

версии	2024	adcm_cluster_adqm_v23.8.2.7	clickhouse-backup-2.4.33-1.x86_64.rpm
size backups & speed		метод BACKUP через клиента clickhouse-client Документация: <a href="https://clickhouse.com/docs/en/operations/backup/#backup-to-a-local-disk">https://clickhouse.com/docs/en/operations/backup/#backup-to-a-local-disk</a> Бэкапы выполнялись в локальную директорию	Clickhouse-backup Документация: <a href="https://github.com/Altinity/clickhouse-backup">https://github.com/Altinity/clickhouse-backup</a> Бэкапы выполнялись на другой хост
пустая база	0 таблиц / 0 строк	FULL 8.0K  real 0m0.035s user0m0.022s sys 0m0.011s	бэкап пустой базы не создается, процесс зависает
-- Создание таблицы CREATE TABLE default.test ( id UInt32, name String, name1 String, name2 String, date Date ) ENGINE = MergeTree() ORDER BY id;  -- Заполнение таблицы данными INSERT INTO default.test (id, name, name1, name2, date) SELECT number as id, concat('name', toString(number)) as name, concat('name1', toString(number+1)) as name1, concat('name2', toString(number+2)) as name2, toDate('2024-02-23') + number as date FROM numbers(200000000);	0 строк	FULL 12K  real 0m0.038s user0m0.024s sys 0m0.012s	FULL 8.0K  real 0m1.067s user0m0.049s sys 0m0.023s
-- Заполнение таблицы данными INSERT INTO default.test (id, name, name1, name2, date) SELECT number as id, concat('name', toString(number)) as name, concat('name1', toString(number+1)) as name1, concat('name2', toString(number+2)) as name2, toDate('2024-02-23') + number as date FROM numbers(200000000);	200млн строк  Elapsed: 245.515 sec.	FULL 1.3G  real 4m22.245s user0m0.035s sys 0m0.090s	FULL 1.6G  real 1m33.652s user0m24.941s sys 0m3.693s
-- Заполнение таблицы данными INSERT INTO default.test (id, name, name1, name2, date) SELECT number as id, concat('name', toString(number)) as name, concat('name1', toString(number+1)) as name1, concat('name2', toString(number+2)) as name2, toDate('2024-02-23') + number as date FROM numbers(100000000);	100млн строк  Elapsed: 90.504 sec	DIFF 635M  real 1m27.484s user0m0.022s sys 0m0.022s	Incremental 795M  real 1m6.624s user0m13.602s sys 0m2.200s
-- Заполнение таблицы данными INSERT INTO default.test (id, name, name1, name2, date) SELECT number as id, concat('name', toString(number)) as name, concat('name1', toString(number+1)) as name1, concat('name2', toString(number+2)) as name2, toDate('2024-02-23') + number as date FROM numbers(50000000);	50млн строк  Elapsed: 45.018 sec	DIFF 947M  real 1m44.800s user0m0.027s sys 0m0.016s	Incremental 593M  real 0m44.273s user0m9.773s sys 0m1.533s
Общая информация	total 350млн строк	1.3G autofull20240228T140103.zip 635M incremental-20240228T141619.zip 947M incremental-20240228T142408.zip 2.8G total	8,0K auto_full_2024-02-28T07-11-44 1,6G auto_full_2024-02-28T08-48-46 795M auto_incremental_2024-02-28T08-58-43 593M auto_incremental_2024-02-28T09-17-47 3,0G total
		<b>Вывод:</b> 1) способ создает локальный полный и дифференциальный бэкап, но не инкриментальный(от последнего инкриментального). 2) название бэкапов имеет значение, иначе вместо дифф бэкала будет создан полный. 3) не работает с ftp и sftp 4) может создавать локальный бэкап в указанную директорию напрямую	<b>Вывод:</b> 1) способ создает локальный только полный бэкап. 2) удаленный полный и инкриментальный бэкап, но не дифференциальный(от последнего полного). 2) название бэкапов не критично. 3) локально не создает инкриментальный бэкап, только полный. 4) может создавать локальный бэкап в указанную директорию путем переноса из /var/lib/clickhouse/backup

версии	2024	adcm_cluster_adqm_v23.8.2.7	clickhouse-backup-2.4.33-1.x86_64.rpm
restore speed		метод RESTORE через клиента clickhouse-client Документация: <a href="https://clickhouse.com/docs/en/operations/backup#backup-to-a-local-disk">https://clickhouse.com/docs/en/operations/backup#backup-to-a-local-disk</a> Бэкапы выполнялись в локальную директорию	Clickhouse-backup RESTORE_REMOTE Документация: <a href="https://github.com/Altinity/clickhouse-backup">https://github.com/Altinity/clickhouse-backup</a> Бэкапы выполнялись на другой хост
FULL	200млн строк	1.3G — рестор с локального диска  real 1m55.178s user0m0.023s sys 0m0.020s <div><div>count()</div><div>200000000</div></div>	1,6G — рестор с другого хоста  real 1m42.094s user0m24.271s sys 0m4.034s <div><div>count()</div><div>200000000</div></div>
-- Заполнение таблицы данными INSERT INTO default.test (id, name, name1, name2, date) SELECT number as id, concat('name', toString(number)) as name, concat('name1', toString(number+1)) as name1, concat('name2', toString(number+2)) as name2, toDate('2024-02-23') + number as date FROM numbers(100000000);	100млн строк Elapsed: 90.504 sec	1.3G + 635M  real 2m54.971s user0m0.028s sys 0m0.018s  Восстановит полный + дифф бэкап. Поэтому если восстанавливать на моетр времени инкриментального бэкапа, то перед этим таблицу надо очистить drop table default.test; Или удалить из таблицы данные, и применить флаг allow_non_empty_tables при восстановлении <div><div>count()</div><div>300000000</div></div>	795M  real 0m52.400s user0m11.837s sys 0m1.771s  Восстановит только инкриментальный бэкап <div><div>count()</div><div>300000000</div></div>
-- Заполнение таблицы данными INSERT INTO default.test (id, name, name1, name2, date) SELECT number as id, concat('name', toString(number)) as name, concat('name1', toString(number+1)) as name1, concat('name2', toString(number+2)) as name2, toDate('2024-02-23') + number as date FROM numbers(500000000);	50млн строк Elapsed: 45.018 sec	1.3G + 947M  real 3m25.542s user0m0.025s sys 0m0.017s  Восстановит полный + последний дифф бэкап. Поэтому если восстанавливать на моетр времени инкриментального бэкапа, то перед этим таблицу надо очистить drop table default.test; Или удалить из таблицы данные, и применить флаг allow_non_empty_tables при восстановлении <div><div>count()</div><div>350000000</div></div>	593M  real 0m29.518s user0m8.694s sys 0m1.207s  Восстановит только инкриментальный бэкап <div><div>count()</div><div>350000000</div></div>
		<b>Вывод:</b> 1) Увеличивает время восстановления из дифференциального бэкапа, для восстановления в эту же таблицу требуется ее удаление и восстановление идет с полного бэкапа. 2) при восстановлении можно брать только последний дифф бэкап, он содержит данные предыдущих	<b>Вывод:</b> 1) позволяет восстанавливать только инкрементальный бэкап без необходимости восстанавливать полный, это сделано для минимизации сетевого трафика

## backup/restore options for ClickHouse

Tool	Description	Configs	Schema	Data	RBAC
Replication	Use ReplicatedMergeTree	✗	✗	✓	✗
ClickHouse Copier	Works with ZooKeeper to cluster data	✗	✗	✓	✗
Altinity clickhouse-backup	Standalone backup utility ClickHouse versions	✓	✓	✓	✓
ClickHouse BACKUP & RESTORE	Built-in SQL operations in ClickHouse (recent versions)	✗	✓	✓	✗

Вывод:  
метод BACKUP через клиента clickhouse-client не бэкапит разрешения и конфигурационные файлы clickhouse, второй поддерживает всё.