Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 3

з дисципліни «Побудова Cloud-систем»
на тему «Віртуалізація сховища даних»

Виконав:
студент І курсу ФІОТ
групи ІМ-21мп
Андрейченко Кирило

Перевірив:
Таран В. І.

Київ – 2023

ЗМІСТ

Мета роботи:

Ознайомитися із багаторівневими архітектурами на прикладі кластеру бази даних.

Завдання:

1. створити та налаштувати RAID масив;
2. створити і налаштувати LVM;
3. налаштувати розподілену файлову систему GlusterFS.

Виконання лабораторної роботи:

Для виконання третьої лабораторної роботи було створено три додаткові віртуальні машини: store1, store2 та store3.

Створення та налаштування RAID масиву

Для створення RAID масиву я використав ZFS, виконуючи наступні кроки:

1. Встановив пакети ZFS:
   sudo apt updatesudo apt
   install zfsutils-linux
2. Створив фізичні пристрої (диски або розділи), які будуть використовуватись у моєму RAID-масиві. Нехай це будуть два диски: dev/sdc та /dev/sdb.
3. Створив пул ZFS, використовуючи команду **zpool create**. Наприклад, для створення RAID-1 масиву (дзеркалювання) з двома дисками виконав наступну команду: **sudo zpool create myraid mirror /dev/sdc /dev/sdb**. Якщо необхідно створити інший тип RAID-масиву, такий як RAID-Z або RAID-Z2, необхідно замінити mirror на raidz або raidz2.
4. Переконуюсь, що мій RAID-масив успішно створено, виконавши команду **zpool status** та **zpool list**.

```
kyrylo@store1:~$ sudo zpool status
  pool: myraid
 state: ONLINE
config:

        NAME        STATE     READ WRITE CKSUM
        myraid      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            sdb     ONLINE       0     0     0
            sdc     ONLINE       0     0     0

errors: No known data errors
```

Рисунок 1 – Інформацію про мій пул ZFS та його стан.

```
kyrylo@store1:~$ zpool list
NAME     SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
myraid  24.5G   140K  24.5G        -         -     0%     0%  1.00x    ONLINE  -
```

Рисунок 2 – Виконання команди zpool list.

ZFS (Zettabyte File System) – це високопродуктивна і масштабована файлова система, розроблена компанією Sun Microsystems (тепер частина Oracle). Вона була створена з метою надання надійності, цілісності даних та розширених можливостей зберігання.
Основні особливості ZFS включають:

1. Пули зберігання даних: ZFS об'єднує фізичні пристрої зберігання (наприклад, жорсткі диски) в пули, що дозволяє ефективніше використовувати простір зберігання та забезпечує можливість динамічного розширення пулів.
2. Контроль цілісності даних: ZFS використовує алгоритми контролю цілісності (наприклад, перевірку суми) для виявлення та відновлення пошкоджених даних. Це дозволяє виявляти помилки на рівні файлової системи та запобігати пошкодженню даних.
3. Копії та знімки: ZFS дозволяє створювати миттєві знімки файлової системи, які є точними копіями даних на певний момент часу. Це корисно для резервного копіювання, відновлення даних та створення тестових середовищ.
4. Поділ та об'єднання: ZFS дозволяє розділяти та об'єднувати пули даних без необхідності переконфігурування або перезапуску системи. Це полегшує управління та масштабування сховища даних.
5. Компресія та дедуплікація: ZFS підтримує стискування даних на рівні файлової системи, що може зменшити використання простору зберігання. Він також надає можливість дедуплікації даних, що дозволяє ідентифікувати та видаляти дублюючі блоки даних.

За рахунок своїх переваг та можливостей, ZFS набула широкого поширення у сферах зберігання даних, серверних системах та інших областях, де потрібне надійне та гнучке управління файловими системами.

Після виконання всіх кроків у створений RAID-масив із використанням ZFS в Ubuntu. Я можу почати використовувати його для зберігання та керування даними.

Для перевірки відмовостійкості RAID масиву, виконую наступні кроки:

1. Тимчасово відключаю один з дисків з RAID-масиву. Використовую команду **sudo zpool offline myraid /dev/sdc**
2. Перевіряю стан масиву після вимкнення диска. Використовую **sudo zpool status**, щоб переконатися, що масив продовжує функціонувати в режимі деградації. Він повинен відображати інформацію про деградований стан, що вказує, що один із членів масиву вимкнено.

```
kyrylo@store1:~$ sudo zpool offline myraid /dev/sdc
kyrylo@store1:~$ sudo zpool status
  pool: myraid
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
config:

        NAME        STATE     READ WRITE CKSUM
        myraid      DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            sdb     ONLINE       0     0     0
            sdc     OFFLINE      0     0     0

errors: No known data errors
```

Рисунок 3 – Перевірка стану масиву після вимкнення диска.

3. Створюю тестовий файл та каталог на файловій системі ZFS, виконавши команду **sudo zfs create myraid/test** для створення каталогу "test" у файловій системі ZFS.
4. Виконую операції читання та запису на вибраному файлі та каталозі. Створюю новий файл за допомогою команди echo **"Hello, ZFS!" | sudo tee test_file.**
5. Перевіряю результати операцій читання та запису, використовуючи команду **cat test_file**.

```
kyrylo@store1:/myraid$ echo "Hello, ZFS!" | sudo tee test_file
Hello, ZFS!
kyrylo@store1:/myraid$ cat test_file
Hello, ZFS!
```

Рисунок 4 – Перевірка результату оперції читання.

6. Для відновлення диска в масиві, додаю ще один диск через VirtualBox та використовую команду **sudo zpool replace myraid /dev/sdc /dev/sdd**. Ця команда замінить відключений диск на новий і запустить процес відновлення даних.

```
kyrylo@store1:~$ sudo zpool replace -f myraid /dev/sdc /dev/sdd
kyrylo@store1:~$ zpool status
  pool: myraid
 state: ONLINE
  scan: resilvered 454K in 00:00:00 with 0 errors on Sun May 28 06:04:07 2023
config:

        NAME        STATE     READ WRITE CKSUM
        myraid      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            sdb     ONLINE       0     0     0
            sdd     ONLINE       0     0     0

errors: No known data errors
```

Рисунок 5 – Диск відновлено.

У процесі виконання операцій читання та запису, слід звернути увагу на будь-які помилки, попередження чи непередбачувану поведінку. Якщо всі операції успішно виконуються, і я можу отримати доступ до збережених даних без проблем, це вказує на працездатність та доступність масиву ZFS.

Налаштування ZFS аналогічно LVM

В мене вже є пул ZFS і я хочу налаштувати його використання аналогічно LVM. Мені необхідно створити всередині пула ZFS датасети, які будуть аналогічні логічним томам LVM.

Кроки для налаштування ZFS як аналог LVM:

1. Перевіряю поточний список датасетів у пулі ZFS за допомогою команди: **zfs list**
2. Створюю нові датасети у пулі ZFS, використовуючи таку команду: **zfs create myraid/testDataset**

```
kyrylo@store1:~$ sudo zfs create myraid/testDataset
kyrylo@store1:~$ zfs list
NAME                  USED  AVAIL     REFER  MOUNTPOINT
myraid                290K  23.7G     26.5K  /myraid
myraid/myfilesystem    24K  23.7G       24K  /myraid/myfilesystem
myraid/test            24K  23.7G       24K  /myraid/test
myraid/testDataset     24K  23.7G       24K  /myraid/testDataset
```

Рисунок 6 – Створено нові датасети.

Тепер у мене є датасети у пулі ZFS, які можуть використовуватися аналогічно логічним томам в LVM. Я можу керувати розміром та атрибутами кожного датасету окремо, створювати знімки та виконувати інші операції, подібні до тих, які доступні в LVM.

Проте, варто зауважити, що ZFS має власні особливості та функціональність, тому не всі аспекти LVM будуть повністю відтворені.

Зміну розмір датасету, виконуючи такі кроки:

Для зміни розміру датасету в ZFS можна використовувати команду zfs set з опцією quota або refreservation. Ось як це зробити:

1. Спочатку дізнаюсь поточний розмір датасету за допомогою команди **zfs list**
2. Щоб змінити розмір датасету, використовую команду **zfs set**. Наприклад, якщо я хочу встановити новий ліміт розміру датасету myraid/testDataset у 10 гігабайт, виконую таку команду: **zfs set quota=10G myraid/testDataset.** Замість 10G можна вказувати потрібний розмір, використовуючи суфікси для гігабайт, терабайт і т.д. Якщо необхідно змінити зарезервований простір (reservation) для датасету, використовую опцію «refreservation» замість «quota». Слід пам’ятати, що установка «quota» обмежує доступний простір

усередині датасету, тоді як установка «refreservation» гарантує наявність вказаного простору для датасету та його нащадків.

3. Перевіряю зміни за допомогою команди **zfs list**, щоб переконатися, що розмір датасету змінився.

Необхідно пам’ятати, що зміна розміру датасету може викликати переміщення даних і зажадати певного часу в залежності від обсягу даних та продуктивності системи.

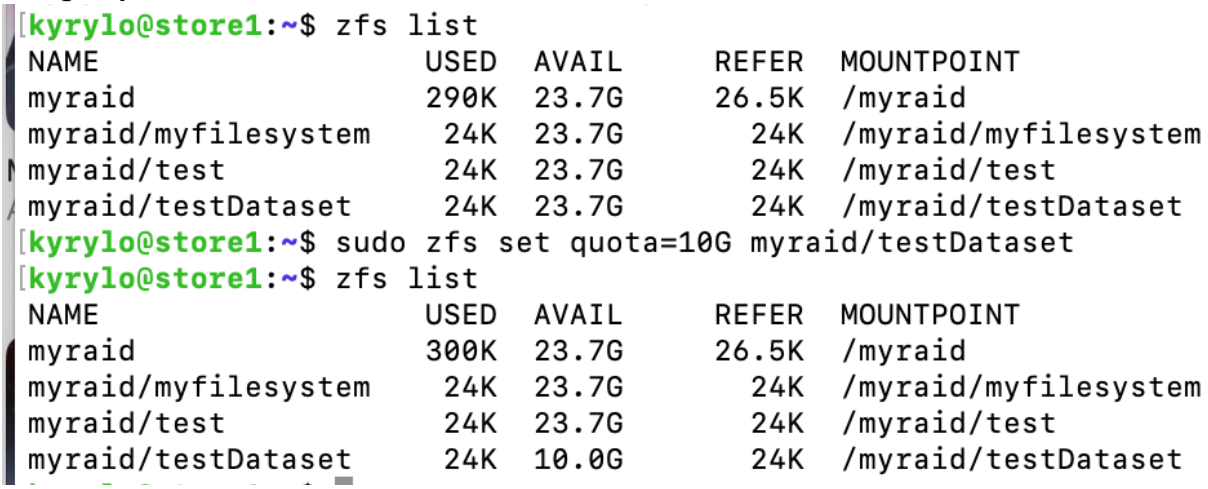

Рисунок 7 – Приклад зміни розміру датасету.

Встановлення та налаштування розподіленої файлової системи Ceph

Ceph – це розподілена файлова система та об'єктне сховище, яке також може надавати блокові пристрої. Вона надає високу стійкість до відмов і масштабованість, а також має безліч можливостей, таких як реплікація, шифрування даних і управління балансуванням навантаження.

Ось чому я обрав саме Ceph, а не Glusterfs:

- Glusterfs менш стабільна і має більше критичних багів, які призводять до пошкодження даних.
- Ceph більш гнучка та функціональна система.
- У Ceph можна додавати диски будь-якого розміру і кожен матиме залежну від розміру вагу. Дані будуть розміщуватись по дисках майже рівномірно. У glusterfs диски додаються парами чи трійками залежно від фактора реплікації. У результаті glusterfs не вдасться просто витягнути старі диски меншого обсягу і поставити нові більшого. Необхідно буде міняти диски одразу у всієї групи, виводячи її з роботи. У Ceph такої проблеми немає, можна спокійно замінювати старі диски на нові більші за обсягом.
- Архітектурно Ceph швидше та надійніше обробляє відмови дисків чи серверів.

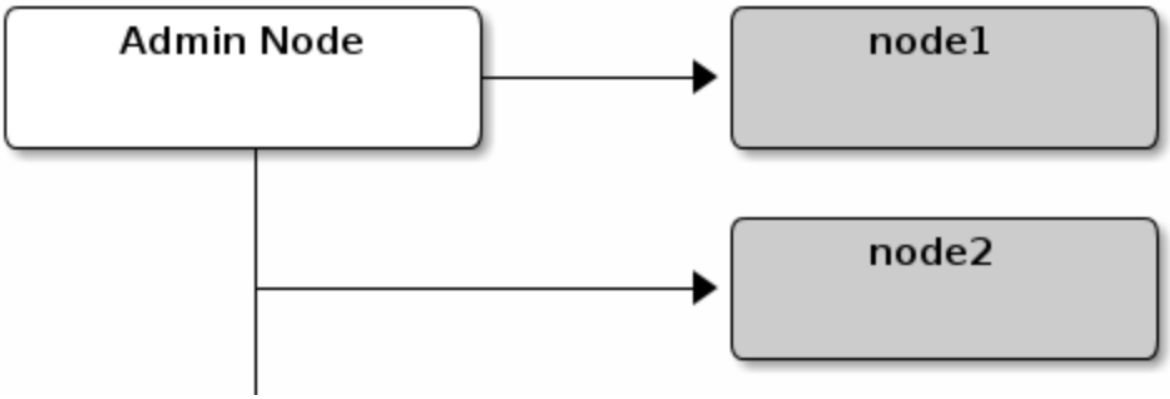

Рисунок 8 – Схема роботи майбутньої розподіленої файлової системи

Встановлення та налаштування Ceph:

1. Store3 буде Admin Node, на ній будуть конфігураційні файли для утиліти ceph-deploy. Для коректної роботи утиліти ceph-deploy всі вузли кластера повинні бути доступні через ssh с Admin node. Я для зручності пропишу в hosts короткі імена для кластера:
   192.168.0.114 store3
   192.168.0.102 store2
   192.168.0.120 store1
2. Встановлення ceph-deploy на кожну node:
   sudo apt update
   sudo apt install python3 python3-pip
   sudo pip3 install ceph-deploy
   ceph-deploy –version

```
root@store3:~# ceph-deploy --version
2.0.1
```

Рисунок 9 – Перевірка встановлення ceph-delpoy.

3. Також, офіційна документація ceph наполягає на встановлені NTP на кожну node, тому так і зроблю:
   apt install ntp
4. Створюю каталог для конфіг файлі та файлів ceph-delpoy:
   mkdir my-cluster
   cd my-cluster
5. Створюю конфіг нового кластеру, при створенні вказую, що в кластері буде три монітори:
   ceph-deploy new store3 store2 store1
6. Налаштовую мережу для кластера, через конфіг файл, який раніше було згенеровано:
   cat ceph.conf

```
root@store3:~/my-cluster# cat ceph.conf
[global]
fsid = 2e0d92b0-e803-475e-9060-0871b63b6e7f
mon_initial_members = store3, store2, store1
mon_host = 192.168.0.114,192.168.0.102,192.168.0.120
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx
```

Рисунок 10 – Налаштування мережі для кластеру.

7. За допомогою ceph-deploy встановлюю всі необхідні пакети ceph на три ноди:
   ceph-deploy install --release mimic store3 store2 store1
8. Після того, як всі пакети встановлені, я створюю та ініціюю монітори кластера. З store3 виконую наступне:
   ceph-deploy mon create-initial
   ceph-deploy admin store3 store2 store1
9. Перевіряю статус кластеру:
   ceph status

```
root@store3:~/my-cluster# ceph status
  cluster:
    id:     2e0d92b0-e803-475e-9060-0871b63b6e7f
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum store3,store2,store1
    mgr: no daemons active
    osd: 0 osds: 0 up, 0 in

  data:
    pools:   0 pools, 0 pgs
    objects: 0  objects, 0 B
    usage:   0 B used, 0 B / 0 B avail
    pgs:
```

10. Записую конфіг на всі хости кластеру:
    ceph-deploy admin store3 store2 store1
11. Створюю MGR:
    ceph-deploy mgr create store3 store2 store1
12. Додаю OSD(тобто, диски):
    ceph-deploy osd create --data /dev/sdb store3
    ceph-deploy osd create --data /dev/sdb store2
    ceph-deploy osd create --data /dev/sdb store1
13. Роблю перевірку, чи все працює, за допомогою команди **ceph status**.

```
root@store3:~/my-cluster# ceph status
  cluster:
    id:     2e0d92b0-e803-475e-9060-0871b63b6e7f
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum store3,store2,store1
    mgr: store3(active)
    osd: 0 osds: 0 up, 0 in
```

Рисунок 11 – Перевірка статусу кластеру

Працездатній кластер налаштовано, лабораторну можна вважати завершеною. Використовуючи команда **ceph osd df** відображаю інформацію про доступні дискові простори в розподіленій системі Ceph.

```
root@store3:~/my-cluster# ceph osd df
ID CLASS WEIGHT  REWEIGHT SIZE    USE     AVAIL   %USE  VAR  PGS
 0   hdd 0.00490  1.00000 5.0 GiB 1.0 GiB 4.0 GiB 20.05 1.00   0
 1   hdd 0.00490  1.00000 5.0 GiB 1.0 GiB 4.0 GiB 20.05 1.00   0
 2   hdd 0.00490  1.00000 5.0 GiB 1.0 GiB 4.0 GiB 20.05 1.00   0
                    TOTAL  15 GiB 3.0 GiB  12 GiB 20.05
```

Рисунок 12 – Інформація про стан дискових ресурсів в мойому Ceph кластері.

Висновок:

Після виконання лабораторної роботи з віртуалізації сховища даних можна зробити наступні висновки:

1. Результати виконання першого завдання, що включало створення та налаштування RAID-масиву, дозволяють підтвердити, що RAID (Redundant Array of Independent Disks) є ефективним методом для забезпечення надійності та високої доступності даних. Шляхом комбінування декількох фізичних дисків в один логічний пристрій, RAID дозволяє досягти резервного копіювання даних, відновлення при випадку відмови диску та підвищення швидкодії доступу до даних.
2. Друге завдання полягало у створенні та налаштуванні ZFS (Zettabyte File System). ZFS є потужною файловою системою, яка надає велику гнучкість та можливості з управління даними. Вона включає в себе функції, такі як детекція та виправлення пошкоджень даних, копіювання з контролем цілісності, засоби для резервного копіювання та реплікації даних. На основі результатів роботи з ZFS можна стверджувати, що ця файлова система є потужним інструментом для управління та захисту даних.

3. Третє завдання передбачало налаштування розподіленої файлової системи Ceph. Ceph забезпечує масштабовану та надійну збереження даних за допомогою розподіленої архітектури. Вона дозволяє об'єднувати різноманітні ресурси зберігання, такі як сервери та диски, у єдину файлову систему. Результати роботи з Ceph підтверджують, що ця технологія дозволяє досягти високої доступності, надійності та швидкодії зберігання даних.

У підсумку, лабораторна робота з віртуалізації сховища даних дозволила ознайомитися з різними багаторівневими архітектурами, такими як RAID, ZFS та Ceph, і підтвердила їх ефективність та значення у забезпеченні надійності, доступності та управління даними. Ці знання можуть бути корисними при розробці та налаштуванні сховищ даних у реальних середовищах.

Отже, після трьох лабораторних робіт, моя схема має наступний вигляд:

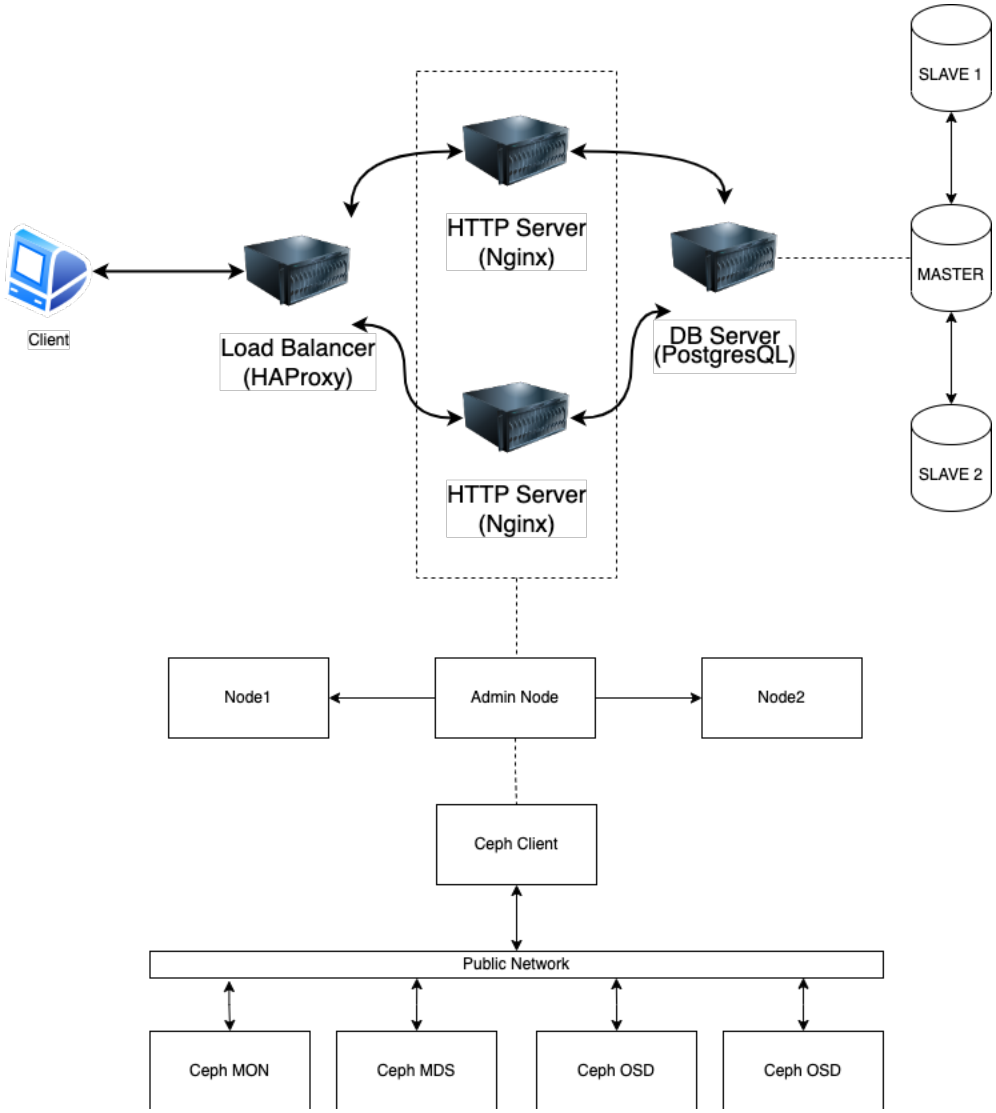

Рисунок 13 – Схема архітектури після виконання трьох лабораторних робіт.