В загальному вигляді, отримана інфраструктура, матиме наступний вигляд (рис. 2.1).

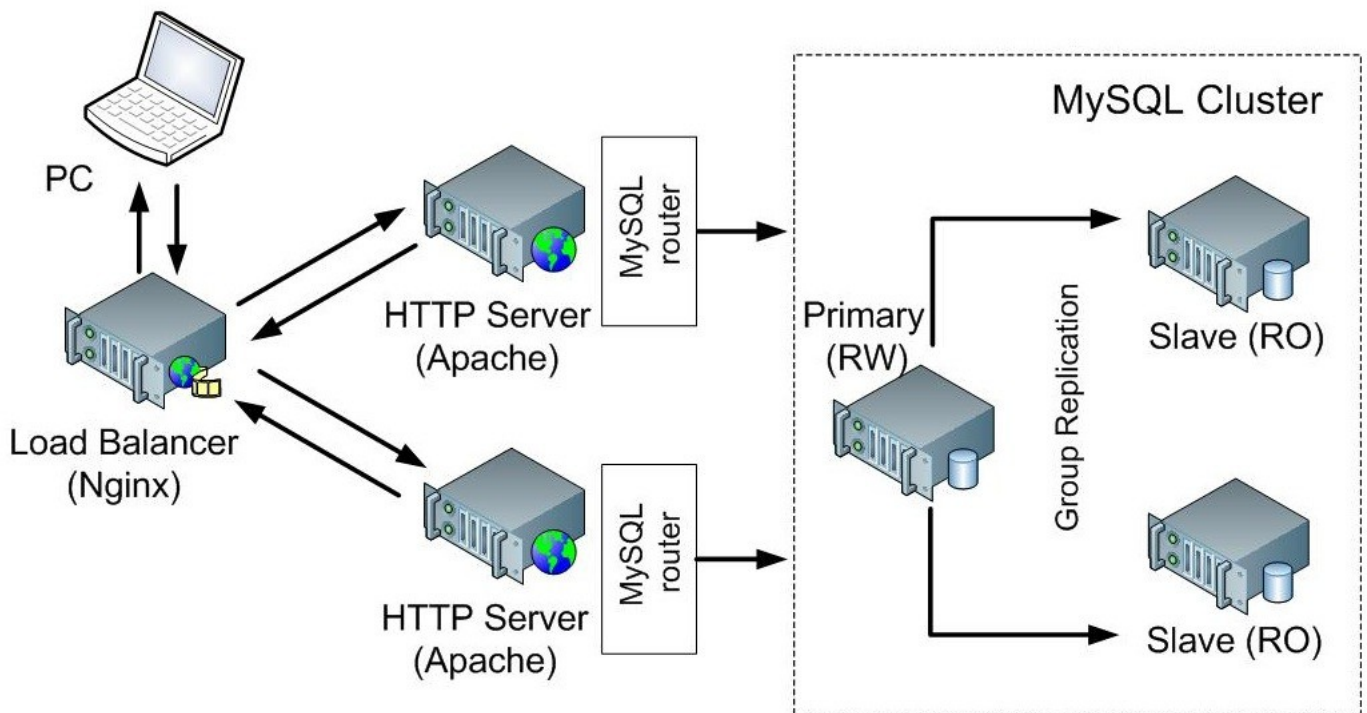
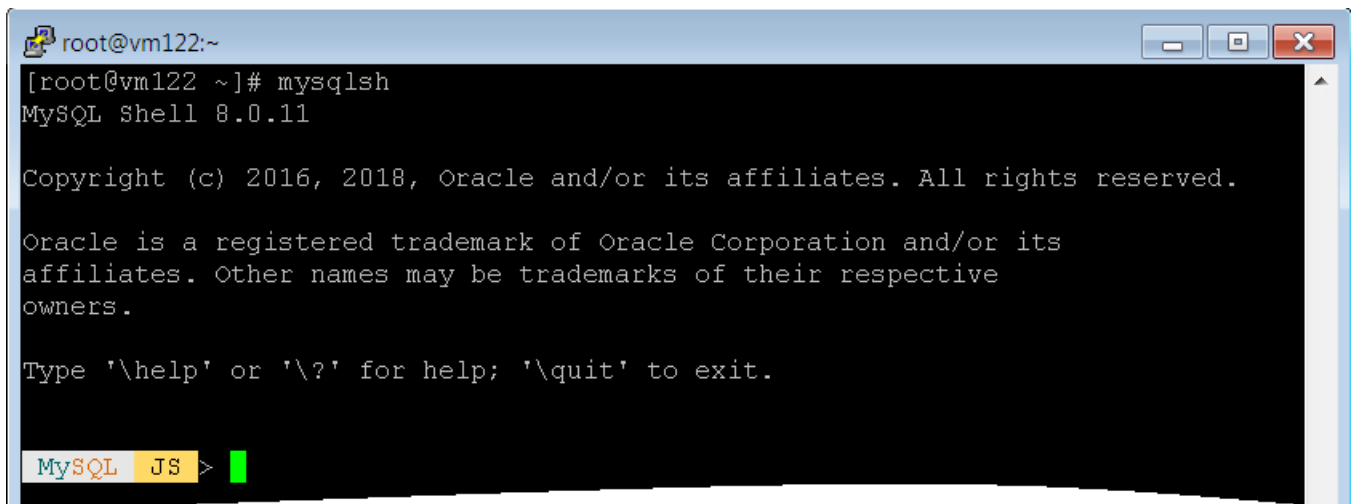


Рис. 2.1. Доповнення інфраструктури кластером бази даних.

На кожному вузлі системи встановіть *mysql-shell*. Приклад команди показано нижче:

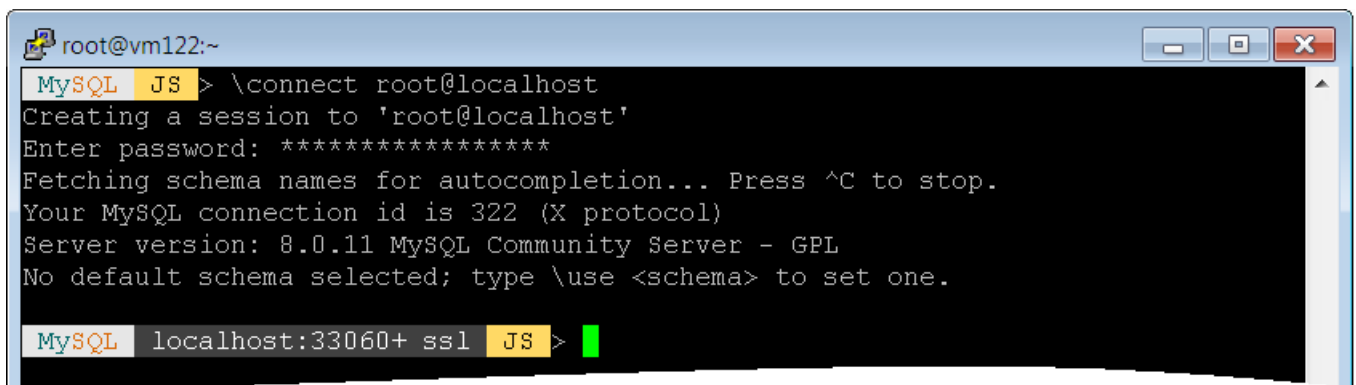
```
#yum install mysql-shell
```

Зайдіть у *MySQL Shell* командою *mysqlsh* (рис. 2.2) та підключіться до локального серверу бази даних (рис. 2.3).



```
root@vm122:~  
[root@vm122 ~]# mysqlsh  
MySQL Shell 8.0.11  
  
Copyright (c) 2016, 2018, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type '\help' or '? ' for help; '\quit' to exit.  
  
MySQL JS >
```

Рис. 2.3. Консоль *mysqlsh*.



```
root@vm122:~  
MySQL JS > \connect root@localhost  
Creating a session to 'root@localhost'  
Enter password: *****  
Fetching schema names for autocompletion... Press ^C to stop.  
Your MySQL connection id is 322 (X protocol)  
Server version: 8.0.11 MySQL Community Server - GPL  
No default schema selected; type \use <schema> to set one.  
  
MySQL localhost:33060+ ssl JS >
```

Рис. 2.4. Підключення до локального серверу бази даних.

Далі необхідно виконати автоматичну перевірку сумісності налаштувань серверу *MySQL* для роботи у кластері (рис. 2.5).

```
root@vm122:~  
MySQL localhost:33060+ ssl JS > dba.checkInstanceConfiguration()  
Validating local MySQL instance listening at port 3306 for use in an InnoDB cluster...  
  
This instance reports its own address as vm122.comsys.kpi.ua  
Clients and other cluster members will communicate with it through this address  
by default. If this is not correct, the report_host MySQL system variable should  
be changed.  
  
Checking whether existing tables comply with Group Replication requirements...  
No incompatible tables detected  
  
Checking instance configuration...  
  
Some configuration options need to be fixed:  
+-----+-----+-----+-----+  
| Variable | Current Value | Required Value | Note  
|-----|-----|-----|-----|  
| binlog_checksum | CRC32 | NONE | Update the server  
variable  
| enforce_gtid_consistency | OFF | ON | Update read-only v
```

Рис. 2.5. Автоматична перевірка налаштувань.

Після цього буде видано перелік налаштувань, які необхідно змінити на сервері (віддалений доступ, обліковий запис адміністратора, тощо). Деякі зміни можна буде внести автоматично в ході виконання наступної команди (рис. 2.6, 2.7).

```

root@vm122:~
MySQL localhost:33060+ ssl JS > dba.configureInstance()
Configuring local MySQL instance listening at port 3306 for use in an InnoDB cluster...

This instance reports its own address as vm122.comsys.kpi.ua
Clients and other cluster members will communicate with it through this address
by default. If this is not correct, the report_host MySQL system variable should
be changed.

WARNING: User 'root' can only connect from localhost.
If you need to manage this instance while connected from other hosts, new account
t(s) with the proper source address specification must be created.

1) Create remotely usable account for 'root' with same grants and password
2) Create a new admin account for InnoDB cluster with minimal required grants
3) Ignore and continue
4) Cancel

Please select an option [1]: 1
Please provide a source address filter for the account (e.g: 192.168.% or % etc)
or leave empty and press Enter to cancel.
Account Host: %

Some configuration options need to be fixed:

```

Рис. 2.6. Налаштування доступу до серверу перед його додаванням у кластер.

```

root@vm122:~
-----+
| Variable          | Current Value | Required Value | Note
|-----+-----+-----+-----+
| binlog_checksum   | CRC32         | NONE           | Update the server
variable
| enforce_gtid_consistency | OFF          | ON             | Update read-only v
variable and restart the server |
| gtid_mode         | OFF          | ON             | Update read-only v
variable and restart the server |
+-----+-----+-----+-----+
-----+

Do you want to perform the required configuration changes? [y/n]: y
Do you want to restart the instance after configuring it? [y/n]: y

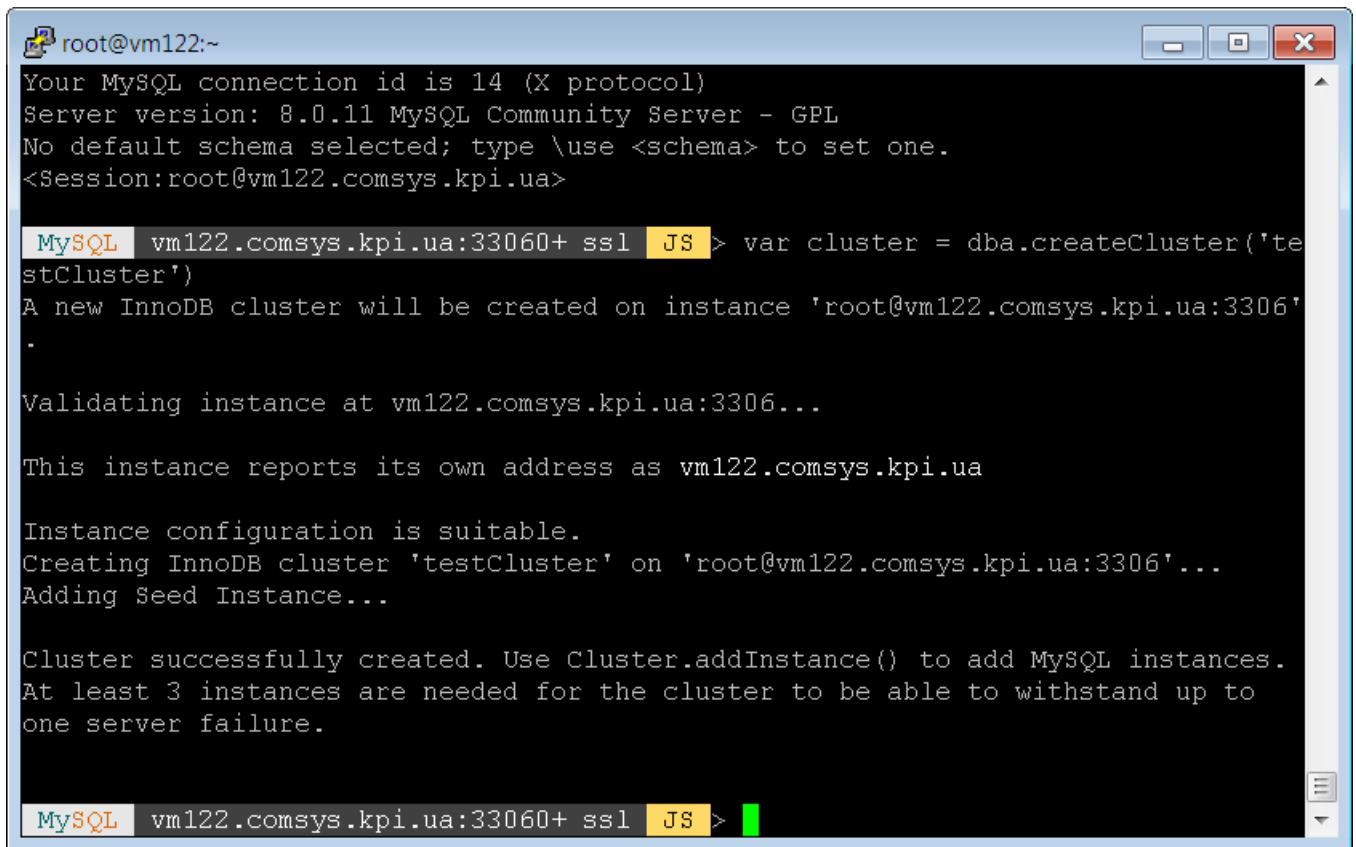
Cluster admin user 'root'@'%' created.
Configuring instance...
The instance 'localhost:3306' was configured for cluster usage.
Restarting MySQL...
MySQL server at localhost:3306 was restarted.

MySQL localhost:33060+ ssl JS > █

```

Рис. 2.7. Оновлення налаштувань серверу перед його додаванням у кластер.

Наступним кроком є створення кластеру бази даних. Даний крок виконується лише один раз на сервері, який був обраний в якості головного вузла (рис. 2.8).



```
root@vm122:~  
Your MySQL connection id is 14 (X protocol)  
Server version: 8.0.11 MySQL Community Server - GPL  
No default schema selected; type \use <schema> to set one.  
<Session:root@vm122.comsys.kpi.ua>  
  
MySQL vm122.comsys.kpi.ua:33060+ ssl JS > var cluster = dba.createCluster('testCluster')  
A new InnoDB cluster will be created on instance 'root@vm122.comsys.kpi.ua:3306'  
.  
  
Validating instance at vm122.comsys.kpi.ua:3306...  
  
This instance reports its own address as vm122.comsys.kpi.ua  
  
Instance configuration is suitable.  
Creating InnoDB cluster 'testCluster' on 'root@vm122.comsys.kpi.ua:3306'...  
Adding Seed Instance...  
  
Cluster successfully created. Use Cluster.addInstance() to add MySQL instances.  
At least 3 instances are needed for the cluster to be able to withstand up to  
one server failure.  
  
MySQL vm122.comsys.kpi.ua:33060+ ssl JS >
```

Рис. 2.8. Налаштування головного вузла кластеру бази даних.

Далі потрібно додати нові вузли до створеного кластеру. Для цього необхідно виконати, описані вище кроки (окрім створення кластеру) на інших серверах баз даних, які будуть виконувати роль підлеглих вузлів. Після завершення підготовчих кроків, можна додати інший вузол до кластеру бази даних (рис. 2.9). Цей крок виконується на головному вузлі кластеру і повторюється для кожного вузла, який додається до кластеру. В якості аргументів вказується ім'я користувача та ім'я серверу (не IP адреса).



```
root@vm122:~  
MySQL localhost:33060+ ssl JS > cluster.addInstance("root@vm125.comsys.kpi.ua:3306");  
A new instance will be added to the InnoDB cluster. Depending on the amount of data on the cluster this might take from a few seconds to several hours.  
  
Please provide the password for 'root@vm125.comsys.kpi.ua:3306': *****  
**  
Adding instance to the cluster ...  
  
Validating instance at vm125.comsys.kpi.ua:3306...  
  
This instance reports its own address as vm125.comsys.kpi.ua  
  
Instance configuration is suitable.  
The instance 'root@vm125.comsys.kpi.ua:3306' was successfully added to the cluster.  
  
MySQL localhost:33060+ ssl JS > cluster.status()  
{  
  "clusterName": "testCluster",  
  "defaultReplicaSet": {  
    "name": "default",  
    "primary": "vm122.comsys.kpi.ua:3306",
```

Рис. 2.9. Додавання нового вузла до створеного кластеру.

Після успішного додавання вузла до кластеру, можна переглянути його статус (рис. 2.10). Додавши ще один вузол, можна буде побачити його у переліку вузлів кластеру (2.11).

```
root@vm122:~  
MySQL vm122.comsys.kpi.ua:33060+ ssl JS > cluster.status()  
{  
  "clusterName": "testCluster",  
  "defaultReplicaSet": {  
    "name": "default",  
    "primary": "vm122.comsys.kpi.ua:3306",  
    "ssl": "REQUIRED",  
    "status": "OK_NO_TOLERANCE",  
    "statusText": "Cluster is NOT tolerant to any failures.",  
    "topology": {  
      "vm122.comsys.kpi.ua:3306": {  
        "address": "vm122.comsys.kpi.ua:3306",  
        "mode": "R/W",  
        "readReplicas": {},  
        "role": "HA",  
        "status": "ONLINE"  
      },  
      "vm125.comsys.kpi.ua:3306": {  
        "address": "vm125.comsys.kpi.ua:3306",  
        "mode": "R/O",  
        "readReplicas": {},  
        "role": "HA",  
        "status": "ONLINE"  
      }  
    }  
  }  
}
```

Рис. 2.10. Статус кластеру із двома вузлами.

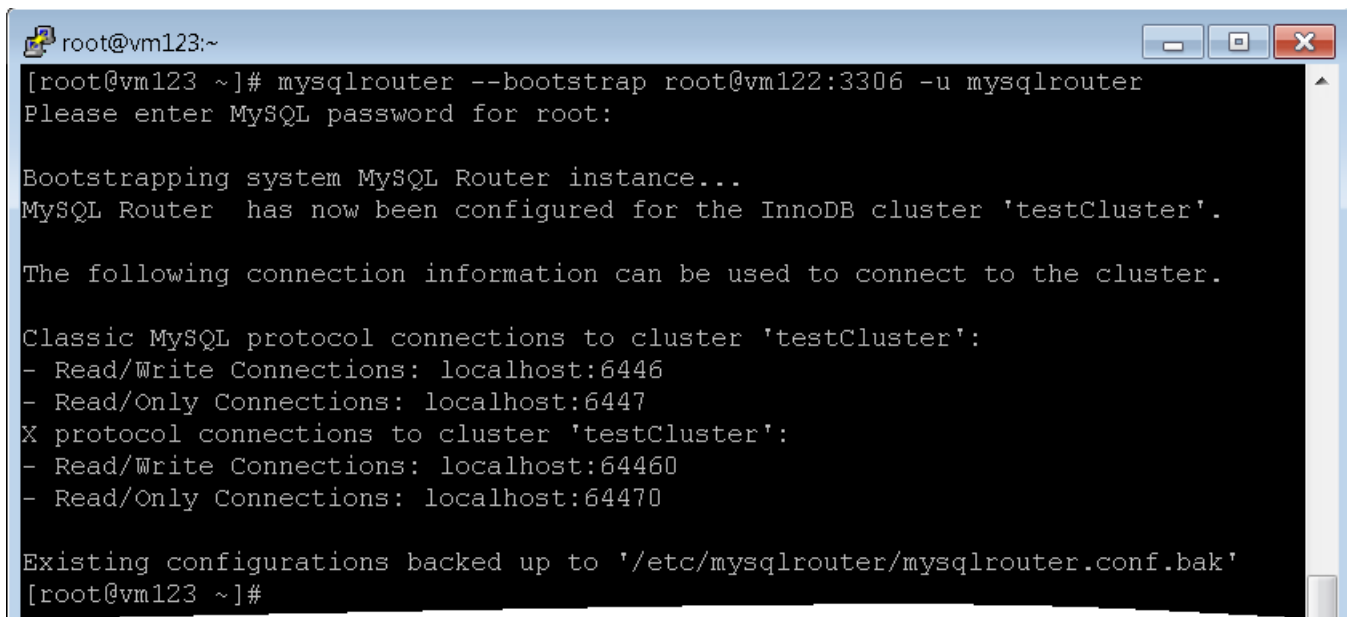
```
root@vm122:~  
MySQL vm122.comsys.kpi.ua:33060+ ssl JS > cluster.status()  
{  
  "clusterName": "testCluster",  
  "defaultReplicaSet": {  
    "name": "default",  
    "primary": "vm122.comsys.kpi.ua:3306",  
    "ssl": "REQUIRED",  
    "status": "OK",  
    "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",  
    "topology": {  
      "vm122.comsys.kpi.ua:3306": {  
        "address": "vm122.comsys.kpi.ua:3306",  
        "mode": "R/W",  
        "readReplicas": {},  
        "role": "HA",  
        "status": "ONLINE"  
      },  
      "vm125.comsys.kpi.ua:3306": {  
        "address": "vm125.comsys.kpi.ua:3306",  
        "mode": "R/O",  
        "readReplicas": {},  
        "role": "HA",  
        "status": "ONLINE"  
      },  
      "vm126.comsys.kpi.ua:3306": {  
        "address": "vm126.comsys.kpi.ua:3306",  
        "mode": "R/O",  
        "readReplicas": {},  
        "role": "HA",  
        "status": "ONLINE"  
      }  
    }  
  },  
  "groupInformationSourceMember": "mysql://root@vm122.comsys.kpi.ua:3306"  
}
```

Рис. 2.11. Статус кластеру із трьома вузлами.

Тепер в кластері бази даних є три вузли. Один виконує операції читання-запису, два інші — лише операції читання. Якщо один із вузлів вийде з ладу, кластер продовжить виконувати свої функції за рахунок інших вузлів.

Щоб підключитися до кластеру бази даних можна використовувати *MySQL Router*. *MySQL Router* це складова кластеру *MySQL InnoDB*. Він забезпечує прозору маршрутизацію запитів між клієнтською програмою та серверами *MySQL*, об'єднаних у кластер. Це забезпечує високу доступність даних, можливість масштабувати системи та відмовостійкість, шляхом ефективної маршрутизації трафіку до відповідних серверів бази даних. *MySQL Router* встановлюється на кожному сервері, з якого потрібно отримати доступ до інформації з кластеру бази даних. Приклад його встановлення та підключення до кластеру (рис. 2.12) показано нижче:

```
#yum install mysql-router  
#mysqlrouter --bootstrap root@ім'я серверу:3306 -u  
mysqlrouter
```

A terminal window titled 'root@vm123:~' showing the execution of the 'mysqlrouter' command. The command is 'mysqlrouter --bootstrap root@vm122:3306 -u mysqlrouter'. The output indicates that the MySQL Router has been configured for the InnoDB cluster 'testCluster'. It provides connection information for both Classic MySQL and X protocol connections. The Classic MySQL protocol connections are: Read/Write on localhost:6446 and Read/Only on localhost:6447. The X protocol connections are: Read/Write on localhost:64460 and Read/Only on localhost:64470. Existing configurations are backed up to '/etc/mysqlrouter/mysqlrouter.conf.bak'. The prompt returns to '[root@vm123 ~]#'.

```
[root@vm123 ~]# mysqlrouter --bootstrap root@vm122:3306 -u mysqlrouter
Please enter MySQL password for root:

Bootstrapping system MySQL Router instance...
MySQL Router has now been configured for the InnoDB cluster 'testCluster'.

The following connection information can be used to connect to the cluster.

Classic MySQL protocol connections to cluster 'testCluster':
- Read/Write Connections: localhost:6446
- Read/Only Connections: localhost:6447
X protocol connections to cluster 'testCluster':
- Read/Write Connections: localhost:64460
- Read/Only Connections: localhost:64470

Existing configurations backed up to '/etc/mysqlrouter/mysqlrouter.conf.bak'
[root@vm123 ~]#
```

Рис. 2.12. Підключення *MySQL Router* до створеного кластеру *MySQL InnoDB*.

Щоб *MySQL Router* автоматично запускався після перезавантаження системи, виконуються команди нижче:

```
# systemctl enable mysqlrouter.service
# systemctl start mysqlrouter.service
```

*MySQL Router* приймає запити на портах 6446 — читання-запис та 6447 — тільки читання. Цими портами слід замінити налаштування у *PHP* файлах, зроблених у попередній роботі. Після цього створена раніше система зможе отримувати дані зі створеного кластеру *MySQL InnoDB*.

## Лабораторна робота №3

### Віртуалізація сховища даних

Мета роботи: ознайомитися із технологіями віртуалізації сховища даних.

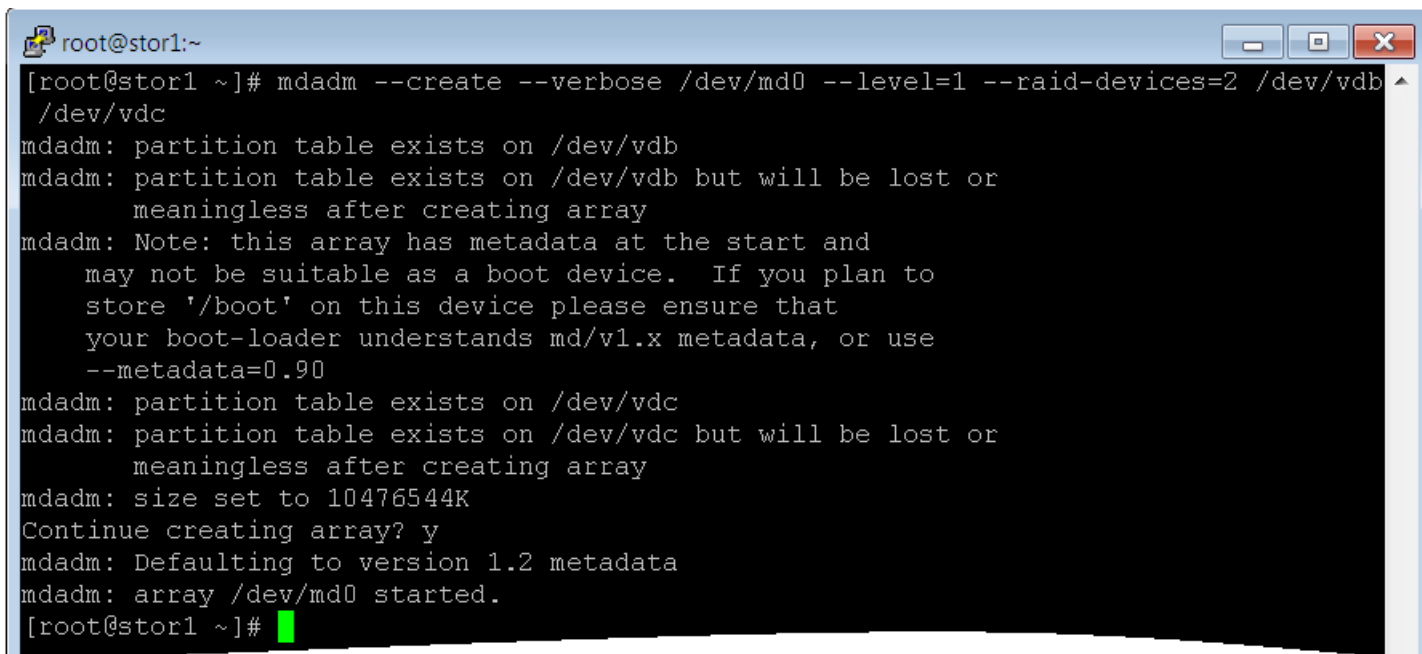
Завдання роботи:

- створити та налаштувати *RAID* масив;
- створити і налаштувати *LVM*;
- налаштувати розподілену файловою системою *GlusterFS*.

#### Програмний *RAID* масив

*RAID* масив це віртуальні носії даних, які складаються з двох і більше фізичних пристроїв носіїв даних. Це дозволяю об'єднувати декілька пристроїв (зазвичай цілого диску або окремих розділів) в один пристрій для зберігання, наприклад, однієї файлової системи. Деякі рівні *RAID* забезпечують відмовостійкість, що дозволяє зберегти дані у випадку виходу із ладу одного із фізичних носіїв даних. Інші рівні можуть забезпечувати розширення об'єму сховища даних, за рахунок розподілу даних між декількома фізичними носіями.

У *Linux* є реалізація програмного *RAID* у вигляді драйверу пристроїв *md* (*Multiple Devices*). Для керування цими *md* пристроями використовується утиліта *mdadm*. Для прикладу створимо *RAID* масив 1-го рівня (дзеркальний) із двох носіїв даних *vdb* та *vdc* (рис. 3.1). В інших системах назви пристроїв можуть відрізнятись.



```
root@stor1:~  
[root@stor1 ~]# mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/vdb  
/dev/vdc  
mdadm: partition table exists on /dev/vdb  
mdadm: partition table exists on /dev/vdb but will be lost or  
meaningless after creating array  
mdadm: Note: this array has metadata at the start and  
may not be suitable as a boot device. If you plan to  
store '/boot' on this device please ensure that  
your boot-loader understands md/v1.x metadata, or use  
--metadata=0.90  
mdadm: partition table exists on /dev/vdc  
mdadm: partition table exists on /dev/vdc but will be lost or  
meaningless after creating array  
mdadm: size set to 10476544K  
Continue creating array? y  
mdadm: Defaulting to version 1.2 metadata  
mdadm: array /dev/md0 started.  
[root@stor1 ~]#
```

Рис. 3.1. Приклад створення *RAID* масиву 1-го рівня.

Після створення масиву можна переглянути статус синхронізації (рис. 3.2).

```
root@stor1:~  
[root@stor1 ~]# cat /proc/mdstat  
Personalities : [raid0] [raid1]  
md0 : active raid1 vdc[1] vdb[0]  
      10476544 blocks super 1.2 [2/2] [UU]  
      [>.....] resync = 2.0% (219520/10476544) finish=4.6min speed=3  
6586K/sec  
  
unused devices: <none>  
[root@stor1 ~]#
```

Рис. 3.2. Статус синхронізації RAID масиву.

Детальну інформацію про RAID масив можна отримати за допомогою утиліти *mdadm*, вказавши назву потрібного пристрою (рис. 3.3).

```
root@stor1:~  
[root@stor1 ~]# mdadm --detail /dev/md0  
/dev/md0:  
  Version : 1.2  
  Creation Time : Thu May 31 08:13:07 2018  
  Raid Level : raid1  
  Array Size : 10476544 (9.99 GiB 10.73 GB)  
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)  
  Raid Devices : 2  
  Total Devices : 2  
  Persistence : Superblock is persistent  
  
  Update Time : Thu May 31 08:15:52 2018  
  State : clean  
  Active Devices : 2  
  Working Devices : 2  
  Failed Devices : 0  
  Spare Devices : 0  
  
Consistency Policy : resync  
  
  Name : stor1.comsys.kpi.ua:0 (local to host stor1.comsys.kpi.ua)  
  UUID : 27069543:744428a5:0ee5d62b:6121c619  
  Events : 17  
  
  Number  Major  Minor  RaidDevice State  
    0     252    16      0     active sync  /dev/vdb  
    1     252    32      1     active sync  /dev/vdc  
[root@stor1 ~]#
```

Рис. 3.3. Детальну інформацію про RAID масив.

Для перевірки працездатності створеного масиву можна зімітувати вихід з ладу одного з пристроїв (рис. 3.4). Статус відповідного пристрою зміниться на пошкоджений - (F) (рис. 3.5). Видалення носія з масиву виконується командою, показаною на рис. 3.6.

```
root@stor1:~  
[root@stor1 ~]# mdadm /dev/md0 --fail /dev/vdc  
mdadm: set /dev/vdc faulty in /dev/md0  
[root@stor1 ~]#
```

Рис. 3.4. Імітація виходу з ладу одного з пристроїв масиву.

```
root@stor1:~  
[root@stor1 ~]# cat /proc/mdstat  
Personalities : [raid0] [raid1]  
md0 : active raid1 vdc[1](F) vdb[0]  
      10476544 blocks super 1.2 [2/1] [U_]  
  
unused devices: <none>  
[root@stor1 ~]#
```

Рис. 3.5. Статус масиву зі зламаним пристроєм.

```
root@stor1:~  
[root@stor1 ~]# mdadm /dev/md0 --remove /dev/vdc  
mdadm: hot removed /dev/vdc from /dev/md0  
[root@stor1 ~]# cat /proc/mdstat  
Personalities : [raid0] [raid1]  
md0 : active raid1 vdb[0]  
      10476544 blocks super 1.2 [2/1] [U_]  
  
unused devices: <none>  
[root@stor1 ~]#
```

Рис. 3.6. Видалення пошкодженого носія даних.

Щоб замінити носій на новий виконаємо команду, показану на рис. 3.7. Новий носій має бути чистим. В разі якщо він вже був у складі іншого *RAID* масив, *MBR (Master Boot Record)* необхідно попередньо видалити, наприклад за допомогою наступної команди:

```
#dd if=/dev/zero of=/dev/vdc bs=1M count=100
```

```
root@stor1:~  
[root@stor1 ~]# mdadm /dev/md0 --add /dev/vdc  
mdadm: added /dev/vdc  
[root@stor1 ~]# cat /proc/mdstat  
Personalities : [raid0] [raid1]  
md0 : active raid1 vdc[2] vdb[0]  
      10476544 blocks super 1.2 [2/1] [U_]  
      [>.....] recovery = 1.5% (160512/10476544) finish=4.2min speed=  
40128K/sec  
  
unused devices: <none>  
[root@stor1 ~]#
```

Рис. 3.7. Додавання нового носія у *RAID* масив.

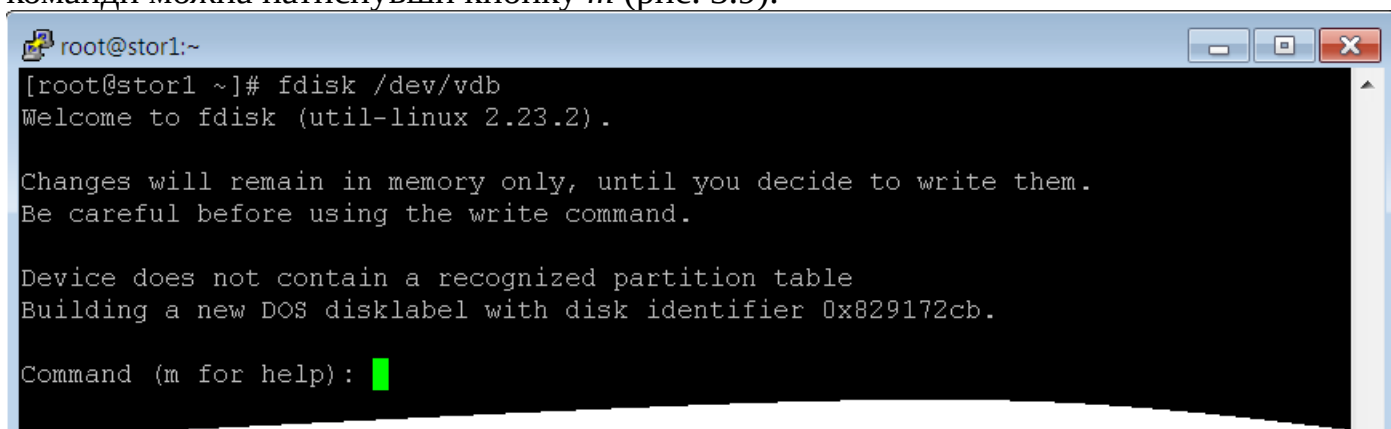
### LVM

*LVM* надає спосіб розподілу простору на пристроях носіїв даних, який є більш гнучким, ніж звичайні схеми розділення томів. Зокрема, диспетчер томів

може об'єднувати, розділяти разом або іншим чином об'єднувати розділи (або блокувати пристрої загалом) у більші віртуальні розділи, розмір яких адміністратори можуть змінювати або переміщувати, потенційно не перериваючи роботу системи.

Керування томами є лише однією з багатьох форм віртуалізації сховищ даних. Його реалізація відбувається на рівні в стеку драйверів пристроїв операційної системи. *LVM* додає рівень абстракції між фізичними/логічними дисками (розділами, з якими працюють *fdisk* і подібні програми) і файловою системою.

Для прикладу будемо використовувати два носії даних *vdb* та *vdc*. Кроки, що показано далі, виконані тільки для одного носія і мають бути повторені для другого. Для почату створимо один розділ з усього вільного об'єму на носії. Для цього необхідно запустити утиліту *fdisk* (рис. 3.8). Переглянути доступні команди можна натиснувши кнопку *m* (рис. 3.9).



```
root@stor1:~
[root@stor1 ~]# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x829172cb.

Command (m for help): █
```

Рис. 3.8. Утиліту *fdisk*.



```
root@stor1:~
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x52275e3a.

Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  g  create a new empty GPT partition table
  G  create an IRIX (SGI) partition table
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)

Command (m for help): █
```

Рис. 3.9. Перелік доступних команд утиліти *fdisk*.

Щоб створити новий розділ необхідно виконати наступні дії: (*n*) - "*add a new partition*", *specify the type - primary (p)*, обираємо номер розділу – 1, за замовчуванням виділяється весь вільний об'єм на носії, розмір сектору задаємо – 512. Переглянути створений розділ можна виконавши команду (*p*) — "*print the partition table*". Далі змінюємо тип розділу командою (*t*) - "*change a partition's system id*" на (*8e*) - "*Linux LVM type*". В кінці зберігаємо зроблені зміни командою (*w*) - "*write table to disk and exit*" (рис. 3.10).

```
root@stor1:~
14 Hidden FAT16 <3 61 SpeedStor ab Darwin boot f2 DOS secondary
16 Hidden FAT16 63 GNU HURD or Sys af HFS / HFS+ fb VMware VMFS
17 Hidden HPFS/NTF 64 Novell Netware b7 BSDI fs fc VMware VMKCORE
18 AST SmartSleep 65 Novell Netware b8 BSDI swap fd Linux raid auto
1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid fe LANstep
1c Hidden W95 FAT3 75 PC/IX be Solaris boot ff BBT
1e Hidden W95 FAT1 80 Old Minix
Hex code (type L to list all codes): 8e
Changed type of partition 'Empty' to 'Linux LVM'

Command (m for help): p

Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x52275e3a

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1                2048     20971519     10484736   8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@stor1 ~]#
```

Рис. 3.10. Створений розділ утилітою *fdisk*.

Наступним кроком починаємо налаштування *LVM*. Для цього командою *pvcreate* ініціалізуємо фізичні томи, щоб вони розпізнавались *LVM* і могли бути використані для створення логічних томів. На рис. 3.11. показано приклад ініціалізації фізичного тому. Переглянути інформацію про нові фізичні томи можна командою *pvdisplay* (рис. 3.12).

```
root@stor1:~
[root@stor1 ~]# pvcreate /dev/vdb1
Physical volume "/dev/vdb1" successfully created.
[root@stor1 ~]# pvcreate /dev/vdc1
Physical volume "/dev/vdc1" successfully created.
[root@stor1 ~]#
```

Рис. 3.11. Приклад ініціалізації фізичного тому *LVM*.

```
root@stor1:~
Allocated PE          7934
PV UUID              VbleAG-MW89-iwHP-jgkP-F9vQ-8WHk-rcdztZ

"/dev/vdb1" is a new physical volume of "<10.00 GiB"
--- NEW Physical volume ---
PV Name              /dev/vdb1
VG Name
PV Size              <10.00 GiB
Allocatable          NO
PE Size              0
Total PE             0
Free PE              0
Allocated PE         0
PV UUID              B0o21u-CdIz-10fu-5Ith-r30a-la4a-95WTH8

"/dev/vdc1" is a new physical volume of "<10.00 GiB"
--- NEW Physical volume ---
PV Name              /dev/vdc1
VG Name
PV Size              <10.00 GiB
Allocatable          NO
PE Size              0
Total PE             0
Free PE              0
Allocated PE         0
PV UUID              ordlYl-pVEj-DcbP-yCww-nhmh-MsRa-mX0QyL

[root@stor1 ~]#
```

Рис. 3.12. Інформацію про нові фізичні томи *LVM*.

Далі потрібно створити нову групу томів (наприклад, *newvg*) із двох фізичних командою *vgcreate* (рис. 3.13).

```
root@stor1:~
[root@stor1 ~]# vgcreate newvg /dev/vdb1 /dev/vdc1
Volume group "newvg" successfully created
[root@stor1 ~]#
```

Рис. 3.13. Приклад створення нову групу томів *newvg* із двох фізичних томів *LVM*.

Переглянути інформацію про нову групу томів можна командою *vgdisplay* (рис. 3.14).

```
root@stor1:~  
PE Size          4.00 MiB  
Total PE        7935  
Alloc PE / Size 7934 / 30.99 GiB  
Free PE / Size  1 / 4.00 MiB  
VG UUID         T4Ud75-vs9G-isrW-Ax03-hikv-vJKO-ytSjt5  
  
--- Volume group ---  
VG Name          newvg  
System ID  
Format          lvm2  
Metadata Areas  2  
Metadata Sequence No 1  
VG Access       read/write  
VG Status       resizable  
MAX LV          0  
Cur LV         0  
Open LV         0  
Max PV          0  
Cur PV         2  
Act PV         2  
VG Size         19.99 GiB  
PE Size         4.00 MiB  
Total PE        5118  
Alloc PE / Size 0 / 0  
Free PE / Size  5118 / 19.99 GiB  
VG UUID         BuI8tR-g2LW-jlpG-jbnv-a2xC-MZDB-Vx2fPx  
  
[root@stor1 ~]#
```

Рис. 3.14. Інформацію про нову групу томів *LVM*.

Останнім кроком є створення логічних томів *LVM* у вже створеній групі томів. Для цього використовується команда *lvcreate*. Для прикладу створимо два логічні томи: *logvol1* розміру 1Гб та *mytestvol* розміром 2Гб у групі томів *newvg* (рис. 3.15). Переглянути інформацію про створені логічні томи у групі *newvg* можна командою *lvdisplay* (рис. 3.16).

```
root@stor1:~  
[root@stor1 ~]# lvcreate -n logvol1 -L 1G newvg  
Logical volume "logvol1" created.  
[root@stor1 ~]# lvcreate -n mytestvol -L 2G newvg  
Logical volume "mytestvol" created.  
[root@stor1 ~]#
```

Рис. 3.15. Приклад створення двох логічних томів *LVM*.

```
root@stor1:~  
[root@stor1 ~]# lvs newvg  
--- Logical volume ---  
LV Path                /dev/newvg/logvol1  
LV Name                logvol1  
VG Name                newvg  
LV UUID                9ZCCuh-PkSN-rtho-iR68-S6Cs-maKH-PfbOQZ  
LV Write Access        read/write  
LV Creation host, time stor1.comsys.kpi.ua, 2018-05-30 21:46:23 +0300  
LV Status              available  
# open                 0  
LV Size                1.00 GiB  
Current LE             256  
Segments               1  
Allocation             inherit  
Read ahead sectors    auto  
- currently set to    8192  
Block device           253:2  
  
--- Logical volume ---  
LV Path                /dev/newvg/mytestvol  
LV Name                mytestvol  
VG Name                newvg  
LV UUID                RFNUQ9-Nqft-xNUU-crpU-vRZg-VWvq-6ngAEO  
LV Write Access        read/write  
LV Creation host, time stor1.comsys.kpi.ua, 2018-05-30 21:47:09 +0300  
LV Status              available  
# open                 0  
LV Size                2.00 GiB  
Current LE             512  
Segments               1  
Allocation             inherit  
Read ahead sectors    auto  
- currently set to    8192  
Block device           253:3  
  
[root@stor1 ~]#
```

Рис. 3.16. Інформацію про створені логічні томи *LVM* у групі *newvg*.

Створити нову файловою систему на логічному томі (наприклад, *ext4*) можна відповідними командами. На рис. 3.17 показано створення файлової системи *ext4* на логічному томі *logvol1*. Аналогічні дії можна зробити і для іншого логічного тому.

```
root@stor1:~  
[root@stor1 ~]# mkfs.ext4 /dev/newvg/logvol1  
mke2fs 1.42.9 (28-Dec-2013)  
Filesystem label=  
OS type: Linux  
Block size=4096 (log=2)  
Fragment size=4096 (log=2)  
Stride=0 blocks, Stripe width=0 blocks  
65536 inodes, 262144 blocks  
13107 blocks (5.00%) reserved for the super user  
First data block=0  
Maximum filesystem blocks=268435456  
8 block groups  
32768 blocks per group, 32768 fragments per group  
8192 inodes per group  
Superblock backups stored on blocks:  
    32768, 98304, 163840, 229376  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (8192 blocks): done  
Writing superblocks and filesystem accounting information: done  
  
[root@stor1 ~]#
```

Рис. 3.17. Приклад створення файлової системи *ext4* на логічному томі *logvol1*.

Для прикладу змонтуємо створені файлові системи на створених логічних томах (рис. 3.18) та запишемо деякі тестові дані (3.19).

```
root@stor1:~  
[root@stor1 ~]# mkdir /mnt/data1  
[root@stor1 ~]# mkdir /mnt/data2  
[root@stor1 ~]# mount /dev/newvg/logvol1 /mnt/data1  
[root@stor1 ~]# mount /dev/newvg/mytestvol /mnt/data2  
[root@stor1 ~]# df -h  
Filesystem                Size      Used Avail Use% Mounted on  
/dev/mapper/centos-root    29G       1.3G   28G   5% /  
devtmpfs                  486M          0  486M   0% /dev  
tmpfs                     497M          0  497M   0% /dev/shm  
tmpfs                     497M       6.6M  490M   2% /run  
tmpfs                     497M          0  497M   0% /sys/fs/cgroup  
/dev/vda1                 1014M     155M   860M  16% /boot  
tmpfs                     100M          0   100M   0% /run/user/0  
/dev/mapper/newvg-logvol1  976M       2.6M   907M   1% /mnt/data1  
/dev/mapper/newvg-mytestvol 2.0G       17M   1.8G   1% /mnt/data2  
[root@stor1 ~]#
```

Рис. 3.18. Монтування файлової системи на логічному томі.

```
root@stor1:~  
[root@stor1 ~]# dd if=/dev/zero of=/mnt/data1/testfile bs=1M count=100  
100+0 records in  
100+0 records out  
104857600 bytes (105 MB) copied, 1.74721 s, 60.0 MB/s  
[root@stor1 ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root 29G  1.3G  28G   5% /  
devtmpfs        486M   0  486M   0% /dev  
tmpfs           497M   0  497M   0% /dev/shm  
tmpfs           497M  6.6M  490M   2% /run  
tmpfs           497M   0  497M   0% /sys/fs/cgroup  
/dev/vda1       1014M 155M  860M  16% /boot  
tmpfs           100M   0  100M   0% /run/user/0  
/dev/mapper/newvg-logvol1 976M 103M  807M  12% /mnt/data1  
/dev/mapper/newvg-mytestvol 2.0G  17M  1.8G   1% /mnt/data2  
[root@stor1 ~]#
```

Рис. 3.19. Запис тестових даних.

Припустимо, що виникла необхідність збільшити або зменшити розмір файлової системи, що знаходиться на логічному томі *LVM*. Засоби *LVM* дозволяють виконати це командою *lvresize* (рис. 3.20, 3.21). При цьому задати новий абсолютний або відносний розмір можна за допомогою параметру *-size*. Параметр *-resizefs* дозволяє виконати автоматичну зміну розміру файлової системи, що міститься у логічному томі.

```
root@stor1:~  
[root@stor1 ~]# lvresize --resizefs --size +1G /dev/newvg/logvol1  
Size of logical volume newvg/logvol1 changed from 1.00 GiB (256 extents) to 2.00 GiB (512 extents).  
Logical volume newvg/logvol1 successfully resized.  
resize2fs 1.42.9 (28-Dec-2013)  
Filesystem at /dev/mapper/newvg-logvol1 is mounted on /mnt/data1; on-line resizing required  
old_desc_blocks = 1, new_desc_blocks = 1  
The filesystem on /dev/mapper/newvg-logvol1 is now 524288 blocks long.  
  
[root@stor1 ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root 29G  1.3G  28G   5% /  
devtmpfs        486M   0  486M   0% /dev  
tmpfs           497M   0  497M   0% /dev/shm  
tmpfs           497M  6.6M  490M   2% /run  
tmpfs           497M   0  497M   0% /sys/fs/cgroup  
/dev/vda1       1014M 155M  860M  16% /boot  
tmpfs           100M   0  100M   0% /run/user/0  
/dev/mapper/newvg-logvol1 2.0G 103M  1.8G   6% /mnt/data1  
/dev/mapper/newvg-mytestvol 2.0G  17M  1.8G   1% /mnt/data2  
[root@stor1 ~]#
```

Рис. 3.20. Збільшення розміру логічного тому *logvol1*.

```
root@stor1:~  
[root@stor1 ~]# lvresize --resizefs --size -1G /dev/newvg/logvol1  
Do you want to unmount "/mnt/data1" ? [Y|n] y  
fsck from util-linux 2.23.2  
/dev/mapper/newvg-logvol1: 12/131072 files (0.0% non-contiguous), 42796/524288 blocks  
resize2fs 1.42.9 (28-Dec-2013)  
Resizing the filesystem on /dev/mapper/newvg-logvol1 to 262144 (4k) blocks.  
The filesystem on /dev/mapper/newvg-logvol1 is now 262144 blocks long.  
  
Size of logical volume newvg/logvol1 changed from 2.00 GiB (512 extents) to 1.00 GiB (256 extents).  
Logical volume newvg/logvol1 successfully resized.  
[root@stor1 ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root 29G  1.3G  28G   5% /  
devtmpfs        486M   0  486M   0% /dev  
tmpfs           497M   0  497M   0% /dev/shm  
tmpfs           497M  6.6M  490M   2% /run  
tmpfs           497M   0  497M   0% /sys/fs/cgroup  
/dev/vda1       1014M  155M  860M  16% /boot  
tmpfs           100M   0  100M   0% /run/user/0  
/dev/mapper/newvg-mytestvol 2.0G   17M  1.8G   1% /mnt/data2  
/dev/mapper/newvg-logvol1 976M  103M  812M  12% /mnt/data1  
[root@stor1 ~]#
```

Рис. 3.21. Зменшення розміру логічного тому *logvol1*.

При цьому всі збережені раніше дані у файльовій системі зберігаються і можна відразу продовжувати використовувати оновлений логічний том.

В кінці можна розмонтувати файлові системи на логічних томах (рис. 3.22).

```
root@stor1:~  
[root@stor1 ~]# umount /mnt/data1  
[root@stor1 ~]# umount /mnt/data2  
[root@stor1 ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root 29G  1.3G  28G   5% /  
devtmpfs        486M   0  486M   0% /dev  
tmpfs           497M   0  497M   0% /dev/shm  
tmpfs           497M  6.6M  490M   2% /run  
tmpfs           497M   0  497M   0% /sys/fs/cgroup  
/dev/vda1       1014M  155M  860M  16% /boot  
tmpfs           100M   0  100M   0% /run/user/0  
[root@stor1 ~]#
```

Рис. 3.22. Приклад розмонтування файлової системи.

Щоб видалити логічні томи, групи томів та фізичні томи *LVM* використовуються команди *lvremove*, *vgremove* та *pvremove* відповідно.

### Розподілена файлова система *GlusterFS*

*GlusterFS* — це масштабована мережева файлова система, що використовується у застосунках із інтенсивним обміном та великим об'ємом



даних, таких як хмарне зберігання та потокова передача медіа. *GlusterFS* — це безкоштовне програмне забезпечення з відкритим програмним кодом, яке може використовувати стандартне апаратне забезпечення. Том — це набір блоків, і більшість операцій файлової системи *gluster* виконуються над томом. Файлова система *Gluster* підтримує різні типи томів залежно від вимог. Деякі томи підходять для збільшення розміру сховища, деякі для підвищення продуктивності, а деякі для обох випадків. Можна створити наступні типи томів:

- *Розподілені* томи розподіляють файли між вузлами кластеру. Ви можете використовувати розподілені томи, де потрібно масштабувати сховище, а надмірність або неважлива, або забезпечується іншими апаратними/програмними рівнями;
- *Репліковані* томи копіюють файли на всі вузли кластеру. Ви можете використовувати репліковані томи в середовищах, де критично важливі висока доступність і висока надійність;
- *Розподілені репліковані* томи розподіляють файли між реплікованими блоками в тому. Ви можете використовувати розподілені репліковані томи в середовищах, де вимога полягає в масштабуванні сховища, а висока надійність є критичною. Розподілені репліковані томи також забезпечують покращену продуктивність читання в більшості середовищ;
- *Розпорошені* томи базуються на кодах стирання, забезпечуючи ефективний захист від збоїв носія даних чи сервера. Він зберігає закодований фрагмент оригінального файлу на кожному вузлі таким чином, що для відновлення вихідного файлу потрібна лише підмножина фрагментів. Кількість блоків, які можуть бути відсутні без втрати доступу до даних, налаштовується адміністратором на час створення тому;
- Для початку налаштування необхідно встановити та запустити сервер *GlusterFS* (пакет *glusterfs-server*). У наступному прикладі будемо створювати кластер із двох вузлів. Треба створити пул із серверів, які будуть працювати у кластері. Після запуску серверу *GlusterFS* на першому сервері, необхідно додати другий сервер у кластер командою *gluster peer probe* (рис. 3.23).



```
root@stor1:~  
[root@stor1 ~]# gluster peer probe stor2.comsys.kpi.ua  
peer probe: success.  
[root@stor1 ~]#
```

Рис. 3.23. Додавання другого серверу сховища даних.

Переглянути список серверів у кластері *GlusterFS* можна командою *gluster pool list* (рис. 3.24).

```
root@stor1:~  
[root@stor1 ~]# gluster pool list  
UUID                               Hostname                            State  
68ce07ba-47b6-45c5-a435-f23e38d4210b stor2.comsys.kpi.ua                 Connected  
03ec6a81-6a18-41a4-8f09-dacc930e7cc6 localhost                            Connected  
[root@stor1 ~]#
```

Рис. 3.24. Список серверів у кластері *GlusterFS*.

Далі на обох серверах необхідно створити директорію де будуть зберігатись файли (наприклад, */storage/gluster/vol0*). Створена директорія буде використовуватись як том *GlusterFS*. Приклад створення реплікованого тому *GlusterFS* показано на рис. 3.25.

```
root@stor1:~  
[root@stor1 ~]# gluster volume create vol0 replica 2 stor1.comsys.kpi.ua:/storage/  
e/gluster/vol0 stor2.comsys.kpi.ua:/storage/gluster/vol0  
Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to avoid th  
is. See: http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.  
Do you still want to continue?  
(y/n) y  
volume create: vol0: success: please start the volume to access data  
[root@stor1 ~]# gluster volume start vol0  
volume start: vol0: success  
[root@stor1 ~]#
```

Рис. 3.25. Приклад створення реплікованого тому *GlusterFS*.

Переглянути інформацію про створений том можна командою *gluster volume info* (рис. 3.26). Переглянути статус можна командою *gluster volume status* (рис. 3.27).

```
root@stor1:~  
[root@stor1 ~]# gluster volume info vol0  
Volume Name: vol0  
Type: Replicate  
Volume ID: e81e66e6-f78f-45a0-8d5f-dbf5cd0eb065  
Status: Started  
Snapshot Count: 0  
Number of Bricks: 1 x 2 = 2  
Transport-type: tcp  
Bricks:  
Brick1: stor1.comsys.kpi.ua:/storage/gluster/vol0  
Brick2: stor2.comsys.kpi.ua:/storage/gluster/vol0  
Options Reconfigured:  
transport.address-family: inet  
nfs.disable: on  
performance.client-io-threads: off  
[root@stor1 ~]#
```

Рис. 3.26. Інформацію про створений том *GlusterFS*.

```
root@stor1:~  
[root@stor1 ~]# gluster volume status vol0 detail  
Status of volume: vol0  
-----  
Brick           : Brick stor1.comsys.kpi.ua:/storage/gluster/vol0  
TCP Port        : 49152  
RDMA Port       : 0  
Online          : Y  
Pid             : 1815  
File System     : ext4  
Device          : /dev/vdb1  
Mount Options   : rw,seclabel,relatime,data=ordered  
Inode Size      : 256  
Disk Space Free : 9.7GB  
Total Disk Space : 9.7GB  
Inode Count     : 655360  
Free Inodes     : 655329  
-----  
Brick           : Brick stor2.comsys.kpi.ua:/storage/gluster/vol0  
TCP Port        : 49152  
RDMA Port       : 0  
Online          : Y  
Pid             : 2545  
File System     : ext4  
Device          : /dev/vdb1
```

Рис. 3.27. Інформацію про статус тому *GlusterFS*.

Підключитись до створеного кластеру можна шляхом встановлення клієнта *GlusterFS* (пакет *glusterfs-client*). Для підключення потрібно змонтувати тому *GlusterFS* (рис. 3.28).

```
root@stor3:~  
[root@stor3 ~]# mkdir -p /mnt/data  
[root@stor3 ~]# mount -t glusterfs stor1.comsys.kpi.ua:/vol0 /mnt/data  
[root@stor3 ~]# df -h  
Filesystem                Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root    29G  1.3G  28G   5% /  
devtmpfs                  485M   0  485M   0% /dev  
tmpfs                      496M   0  496M   0% /dev/shm  
tmpfs                      496M  6.7M  490M   2% /run  
tmpfs                      496M   0  496M   0% /sys/fs/cgroup  
/dev/vda1                  1014M  155M  860M  16% /boot  
tmpfs                      100M   0  100M   0% /run/user/0  
stor1.comsys.kpi.ua:/vol0  9.8G   37M  9.2G   1% /mnt/data  
[root@stor3 ~]#
```

Рис. 3.28. Підключення до тому *GlusterFS*.

Після цього можна копіювати файли у директорію куди була змонтована файлова система тому *GlusterFS*. Для прикладу, завантажимо до неї тестовий файл і перевіримо вміст директорій на обох серверах (рис. 3.29, 3.30).



```
root@stor1:~  
[root@stor1 ~]# ls -lah /storage/gluster/vol0/  
total 101M  
drwxr-xr-x. 3 root root 4.0K Jun  3 00:03 .  
drwxr-xr-x. 4 root root 4.0K Jun  2 23:19 ..  
drw-----. 11 root root 4.0K Jun  3 00:03 .glusterfs  
-rw-r--r--. 2 root root 100M Jun  3 00:03 test1  
[root@stor1 ~]#
```

Рис. 3.29. Вміст директорій на 1-му сервері *GlusterFS*.



```
root@stor2:~  
[root@stor2 ~]# ls -lah /storage/gluster/vol0/  
total 101M  
drwxr-xr-x. 3 root root 4.0K Jun  3 00:03 .  
drwxr-xr-x. 4 root root 4.0K Jun  2 23:19 ..  
drw-----. 11 root root 4.0K Jun  3 00:03 .glusterfs  
-rw-r--r--. 2 root root 100M Jun  3 00:03 test1  
[root@stor2 ~]#
```

Рис. 3.30. Вміст директорій на 2-му сервері *GlusterFS*.

Як бачимо завантажений файл було репліковано на обидва сервери. Також можна зімітувати вихід з ладу одного із серверів (вимкнувши його) і перевірити статус кластеру (рис. 3.31).



```
root@stor2:~  
[root@stor2 ~]# gluster pool list  
UUID                               Hostname      State  
03ec6a81-6a18-41a4-8f09-dacc930e7cc6 stor1         Disconnected  
68ce07ba-47b6-45c5-a435-f23e38d4210b localhost    Connected  
[root@stor2 ~]#
```

Рис. 3.31. Імітація виходу з ладу одного із серверів кластеру.

При цьому якщо знову завантажити до змонтованої директорії новий файл, він має зберегтись на сервері, який залишився працювати. Коли другий сервер буде відновлено (увімкнено), файли мають синхронізуватись між обома серверами.