Первоначальная настройка ПО «Биллинг СБС»

Необходимые инструменты:

- helm
- helm plugin: secrets
- sops
- gpg key
- pg_bouncer

Настройка георезерва:

- Необходимо создать два тенанта в разных ЦОД
- Объединить сети в одну (растянуть).
- Настроить репликацию кластера postgres с помощью patroni на оба сервера (конфиг patroni.yml)

```
scope: billing
name: postgres-1
namespace: /service/

restapi:
  listen: IP:port
  connect_address: IP:port
#  certfile: /etc/ssl/certs/ssl-cert-snakeoil.pem
#  keyfile: /etc/ssl/private/ssl-cert-snakeoil.key
#  authentication:
#    username: username
#    password: password

etcd:
  hosts: IP: port, основной и резервной БД

bootstrap:
  method: initdb
  dcs:
    ttl: 30
```

```
loop_wait: 10
retry_timeout: 10
maximum_lag_on_failover: 1048576
master_start_timeout: 300
synchronous_mode: false
synchronous_mode_strict: false
synchronous_node_count: 1
# standby_cluster:
  # host: 127.0.0.1
  # port: 1111
  # primary_slot_name: patroni
postgresql:
  use_pg_rewind: true
  use_slots: true
  parameters:
    max_connections: 150
    superuser_reserved_connections: 50
    max_locks_per_transaction: 64
    max_prepared_transactions: 0
    huge_pages: try
    shared_buffers: 512MB
    work_mem: 32MB
    maintenance_work_mem: 1600MB
    effective_cache_size: 12GB
    checkpoint_timeout: 30min
    checkpoint_completion_target: 0.9
    min_wal_size: 2GB
    max_wal_size: 4GB
    wal_buffers: 32MB
    default_statistics_target: 1000
    seq_page_cost: 1
    random_page_cost: 1.1
    effective_io_concurrency: 200
    synchronous_commit: on
    autovacuum: on
```

autovacuum_max_workers: 5
autovacuum_vacuum_scale_factor: 0.01
autovacuum_analyze_scale_factor: 0.02
autovacuum_vacuum_cost_limit: 200
autovacuum_vacuum_cost_delay: 20
autovacuum_naptime: 1s
max_files_per_process: 4096
archive_mode: on
archive_timeout: 1800s
archive_command: cd.
wal_level: replica
wal_keep_segments: 130
max_wal_senders: 10
max_replication_slots: 10
hot_standby: on
wal_log_hints: on
shared_preload_libraries: pg_stat_statements,auto_explain
pg_stat_statements.max: 10000
pg_stat_statements.track: all
pg_stat_statements.save: off
auto_explain.log_min_duration: 10s
auto_explain.log_analyze: true
auto_explain.log_buffers: true
auto_explain.log_timing: false
auto_explain.log_triggers: true
auto_explain.log_verbose: true
auto_explain.log_nested_statements: true
track_io_timing: on
log_lock_waits: on
log_temp_files: 0
track_activities: on
track_counts: on
track_functions: all
log_checkpoints: on
logging_collector: on

```
        log_truncate_on_rotation: on
        log_rotation_age: 1d
        log_rotation_size: 0
        log_line_prefix: '%t [%p-%l] %r %q%u@%d '
        log_filename: 'postgresql-%a.log'
        log_directory: /var/log/postgresql

  initdb:  # List options to be passed on to initdb
    - encoding: UTF8
    - locale: en_US.UTF-8
    - data-checksums

  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
    - host replication replicator 127.0.0.1/32 md5
    - host all all 0.0.0.0/0 md5

postgresql:
  listen: IP:port
  connect_address: IP:port
  use_unix_socket: true
  data_dir: /var/lib/postgresql/14/main
  bin_dir: /usr/lib/postgresql/14/bin
  config_dir: /etc/postgresql/14/main
  pgpass: /var/lib/postgresql/.pgpass_patroni
  authentication:
    replication:
      username: replicator
      password: replicator_pass
    superuser:
      username: postgres
      password: postgres_pass
#    rewind:  # Has no effect on postgres 10 and lower
#      username: rewind_user
#      password: rewind_password
  parameters:
```

```
    unix_socket_directories: /var/run/postgresql
    stats_temp_directory: /var/lib/postgresql/pgsql_stats_tmp

  remove_data_directory_on_rewind_failure: false
  remove_data_directory_on_diverged_timelines: false

#  callbacks:
#    on_start:
#    on_stop:
#    on_restart:
#    on_reload:
#    on_role_change:

  create_replica_methods:
    - basebackup
  basebackup:
    max-rate: '100M'
    checkpoint: 'fast'

watchdog:
  mode: off # Allowed values: off, automatic, required
  device: /dev/watchdog
  safety_margin: 5

tags:
  nofailover: false
  noloadbalance: false
  clonefrom: false
  nosync: false
```

Подготовка БД:

- Прокатить init script, который создаёт: схему, базу данных, пользователя и выдаёт ему гранты
- Узнать хеш пароля созданного пользователя

```
select * from pg_shadow;
```

- Указать в настройках pg_bouncer (/etc/pg_bouncer/userlist.conf) пользователя и хеш пароля
- Сделать рестарт службы pg_bouncer

```
# systemctl restart pg_bouncer
```

Подготовка kubernetes:

- Создать два файла yaml: values.yaml & secrets.yaml
- values.yaml содержит в себе необходимые параметры для helm chart в открытом виде
- secrets.yaml хранит в закрытом виде чувствительные переменные (логины, пароли, токены и т.д.)
- Разворачивание на kubernetes осуществляется helm & helm secrets
- Можно обойтись без плагина helm secrets, тогда файл secrets.yaml нужно хранить в открытом виде

```
helm secrets upgrade -i --atomic --timeout 3m --history-max 3 monetization monetization -n billing -f values.yaml -f 'secrets+gpg-import://${pgp_key}?secrets.yaml
```

Подготовка nginx:

Прописать в nginx location /monetization с директивой обратного прокси сервера

```
location /monetization {
    proxy_pass https://IP_KUBERNETES/monetization;
    proxy_set_header Host  $host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto https;
}
```