

УДК 004.658.2

DOI: <https://doi.org/10.36994/2788-5518-2024-01-07-12>

ОПТИМІЗАЦІЯ ІНФРАСТРУКТУРИ БАЗ ДАНИХ НА ОСНОВІ AWS I3: ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ТА ЗНИЖЕННЯ ВИТРАТ

Рожков С. М., Інститут комп'ютерних технологій Університету «Україна», Київ, Україна.
<https://orcid.org/0009-0004-2972-7163>, serj.rozhkov@gmail.com

Павленко В. І., к.ф.-м.н., доц., Відкритий міжнародний університет розвитку людини «Україна»,
Інститут комп'ютерних технологій, Київ, Україна. <https://orcid.org/0000-0002-3958-0415>,
pavlenko.v@i.ua

Анотація. У статті розглянуто оптимізацію інфраструктури баз даних шляхом упровадження ефемерних реплік на інстансах AWS i3, що дало змогу підвищити продуктивність та економічну ефективність систем, що працюють із великими обсягами даних. У сучасних умовах бізнесу, який усе більше покладається на аналітичну обробку в режимі реального часу, зниження витрат та підвищення продуктивності інфраструктури стають ключовими завданнями. Інстанси AWS i3, що використовують локальні NVMe SSD-диски, є ідеальним рішенням для високонавантажених баз даних, оскільки забезпечують високу продуктивність вводу-виводу та значно зменшують затримки під час запису даних.

Автори пропонують архітектуру, що включає ефемерні репліки для обробки запитів лише для читання та аналітичних запитів. Також розглянуто впровадження додаткових резервних вузлів для підвищення надійності та відмовостійкості системи. Запропоновані зміни передбачають збереження Master DB без змін, додавання третього вузла бази даних з ефемерним сховищем для перенесення навантаження лише для читання, а також зменшення розміру резервного вузла DR до мінімального робочого рівня.

На основі проведеного дослідження автори демонструють ефективність використання ефемерних реплік у середовищах із високим навантаженням. Використання інстансів i3 дає змогу знизити витрати на інфраструктуру до 30% порівняно зі стандартними схемами та до 75–80% порівняно з RDS (SaaS). Водночас підвищується продуктивність системи, що дає змогу ефективніше обслуговувати запити для читання.

Висновки статті підкреслюють переваги ефемерних реплік у підвищенні продуктивності вводу-виводу, зниженні витрат на інфраструктуру, підвищенні надійності та відмовостійкості, а також у зменшенні навантаження на основний сервер. Автори рекомендують упровадження цього підходу для організацій, які працюють із великими обсягами даних і прагнуть досягти високої продуктивності та економічної ефективності своєї інфраструктури баз даних.

Ключові слова: бази даних, AWS, публічна хмара, безсерверні технології, RDS, EC2, фінансовий сектор.

OPTIMIZING DATABASE INFRASTRUCTURE BASED ON AWS I3: IMPROVING EFFICIENCY AND REDUCING COSTS

Sergii Rozhkov, Institute of Computer Technologies of the University «Ukraine», Kyiv, Ukraine.
<https://orcid.org/0009-0004-2972-7163>, serj.rozhkov@gmail.com

Volodymyr Pavlenko, Ph.D., Ass. Prof., Open University of Human Development «Ukraine»,
Kyiv, Ukraine. <https://orcid.org/0000-0002-3958-0415>, pavlenko.v@i.ua

Abstract. The article discusses the optimization of database infrastructure through the implementation of ephemeral replicas on AWS i3 instances, which enhances the performance and cost efficiency of systems that handle large volumes of data. In today's business environment, which increasingly relies on real-time analytical processing, reducing costs and improving infrastructure performance has become a key priority. AWS i3 instances, utilizing local NVMe SSDs, provide an ideal solution for high-load databases, ensuring high I/O performance and significantly reducing data write latency.

The authors propose an architecture that includes ephemeral replicas for handling read-only and analytical queries. The implementation of additional backup nodes to enhance the reliability and fault tolerance of the system is also discussed. The proposed changes involve maintaining the Master DB unchanged, adding a third database node with ephemeral storage to offload read-only tasks, and reducing the size of the DR backup node to a minimal operational level.

Based on the research conducted, the authors demonstrate the effectiveness of using ephemeral replicas in high-load environments. Utilizing i3 instances can reduce infrastructure costs by up to 30% compared to standard setups and by 75-80% compared to RDS (SaaS). At the same time, system performance is improved, allowing more efficient processing of read queries.

The article's conclusions highlight the advantages of ephemeral replicas in enhancing I/O performance, reducing infrastructure costs, increasing reliability and fault tolerance, and alleviating the load on the primary server. The authors recommend adopting this approach for organizations that work with large volumes of data and seek to achieve high performance and cost efficiency in their database infrastructure.

Key words: databases, AWS, public cloud, serverless services, RDS, EC2, fintech.

Вступ

У сучасних умовах бізнесу, який усе більше покладається на великі обсяги даних і аналітичну обробку в режимі реального часу, забезпечення високої продуктивності та економічної ефективності інфраструктури баз даних стає критично важливим. Одним із рішень, яке може значно покращити ці показники, є використання ефемерних реплік на інстансах AWS i3 [1].

Інстанси i3 в Amazon Web Services (AWS) спеціально розроблені для забезпечення високої продуктивності вводу-виводу завдяки використанню локальних NVMe та SSD дисків. Це робить їх ідеальним вибором для високонавантажених баз даних, які обробляють великі обсяги SQL-запитів та OLAP-запитів. Використання таких інстансів дає змогу не лише знизити затримки під час запису в базу даних, а й значно зменшити витрати на інфраструктуру, що є важливим чинником для багатьох організацій.

Тож розглянемо підходи до оптимізації інфраструктури баз даних на основі AWS i3 інстансів, які спрямовані на підвищення продуктивності та зниження витрат. Зокрема, розглянемо використання ефемерних реплік для обробки запитів лише для читання та аналітичних запитів, а також упровадження додаткових резервних вузлів для підвищення надійності та відмовостійкості системи.

Метою дослідження є оцінка ефективності запропонованих рішень, демонстрація їхнього впливу на продуктивність баз даних та надання рекомендацій щодо впровадження ефемерних реплік у високонавантажених середовищах.

Виконання досліджень

Для економії на витратах EBS і EC2 та забезпечення кращого обслуговування SQL-запитів лише для читання (англ. online analytical processing (OLAP), аналітична обробка у реальному часі), високонавантажені бази даних (приміром, для застосунків, що використовуються для торгівлі на ринках цінних паперів, і т. д.) та торгівлі гарним тоном є використання реплік для читання (read only replica), що зменшує навантаження на основний сервер (primary), що дає змогу зменшити затримки під час запису в базу даних [3].

Інстанси i3 в AWS використовують локальні NVMe- та SSD-диски, які забезпечують високу продуктивність вводу-виводу. Параметри продуктивності для дисків на інстансах i3 можуть варіюватися залежно від конкретної моделі інстансу. Ось основна інформація про продуктивність вводу-виводу (IOPS) для деяких інстансів i3[1].

Характеристики дисків NVMe SSD на інстансах i3

i3.large:

- NVMe SSD обсягом 475 ГБ;
- до 100 000 IOPS;

i3.xlarge:

- NVMe SSD обсягом 950 ГБ;
- до 200 000 IOPS.

Розглянемо стандартну архітектуру (рис. 1) [2]:

- 2 вузли бази даних (primary + standby);
- 1 резервний вузол (для BCP та DR).

Вузли бази даних однієї платформи асинхронно реплікуються між двома зонами доступності AWS, щоб забезпечити можливість відновлення після катастроф (DR) для наших систем [4]. Ми також використовуємо репліку DR як джерело лише для читання для SAS та подібних систем.

Запропоновані зміни:

- master DB залишається без змін;
- додавання третього вузла бази даних із родини i3 з «ефемерним» сховищем (інстанс має додатковий розділ диску, який є дуже швидким, бо працює на локальних NVMe-, SSD-дисках, але за рестарту такого типу інстансу дані не зберігаються);
- перенесення навантаження лише для читання (RO) на цей «ефемерний» резервний вузол;
- зменшення розміру резервного вузла DR до мінімально робочого.

Ці зміни дадуть змогу зробити вузол для читання значно продуктивнішим, оскільки OLAP-запити є дуже інтенсивними щодо вводу-виводу та зменшать загальні витрати на інфраструктуру. Упровадження четвертого вузла бази даних на іншому інстансі i3 в іншій зоні доступності (AZ) додатково підвищує надійність та відмовостійкість системи.

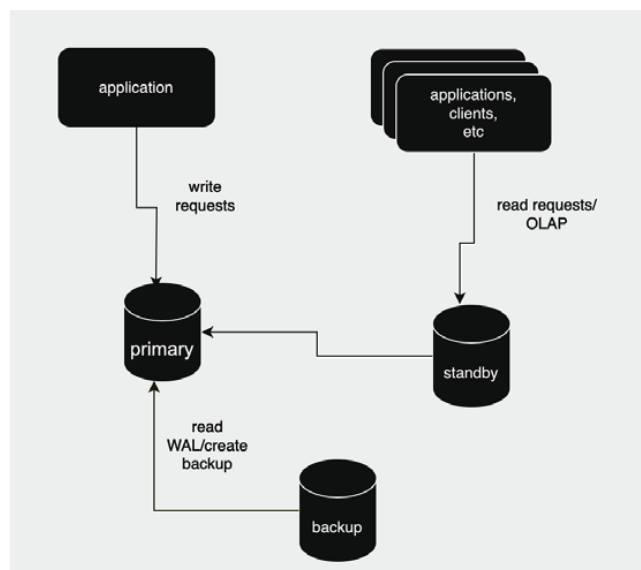


Рис. 1 Стандартна архітектура бази даних

Загальний огляд.

Варіант 1 (рис. 2):

- Master DB (primary): велика інстанція m6 із великою потужністю та багатьма PIOPS EBS-томами;
- Replica DR (standby): мала інстанція m6, без автоматичного переходу, перемикання здійснюється вручну;
- Ephemeral replica: два інстанси сімейства i3 в різних зонах.

Ефемерні репліки знаходяться у кластері Keepalived, який перевіряє працездатність служби Postgres і здійснює перехід у разі непрацездатності. Ефемерні репліки ніколи не повинні бути підвищені до основних. Перехід перемикає DNS псевдонім standby.replica на іншу репліку.

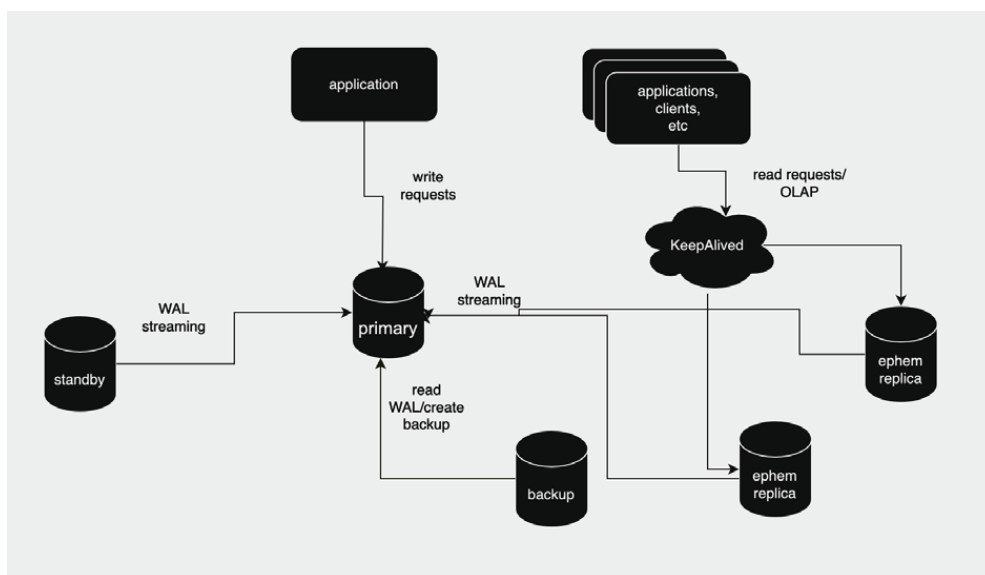


Рис. 2. Архітектура з двома ефемерним репліками

Варіант 2 (рис. 3):

- Master DB (primary): велика інстанція m6 із великою потужністю та багатьма PIOPS EBS-томами;
- Replica DR (standby): мала інстанція m6, без автоматичного переходу, перемикання здійснюється вручну;
- Ephemeral replica: один інстанс сімейства i3 у тій самій зоні.

Replica DR та Ephemeral replica знаходяться у кластері Keepalived, який перевіряє працездатність служби Postgres на Ephemeral replica і здійснює перехід на standby у разі непрацездатності [5]. Перехід перемикає DNS псевдонім standby.replica із Replica DR на Ephemeral replica.

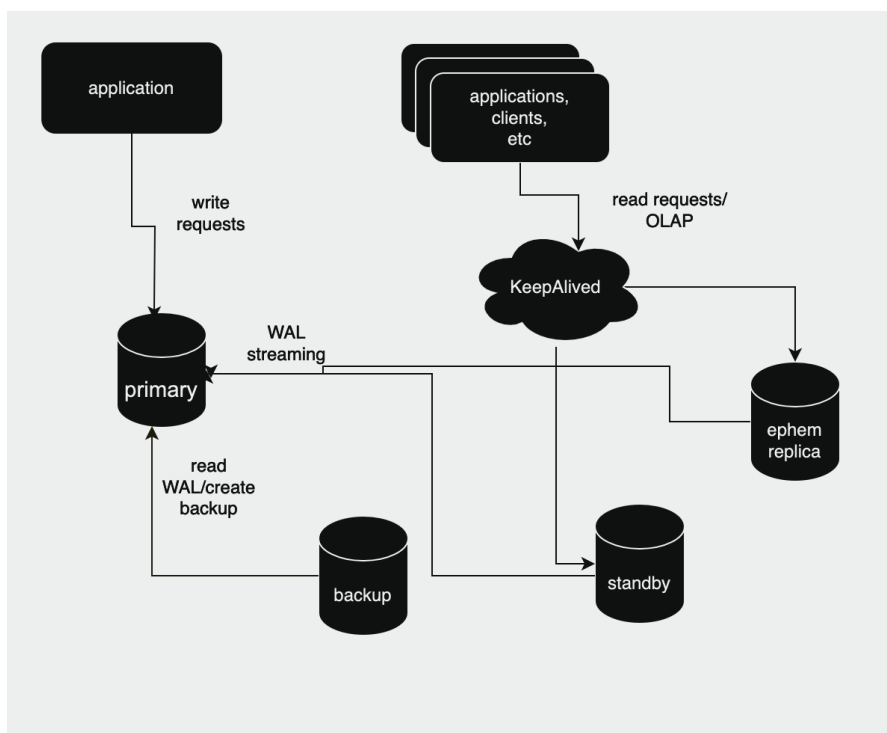


Рис. 3. Архітектура з одною ефемерною реплікою

Процедури переходу реплік:

- Автоматичні.
- Сповідання для перевірки інженером.
- Репліка завантажується з іншої репліки.
- Репліка реплікується з поточного первинного вузла.

У разі рестарту обох реплік їх потрібно буде завантажити з первинної або резервної репліки. Звичайне завантаження можна здійснити автоматично через crontab, використовуючи ansible:

```
[root@i3.db03 ~]# crontab -l | grep reboot
@reboot ansible-playbook -b restart-setup 1>/root/ansible.log 2>&1
Playbook викликає ролі максимально автоматизовано:
- hosts: localhost roles: - postgres_ephemeral_ro_standby_setup.
```

Нижче приведено Ansible-роль, за допомогою якої можна синхронізувати ефемерну репліку після рестарту:

- name: Create partitions
 parted:
 device: «{{ device }}»
 label: gpt
 number: 1
 name: «{{ systemname }}»
 state: present
- name: Create filesystems
 filesystem:
 fstype: «{{ fstype }}»
 dev: «{{ device }}p1»
 opts: «-L {{ fslabel }} -E {{ fsctopts }}»
- name: Mount filesystems (paths are created automatically)
 mount:

```

path: «/db{{ systemname }}»
src: «LABEL={{ fslabel }}»
fstype: «{{ fstype }}»
opts: «{{ fstabopts }}»
dump: 0
passno: 2
state: mounted

```

PG part

- name: Create cluster directories and set owner

file:

```

path: «{{ item }}»
state: directory
owner: «{{ pg_user }}»
group: «{{ pg_group }}»
mode: 0700

```

with_items:

```

- «{{ pg_data_dir }}»
- «{{ pg_tbs_dir }}»

```

- name: Make sure scripts directory exists

file:

```

path: «{{ pg_scripts_dir }}»
state: directory
owner: «{{ pg_user }}»
group: «{{ pg_group }}»

```

- name: Copy primary cluster with pg_basebackup

command: '{{ pg_bin_dir }}/pg_basebackup -h {{ pg_bootstrap_hostname }} -p {{ pg_port|default(5432) }}

}} -D {{ pg_data_dir }} {{ pg_basebackup_options }}'

args:

```

creates: «{{ pg_data_dir }}/pg_xlog»
become_user: «{{ pg_user }}»
register: copy_primary

```

- name: Remove primary serverlog

file:

```

path: «{{ pg_data_dir }}/serverlog»
state: absent

```

when: copy_primary.changed

- name: Copy templated recovery.conf

template:

```

src: «{{ role_path }}/templates/recovery.conf.j2»
dest: «{{ pg_data_dir }}/recovery.conf»
owner: «{{ pg_user }}»
group: «{{ pg_group }}»

```

- name: Start postgres cluster

service:

```

name: «postgres-{{ systemname }}»
state: started

```

- name: Chown serverlog

file:

```

path: «{{ pg_data_dir }}/serverlog»
owner: «{{ pg_user }}»
group: «{{ pg_group }}»

```

when: copy_primary.changed

Розрахунок кошторису.

Нижче буде проведено розрахунок вартості вищеписаних варіантів. Розраховувалася вартість EC2 інстанса, а також вартість дисків. Розрахунки зроблено з реальної високонавантаженої бази даних [1].

		CPU	RAM	EBS Tb	hour	day	month	year
primary	m6a.4xlarge	16	64	6	0,7704	18,4896	1154,688	13994,8186
standby	m6a.large	2	8	1,4	0,0963	2,3112	209,336	2537,15232
ephemeral replica db1	i3.xlarge	4	30	0,95	0,344	8,256	247,68	3013,44
ephemeral replica db2	i3.xlarge	4	30	0,95	0,344	8,256	247,68	3013,44
							Total year	22558,8509

Рис. 4. Розрахунок кошторису для варіанта з двома ефемерними репліками

		CPU	RAM	EBS Tb	hour	day	month	year
primary	m6a.4xlarge	16	64	6	0,7704	18,4896	1154,688	13994,8186
standby	m6a.large	2	8	1,4	0,0963	2,3112	209,336	2537,15232
ephemeral replica db1	i3.xlarge	4	30	0,95	0,344	8,256	247,68	3013,44
							Total year	19545,4109

Рис. 5. Розрахунок кошторису для варіанта з однією ефемерною реплікою

		CPU	RAM	EBS Tb	hour	day	month	year
primary	m6a.4xlarge	16	64	6	0,7704	18,4896	1154,688	13994,8186
standby	m6a.4xlarge	16	64	6	0,7704	18,4896	1154,688	13994,8186
							Total year	27989,6371

Рис. 6. Розрахунок кошторису для класичного варіанта

		CPU	RAM	EBS Tb	IOPs	month	year
RDS multicluster	m6gd.xlarge	16	64	1,5 TB	15000	8942,14	107305,68
						Total year	107305,68

Рис. 7. Розрахунок кошторису за використання RDS

Classic	27989	Saving of costs vs Classic	Saving of costs vs RDS
RDS	107305		
1 ephem repl	19545	30,17%	81,79%
2 ephem repl	22558	19,40%	78,98%

Рис. 8. Порівняльна таблиця

Як ми бачимо, це рішення дає змогу оптимізувати витрати на інфраструктуру порівняно зі стандартною схемою на 30%, порівняно з RDS(SaaS) – на 75–80%, що є дуже суттєвим. Також за рахунок дуже швидких дисків можна підвищити продуктивність системи та забезпечити надійне обслуговування запитів для читання.

Висновки

Упровадження ефемерних реплік на інстансах AWS i3 є ефективним рішенням для оптимізації витрат на інфраструктуру та підвищення продуктивності високонавантажених баз даних, що використовуються для аналітичних запитів (OLAP) та обробки SQL-запитів лише для читання. Основні переваги цього підходу включають:

1. Підвищення продуктивності вводу-виводу: використання локальних NVMe- та SSD-дисків на інстансах i3 значно покращує продуктивність вводу-виводу, що особливо важливо для OLAP-запитів, які є дуже інтенсивними щодо вводу-виводу.

2. Зниження витрат: завдяки високій продуктивності ефемерних реплік можна зменшити розмір резервного вузла DR до мінімально необхідного, що знижує загальні витрати на інфраструктуру. Наші розрахунки показують, що це рішення дає змогу знизити витрати на інфраструктуру на 30% порівняно зі стандартною схемою та на 75–80% – порівняно з RDS (SaaS).

3. Підвищення надійності та відмовостійкості: упровадження додаткових ефемерних реплік у різних зонах доступності (AZ) підвищує надійність та відмовостійкість системи. Репліки знаходяться

у кластері Keeralived, який перевіряє працездатність служби Postgres і здійснює перехід у разі непрацездатності, що забезпечує безперервність обслуговування запитів.

4. Зменшення навантаження на основний сервер: перенесення навантаження лише для читання на ефемерні репліки дає змогу зменшити затримки під час запису в базу даних, що сприяє підвищенню продуктивності основного сервера.

Таким чином, упровадження ефемерних реплік на інстансах AWS i3 є доцільним рішенням для забезпечення високої продуктивності та економічної ефективності інфраструктури баз даних, що відповідає сучасним вимогам бізнесу щодо обробки великих обсягів даних у режимі реального часу.

Література

1. Amazon Web Services Documentation: AWS Instance Types. Amazon Web Services, Inc. Available at: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html> Amazon EC2 Instance Storage. Amazon Web Services, Inc. Available at: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>
2. High Performance MySQL: Optimization, Backups, and Replication. Baron Schwartz, Peter Zaitsev, Vadim Tkachenko. O'Reilly Media, 2012. ISBN: 978-1-449-35650-3.
3. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Martin Kleppmann. O'Reilly Media, 2017. ISBN: 978-1-449-37453-8.
4. Database Systems: The Complete Book. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Pearson, 2008. ISBN: 978-0-136-00864-4.
5. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. Regina O. Obe, Leo S. Hsu. O'Reilly Media, 2017. ISBN: 978-1-491-96926-6.

References

1. Amazon Web Services Documentation: AWS Instance Types. Amazon Web Services, Inc. Available at: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html> Amazon EC2 Instance Storage. Amazon Web Services, Inc. Available at: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>
2. High Performance MySQL: Optimization, Backups, and Replication. Baron Schwartz, Peter Zaitsev, Vadim Tkachenko. O'Reilly Media, 2012. ISBN: 978-1-449-35650-3.
3. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Martin Kleppmann. O'Reilly Media, 2017. ISBN: 978-1-449-37453-8.
4. Database Systems: The Complete Book. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Pearson, 2008. ISBN: 978-0-136-00864-4.
5. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. Regina O. Obe, Leo S. Hsu. O'Reilly Media, 2017. ISBN: 978-1-491-96926-6.