ООО «Астра Консалтинг»

Астра ИС МД (Инфраструктурные Сервисы)

Модуль Astra Redis

ДОКУМЕНТАЦИЯ, СОДЕРЖАЩАЯ ИНФОРМАЦИЮ, НЕОБХОДИМУЮ ДЛЯ ЭКСПЛУАТАЦИИ ЭКЗЕМПЛЯРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРЕДОСТАВЛЕННОГО ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРТНОЙ ПРОВЕРКИ

Москва, 2026

СОДЕРЖАНИЕ

1. Введение

Документ содержит описание функциональных характеристик экземпляра программного обеспечения Astra Redis, предоставленного для проведения экспертной проверки.

Astra Redis (Remote Dictionary Server) — это высокопроизводительная база данных типа «ключ-значение» в памяти, которая используется для кэширования, управления сессиями, очередями сообщений и других задач.

2. Аутентификация в Astra Redis

Аутентификация в Astra Redis осуществляется с использованием команды AUTH. При установленном пароле необходимо пройти аутентификацию для выполнения команд.

Подключение к Astra Redis через утилиту redis-cli:
redis-cli -h 158.160.201.100 -p 6379

Аутентификация с использованием пароля:
AUTH password
, где
password – пароль для доступа к Redis.

Пример подключения с аутентификацией:
redis-cli -h 158.160.201.100 -p 6379 -a mypassword

В случае неуспешной аутентификации возвращается ошибка:
(error) WRONGPASS invalid username-password pair

При успешной аутентификации возвращается:
OK

3. Основные команды Astra Redis

3.1. Получение информации о сервере

Для получения подробной информации о сервере Astra Redis используется команда INFO:

INFO
INFO SERVER
INFO CLIENTS
INFO MEMORY
INFO PERSISTENCE
INFO STATS

Пример выполнения команды INFO SERVER:
INFO SERVER

Результат выполнения:
Server
redis_version:7.0.12
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:a3fdef85a1d2d7ef

redis_mode:standalone
os:Linux 5.4.0-131-generic x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:c11-builtin
gcc_version:9.4.0
process_id:1
process_supervised:systemd
run_id:7c8f9e3b5d2a1c4e6f7g8h9i0j1k2l3m4n5o6p7
tcp_port:6379
server_time_usec:1762438968000000
uptime_in_seconds:3600
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:123456
executable:/usr/local/bin/redis-server
config_file:/etc/redis/redis.conf
io_threads_active:0

3.2. Работа со строками (String)

Строки в Astra Redis — это базовый тип данных, максимальный размер строки — 512 МБ.

Установка значения ключа:
SET key value
SET user:1000 "John Doe"
SETEX session:abc 3600 "user_data"

Получение значения ключа:
GET key
GET user:1000

Установка значения только если ключ не существует:
SETNX key value
SETNX newkey "value"

Получение и изменение значения:
GETSET key newvalue
GETSET counter "100"

Проверка существования ключа:
EXISTS key
EXISTS user:1000

Удаление ключа:
DEL key
DEL oldkey

Примеры использования:
SET product:1001 "Laptop"
GET product:1001
EXISTS product:1001

DEL product:1001
GET product:1001

3.3. Работа с хешами (Hash)

Хеши в Astra Redis — это записи с несколькими полями, идеально подходят для хранения объектов.

Установка полей в хеше:
HSET key field value
HSET user:1000 name "John Doe"
HSET user:1000 email "john@example.com"
HMSET user:1000 field1 value1 field2 value2

Получение полей из хеша:
HGET key field
HGET user:1000 name
HMGET user:1000 name email

Получение всех полей и значений:
HGETALL key
HGETALL user:1000

Получение всех полей:
HKEYS key
HKEYS user:1000

Получение всех значений:
HVALS key
HVALS user:1000

Удаление поля из хеша:
HDEL key field
HDEL user:1000 email

Проверка существования поля:
HEXISTS key field
HEXISTS user:1000 email

Примеры использования:
HSET user:2000 name "Jane Smith"
HSET user:2000 age "30"
HGETALL user:2000
HKEYS user:2000

3.4. Работа со списками (List)

Списки в Astra Redis — это упорядоченные коллекции строк, поддерживают добавление в начало и конец.

Добавление элементов в список:
LPUSH key value [value ...]
RPUSH key value [value ...]
LPUSH messages "first message"

RPUSH messages "second message" "third message"

Извлечение элементов из списка:
LPOP key [count]
RPOP key [count]
LPOP messages
RPOP messages

Получение элементов по индексу:
LINDEX key index
LINDEX messages 0

Получение диапазона элементов:
LRANGE key start stop
LRANGE messages 0 -1

Установка значения по индексу:
LSET key index value
LSET messages 1 "updated value"

Удаление элементов:
LREM key count value
LREM messages 1 "value"

Получение длины списка:
LLEN key
LLEN messages

Примеры использования:
LPUSH queue:tasks task1
LPUSH queue:tasks task2
LPUSH queue:tasks task3
LRANGE queue:tasks 0 -1
LLEN queue:tasks
RPOP queue:tasks

3.5. Работа с множествами (Set)

Множества в Astra Redis — это неупорядоченные коллекции уникальных строк.

Добавление элементов в множество:
SADD key member [member ...]
SADD tags "redis" "database" "cache"

Извлечение элементов из множества:
SMEMBERS key
SMEMBERS tags

Проверка принадлежности элемента к множеству:
SISMEMBER key member
SISMEMBER tags "redis"

Удаление элементов из множества:

SREM key member [member ...]
SREM tags "cache"

Получение количества элементов:
SCARD key
SCARD tags

Извлечение случайного элемента:
SRANDMEMBER key [count]
SRANDMEMBER tags

Извлечение и удаление случайного элемента:
SPOP key [count]
SPOP tags

Операции над множествами:
SINTER key [key ...]
SUNION key [key ...]
SDIFF key [key ...]

Примеры использования:
SADD users:online "user1" "user2" "user3"
SADD users:online "user4"
SMEMBERS users:online
SISMEMBER users:online "user2"
SCARD users:online

3.6. Работа с упорядоченными множествами (Sorted Set)

Упорядоченные множества (Sorted Sets) — это коллекции уникальных элементов, отсортированных по счету.

Добавление элементов со счетом:
ZADD key score member [score member ...]
ZADD leaderboard 100 "player1" 85 "player2" 95 "player3"

Получение количества элементов:
ZCARD key
ZCARD leaderboard

Получение элементов по счету:
ZRANGE key start stop [WITHSCORES]
ZRANGE leaderboard 0 -1 WITHSCORES

Получение элементов в обратном порядке:
ZREVRANGE key start stop [WITHSCORES]
ZREVRANGE leaderboard 0 -1 WITHSCORES

Получение счета элемента:
ZSCORE key member
ZSCORE leaderboard "player1"

Удаление элементов:

ZREM key member [member ...]
ZREM leaderboard "player2"

Увеличение счета:
ZINCRBY key increment member
ZINCRBY leaderboard 10 "player3"

Примеры использования:
ZADD rankings 100 "Alice" 95 "Bob" 90 "Charlie"
ZRANK rankings "Bob"
ZREVRANGE rankings 0 -1 WITHSCORES
ZSCORE rankings "Alice"
ZINCRBY rankings 5 "Alice"
ZREVRANGE rankings 0 -1 WITHSCORES

4. Мониторинг и статистика Astra Redis

4.1. Получение метрик сервера

Команда MONITOR позволяет в реальном времени видеть все команды, выполняемые на сервере:

MONITOR

Команда CLIENT LIST показывает список всех подключенных клиентов:

CLIENT LIST
Результат:
id=3 addr=192.168.1.100:54321 fd=8 name= age=120 idle=0 flags=N db=0 sub=0 psub=0 multi=-1 qbuf=0 qbuf-free=32768 obl=0 oll=0 omem=2 events=r cmd=info
id=4 addr=192.168.1.101:54322 fd=9 name= age=60 idle=5 flags=N db=0 sub=0 psub=0 multi=-1 qbuf=0 qbuf-free=32768 obl=0 oll=0 omem=0 events=r cmd=info

Команда INFO STATS показывает статистику сервера:

INFO STATS
Результат:
Stats
total_connections_received:1000
total_commands_processed:50000
instantaneous_ops_per_sec:50
total_net_input_bytes:1024000
total_net_output_bytes:2048000
instantaneous_input_kbps:10.5
instantaneous_output_kbps:20.3
rejected_connections:5
sync_full:0
sync_partial_ok:0
sync_partial_err:0
expired_keys:100
evicted_keys:0
keyspace_hits:45000

keyspace_misses:5000
pubsub_channels:5
pubsub_patterns:10
latest_fork_usec:1000
migrate_cached_sockets:0

4.2. Мониторинг производительности

Команда SLOWLOG GET показывает медленные команды:

SLOWLOG GET 10
Результат:
1) 1) (integer) 1000
   2) (integer) 1625098000
   3) (integer) 15000
   4) 1) "KEYS"
      2) "*"
2) 1) (integer) 999
   2) (integer) 1625097900
   3) (integer) 12000
   4) 1) "LRANGE"
      2) "queue"
      3) "0"
      4) "-1"

Команда SLOWLOG LEN показывает количество медленных команд в логе:

SLOWLOG LEN

Команда INFO MEMORY показывает использование памяти:

INFO MEMORY
Результат:
Memory
used_memory:2097152
used_memory_human:2.00M
used_memory_rss:4194304
used_memory_rss_human:4.00M
used_memory_peak:3145728
used_memory_peak_human:3.00M
used_memory_peak_perc:66.67%
used_memory_overhead:1048576
used_memory_startup:1048576
used_memory_dataset:1048576
used_memory_datastore_perc:50.00%
total_system_memory:8589934592
total_system_memory_human:8.00G
mem_fragmentation_ratio:2.00
mem_fragmentation_bytes:2097152
mem_not_counted_for_evict:0
mem_replication_backlog:0
mem_clients_slaves:0
mem_clients_normal:1048576

mem_aof_buffer:0
mem_allocator:jemalloc-5.1.0
active_defrag_running:0
lazyfree_pending_objects:0

5. Резервное копирование и восстановление

Создание резервной копии (snapshot):

SAVE - создает синхронную резервную копию
BGSAVE - создает асинхронную резервную копию в фоне

Команда BGSAVE:
BGSAVE
Результат:
Background saving started

Проверка последнего сохранения:
LASTSAVE

Восстановление из резервной копии:
1. Остановить Redis сервер: systemctl stop redis
2. Скопировать файл dump.rdb в директорию данных Redis: cp /path/to/dump.rdb /var/lib/redis/
3. Запустить Redis сервер: systemctl start redis

Расположение файла резервной копии можно узнать из команды INFO PERSISTENCE:
INFO PERSISTENCE

Проверка прав доступа к файлам данных:
ls -la /var/lib/redis/
chown redis:redis /var/lib/redis/dump.rdb

6. Настройка персистентности

Redis поддерживает два способа персистентности:

1. RDB (Snapshotting) - периодическое создание снимков базы данных
2. AOF (Append Only File) - логирование всех операций записи

Настройка RDB в конфигурационном файле /etc/redis/redis.conf:

save 900 1   # через 900 сек, если был изменен хотя бы 1 ключ
save 300 10   # через 300 сек, если было изменено хотя бы 10 ключей
save 60 10000   # через 60 сек, если было изменено хотя бы 10000 ключей

Настройка AOF:

appendonly yes
appendfilename "appendonly.aof"
appendfsync everysec

Применение изменений в конфигурации:
redis-cli config rewrite

Проверка текущих настроек персистентности:
INFO PERSISTENCE
Результат:
Persistence
loading:0
rdb_changes_since_last_save:100
rdb_bgsave_in_progress:0
rdb_last_save_time:1625098000
rdb_last_bgsave_status:ok
rdb_last_bgsave_time_sec:1
rdb_saving_process_pid:1234
rdb_compression_enabled:1
rdb_checksum_enabled:1
rdb_created_last_time:1625098000
rdb_last_cow_size:4194304
aof_enabled:1
aof_rewrite_in_progress:0
aof_rewrite_scheduled:0
aof_last_rewrite_time_sec:2
aof_current_rewrite_time_sec:0
aof_last_bgrewrite_status:ok
aof_last_write_status:ok