Postgresql Operator 资产文档

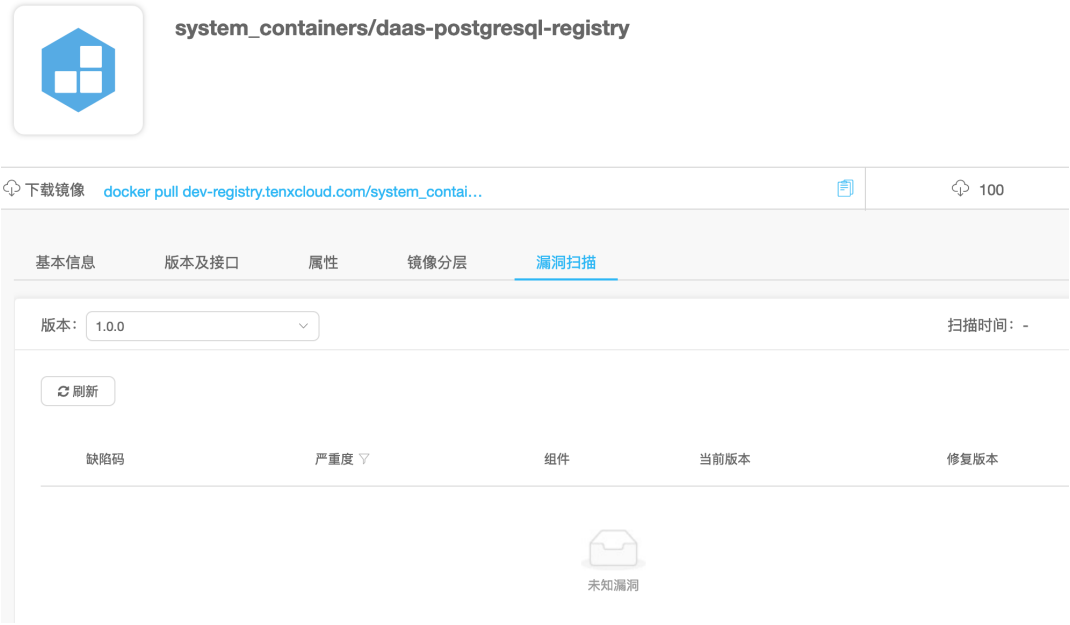
一、资产基本介绍

- **资产简介**

> **PostgreSQL** 是一种非常先进的对象-关系型数据库管理系统（ORDBMS），目前功能最强大，特性最丰富和最先进的自由软件数据库系统。 有些特性甚至连商业数据库都不具备。 这个起源于伯克利（BSD）的数据库研究计划目前已经衍生成一项国际开发项目，并且有非常广泛的用户。
>
> 主要应用场景是：关系型数据库和复杂数据对象处理。
>
> PostgreSQL 主要特征如下：
>
> - **函数**：通过函数，可以在数据库服务器端执行指令程序。
> - **索引**：用户可以自定义索引方法，或使用内置的 B 树，哈希表与 GiST 索引。
> - **触发器**：触发器是由 SQL 语句查询所触发的事件。如：一个 INSERT 语句可能触发一个检查数据完整性的触发器。触发器通常由 INSERT 或 UPDATE 语句触发。 多版本并发控制：PostgreSQL 使用多版本并发控制（MVCC，Multiversion concurrency control）系统进行并发控制，该系统向每个用户提供了一个数据库的"快照"，用户在事务内所作的每个修改，对于其他的用户都不可见，直到该事务成功提交。
> - **规则**：规则（RULE）允许一个查询能被重写，通常用来实现对视图（VIEW）的操作，如插入（INSERT）、更新（UPDATE）、删除（DELETE）。
> - **数据类型**：包括文本、任意精度的数值数组、JSON 数据、枚举类型、XML 数据等。
> - **全文检索**：通过 Tsearch2 或 OpenFTS，8.3 版本中内嵌 Tsearch2。
> - **NoSQL**：JSON，JSONB，XML，HStore 原生支持，至 NoSQL 数据库的外部数据包装器。
> - **数据仓库**：能平滑迁移至同属 PostgreSQL 生态的 GreenPlum，DeepGreen，HAWK 等，使用 FDW 进行 ETL。
>
> PostgreSQL Operator 可以在 Kubernetes 平台上一键部署一个高可用的 PostgreSQL 集群，同时集成时速云公有云 PaaS 平台的运维功能，实现对 PostgreSQL 集群的自动化运维。

- **核心能力**
    - 隐藏了高可用部署的复杂性：提供高可用模式，适合生产环境使用。
    - 通过自动化 Opeartor 生命周期管理简化运维工作：通过内置的 OLM（Operator Lifecycle Management）框架，实现 Operator 的全生命周期自动化管理，大大简化了 Operator 运维工作。
    - 无缝集成时速云公有云 PaaS 平台的存储供给与出口代理能力：创建 PostgreSQL 集群前服务无需提前准备集群存储，可以直接使用云原生应用平台提供的动态存储供给能力；集群创建完成后，可以轻松通过PaaS 平台提供的出口代理功能将集群暴露给外部访问。
    - 常见运维操作完全自动化：支持集群的自动化备份（手动、定时）、恢复、扩容。
    - 企业级安全支持：所有镜像经过安全加固，通过镜像安全扫描。
    - 完善的集群监控支持：支持查看集群的监控、日志、事件、审计信息，同时可以对集群设置告警策略，大大缩短了集群问题的发现、排查时间。
    - 支持跨区高可用。
    - 支持 PostgreSQL 自身性能监控。
- **资产镜像安全扫描结果**

- CatalogSource 镜像：dev-registry.tenxcloud.com/system_containers/daas-postgresql-registry:1.0.0 | 91ea3bf67a9d



- Olm-bundle 镜像：dev-registry.tenxcloud.com/system_containers/postgresql-operator-bundle:1.0.0 | 3abae490094c
- Operator 镜像：dev-registry.tenxcloud.com/system_containers/postgres-operator:v1.5.0-fp1 | 3be7d915fefe
- pg 相关镜像：
  - dev-registry.tenxcloud.com/system_containers/pgadmin4:4.24 | c520f7001785
  - dev-registry.tenxcloud.com/system_containers/spilo-12:1.6-p5 | eccace51ad27
  - dev-registry.tenxcloud.com/system_containers/pgbouncer:master-9 | 9dc85335bda6
  - dev-registry.tenxcloud.com/system_containers/logical-backup:master-58 | 9dc85335bda6

二、应用场景

- **互联网电商**

> PostgreSQL 在互联网应用高并发场景下具有较高稳定性，并且所有操作都可以在 SQL 中完成，无需来回进行数据导入，提高开发效率。

- **位置应用系统**

> PostgreSQL 支持 PostGIS 插件，PostGIS 提供如下空间信息服务功能：空间对象、空间索引、空间操作函数和空间操作符，非常适合位置应用类产品。

- **信息系统**

> PostgreSQL支持更复杂的数据类型，并且能够自定义数据类型。可将不常用的数据转存到 OBS 云存储，节省存储成本和主机空间。

- **金融保险系统**

> PostgreSQL使用多版本并发控制（MVCC），保证数据一致性，主备实例数据同步复制实现数据双保险，确保数据不丢失，并且配合 OBS 实现存储空间扩展，将冷数据转存到 OBS 中，进一步节省历史数据存储成本。

三、资产购买流程（补充注册、充值）

- **在“云市场”中查询需要购买的资产**
- **查看资产详情**
- **购买资产**：在资产详情中点击“购买”
  - 阅读《云原生应用市场用户协议》，同意后勾选“我已阅读并同意...”确认
  - 点击“支付”
- **查看已购资产**：购买资产后会自动跳转到“已购资产”页面显示被购买的资产

四、资产部署指南

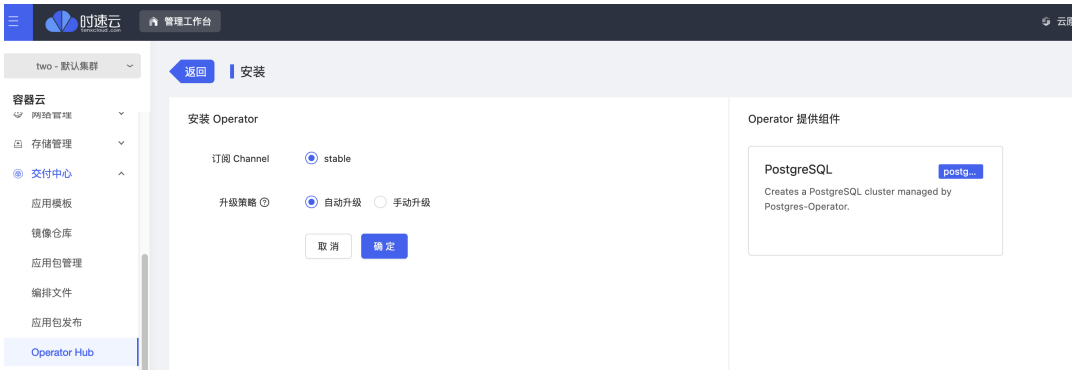
- **添加 PostgreSQL CatalogSource**
  - 目前需要管理员进入后台手动创建 CatalogSource，样例 catsrc.yaml 其中 namespace 按需修改：

```yaml
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: daas-registry-postgresql
  namespace: one
spec:
  displayName: tenxcoud registry
  image: 192.168.2.110/system_containers/daas-postgresql-registry:1.0.0
  publisher: TenxCloud
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

```
## 安装 PostgreSQL CatalogSource
[root@press2 ~]# kubectl apply -f catsrc.yaml

## 查看创建好的 CatalogSource
[root@press2 ~]# kubectl -n one get catsrc | grep PostgreSQL
daas-registry-postgresql        tenxcoud registry   grpc   TenxCloud
47h

## 创建好的 CatalogSource 会自动创建对应的 pod
[root@press2 ~]# kubectl -n one get po | grep postgresql
daas-registry-postgresql-jd6wq
1/1     Running     0          48m
```

- **查询购买的 PostgreSQL Operator**
  - 在容器云--交付中心--Opeartor Hub 里搜索 “ PostgreSQL Cluser Operator”

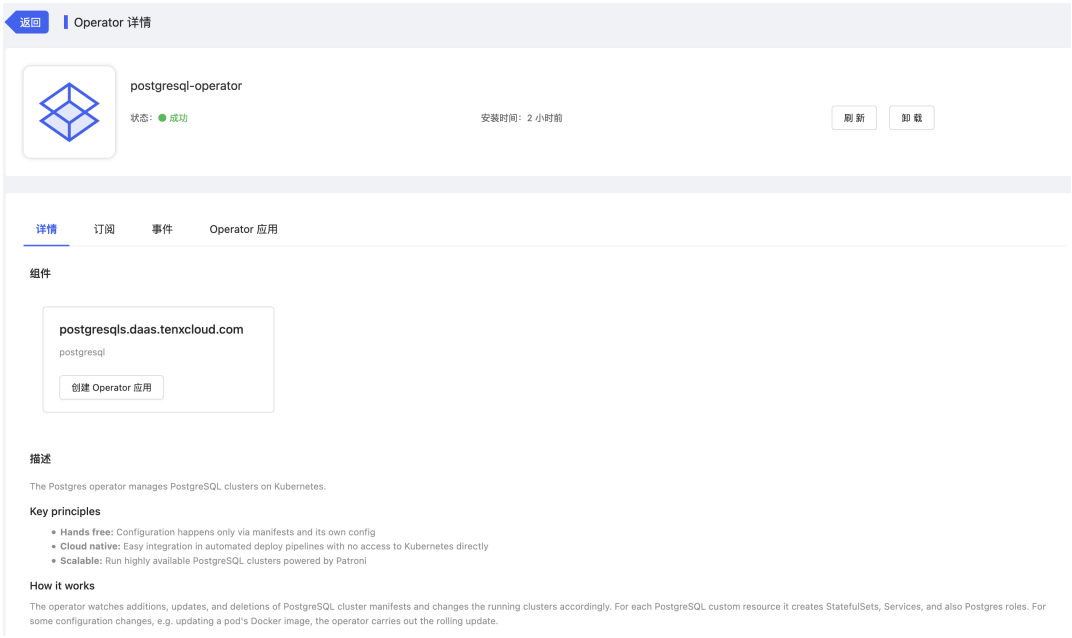
- **安装 PostgreSQL Operator**
  - 点击“安装”



    - 订阅 Channel：一个 Channel 中可以包含一个或多个 Operator 版本
    - 升级策略：后续资产提供商有新 Operator 版本推送到该 Channel 后，有两种升级策略：
      - 自动升级：自动升级到 Channel 中新的版本
      - 手动升级：需要手动确认后，才能升级到 Channel 中新的版本
  - 点击“确定”：自动跳转到已安装 Operator 菜单

  - 等待一段时间，安装状态自动变为“成功”

  - 点击 Operator 名称，进入 Operator 详情



- 查看 Operator 详情

描述

The Postgres operator manages PostgreSQL clusters on Kubernetes.

Key principles

- **Hands free:** Configuration happens only via manifests and its own config
- **Cloud native:** Easy integration in automated deploy pipelines with no access to Kubernetes directly
- **Scalable:** Run highly available PostgreSQL clusters powered by Patroni

How it works

The operator watches additions, updates, and deletions of PostgreSQL cluster manifests and changes the running clusters accordingly. For each PostgreSQL custom resource it creates StatefulSets, Services, and also Postgres roles. For some configuration changes, e.g. updating a pod's Docker image, the operator carries out the rolling update.

Creating a Postgres cluster

After installing the Postgres Operator via OLM you can use the provided YAML examples to create a minimal cluster setup with two instances.

```
# First, make sure the operator is running
kubectl get pod -l name=postgres-operator -n operators

# Then create a new Postgres cluster with a manifest file
kubectl create -n <namespace> -f manifests/minimal-postgres-manifest.yaml

# check for deployed clusters
kubectl get postgresql -n <namespace>
```

The StatefulSets, Services and Pods will be given the same name as you cluster manifest. The database pods can be identified by their number suffix, starting from `-0`. With the default setup they run the Spilo container image by Zalando. As for the services and endpoints, there will be one for the master pod and another one for all the replicas (`-repl` suffix), plus a headless service `-config` for communication with Patroni. Check if all components are coming up. Use the label `application=spilo` to filter and list the label `spilo-role` to see which Pod currently contains the master.

```
# check created database pods
kubectl get pods -l application=spilo -L spilo-role

# check created service resources
kubectl get svc -l application=spilo
```

Configuring the operator

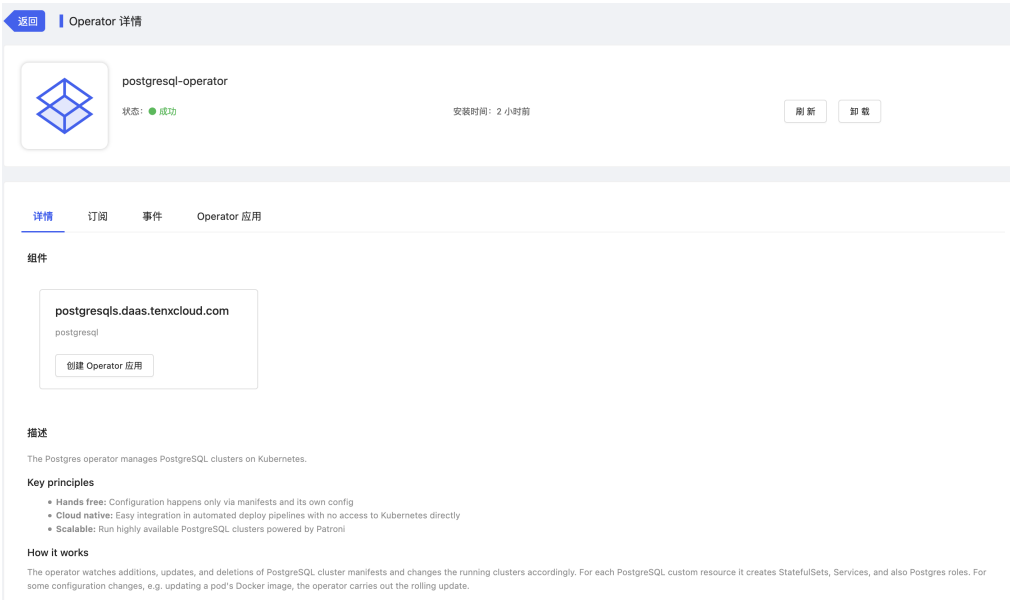
The operator can be configured by defining an `OperatorConfiguration` custom resource or by creating a ConfigMap that contains the parameters to be changed. The YAML example shows the default configuration used internally when creating the operator via OLM. For a detailed description of each parameter checkout out the operator reference

In order to obtain changes from the configuration first create an `OperatorConfiguration` or ConfigMap. Then edit the Postgres Operator CSV and add a reference to this resource in the `spec.containers` section of the `postgres-operator` deployment. Then delete the running operator deployment so that the CSV respawns an updated deployment.

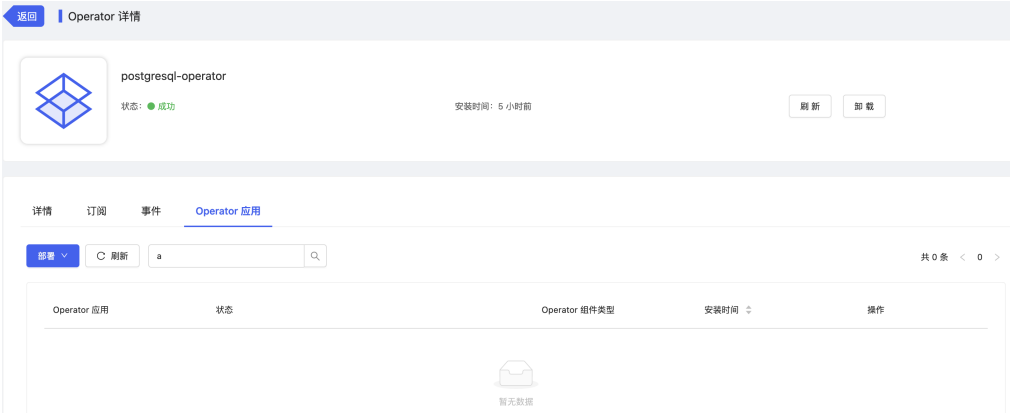
```
# for CRD-based configuration
- env:
  - name: POSTGRES_OPERATOR_CONFIGURATION_OBJECT
    value: postgresql-operator-default-configuration

# for ConfigMap-based configuration
```

- **创建 PostgreSQL 集群**（补充界面部署方式）
  - PostgreSQL Operator 部署完成后，可以从三个地方创建 PostgreSQL 集群：
    - 从 PostgreSQL Operator 列表

- 从 PostgreSQL Operator 详情



- 从 PostgreSQL Operator 详情里的 Operator 应用



- 创建 PostgreSQL 集群
  - 首先必须要先手动创建一个 configmap 配置文件
    - 在“容器云--服务配置--普通配置”中可以"创建配置组"如 one-pg-olm：

- 点击“+配置文件”，填写集群高级配置，填写内容参考如下配置：

- 参考高级配置如下（一般无需修改）：

```
## version
# ------------------------------
# PostgreSQL configuration file
# ------------------------------
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are
introduced with
# "#" anywhere on a line.  The complete list of parameter names
and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the
default values.
# Re-commenting a setting is NOT sufficient to revert it to the
default value;
# you need to reload the server.
#
# This file is read on server startup and when the server
receives a SIGHUP
# signal.  If you edit the file on a running system, you have
to SIGHUP the
# server for the changes to take effect, run "pg_ctl reload",
or execute
# "SELECT pg_reload_conf()".  Some parameters, which are marked
below,
# require a server shutdown and restart to take effect.
#
# Any parameter can also be given as a command-line option to
the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be
changed at run time
```

```
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes        Time units:  ms  =
milliseconds
#                MB = megabytes                     s   =
seconds
#                GB = gigabytes                     min =
minutes
#                TB = terabytes                     h   = hours
#                                                   d   = days




#------------------------------------------------------------------
----------------
# FILE LOCATIONS
#------------------------------------------------------------------
----------------


# The default values of these variables are driven from the -D
command-line
# option or PGDATA environment variable, represented here as
ConfigDir.


#data_directory = 'ConfigDir'       # use data in another
directory
                    # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf' # host-based authentication
file
                    # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration
file
                    # (change requires restart)


# If external_pid_file is not explicitly set, no extra PID file
is written.
#external_pid_file = ''         # write an extra PID file
                    # (change requires restart)




#------------------------------------------------------------------
----------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------
----------------


# - Connection Settings -


#listen_addresses = 'localhost'     # what IP address(es) to
listen on;
```

```
                    # comma-separated list of addresses;
                    # defaults to 'localhost'; use '*' for all
                    # (change requires restart)
#port = 5432                # (change requires restart)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directories = '/var/run/postgresql'    # comma-
separated list of directories
                    # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777     # begin with 0 to use octal
notation
                    # (change requires restart)
#bonjour = off              # advertise server via Bonjour
                    # (change requires restart)
#bonjour_name = ''          # defaults to the computer name
                    # (change requires restart)


# - TCP settings -
# see "man 7 tcp" for details


#tcp_keepalives_idle = 0        # TCP_KEEPIDLE, in seconds;
                    # 0 selects the system default
#tcp_keepalives_interval = 0        # TCP_KEEPINTVL, in
seconds;
                    # 0 selects the system default
#tcp_keepalives_count = 0       # TCP_KEEPCNT;
                    # 0 selects the system default
#tcp_user_timeout = 0           # TCP_USER_TIMEOUT, in
milliseconds;
                    # 0 selects the system default


# - Authentication -


#authentication_timeout = 1min      # 1s-600s
#password_encryption = md5      # md5 or scram-sha-256
#db_user_namespace = off


# GSSAPI using Kerberos
#krb_server_keyfile = ''
#krb_caseins_users = off


# - SSL -


#ssl = off
#ssl_ca_file = ''
#ssl_cert_file = 'server.crt'
#ssl_crl_file = ''
#ssl_key_file = 'server.key'
#ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL' # allowed SSL ciphers
#ssl_prefer_server_ciphers = on
```

```
#ssl_ecdh_curve = 'prime256v1'
#ssl_min_protocol_version = 'TLSv1'
#ssl_max_protocol_version = ''
#ssl_dh_params_file = ''
#ssl_passphrase_command = ''
#ssl_passphrase_command_supports_reload = off




#------------------------------------------------------------------
----------------
# RESOURCE USAGE (except WAL)
#------------------------------------------------------------------
----------------


# - Memory -


shared_buffers = 128MB          # min 128kB
                    # (change requires restart)
#huge_pages = try           # on, off, or try
                    # (change requires restart)
#temp_buffers = 8MB         # min 800kB
#max_prepared_transactions = 0      # zero disables the feature
                    # (change requires restart)
# Caution: it is not advisable to set max_prepared_transactions
nonzero unless
# you actively intend to use prepared transactions.
#work_mem = 4MB             # min 64kB
#maintenance_work_mem = 64MB        # min 1MB
#autovacuum_work_mem = -1       # min 1MB, or -1 to use
maintenance_work_mem
#max_stack_depth = 2MB          # min 100kB
#shared_memory_type = mmap      # the default is the first
option
                    # supported by the operating system:
                    #   mmap
                    #   sysv
                    #   windows
                    # (change requires restart)
dynamic_shared_memory_type = posix  # the default is the first
option
                    # supported by the operating system:
                    #   posix
                    #   sysv
                    #   windows
                    #   mmap
                    # (change requires restart)


# - Disk -


#temp_file_limit = -1           # limits per-process temp file
space
                    # in kB, or -1 for no limit
```

```
# - Kernel Resources -

#max_files_per_process = 1000       # min 25
                    # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0          # 0-100 milliseconds (0
disables)
#vacuum_cost_page_hit = 1       # 0-10000 credits
#vacuum_cost_page_miss = 10     # 0-10000 credits
#vacuum_cost_page_dirty = 20        # 0-10000 credits
#vacuum_cost_limit = 200        # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms         # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100        # max buffers
written/round, 0 disables
#bgwriter_lru_multiplier = 2.0      # 0-10.0 multiplier on
buffers scanned/round
#bgwriter_flush_after = 512kB       # measured in pages, 0
disables

# - Asynchronous Behavior -

#effective_io_concurrency = 1       # 1-1000; 0 disables
prefetching
#max_worker_processes = 8       # (change requires restart)
#max_parallel_maintenance_workers = 2   # taken from
max_parallel_workers
#max_parallel_workers_per_gather = 2    # taken from
max_parallel_workers
#parallel_leader_participation = on
#max_parallel_workers = 8       # maximum number of
max_worker_processes that
                    # can be used in parallel operations
#old_snapshot_threshold = -1        # 1min-60d; -1 disables; 0
is immediate
                    # (change requires restart)
#backend_flush_after = 0        # measured in pages, 0 disables



#------------------------------------------------------------
----------------
# WRITE-AHEAD LOG
```

```
#------------------------------------------------------------------
----------------


# - Settings -


#wal_level = replica            # minimal, replica, or logical
                    # (change requires restart)
#fsync = on              # flush data to disk for crash safety
                    # (turning this off can cause
                    # unrecoverable data corruption)
#synchronous_commit = on        # synchronization level;
                    # off, local, remote_write, remote_apply,
or on
#wal_sync_method = fsync        # the default is the first
option
                    # supported by the operating system:
                    #   open_datasync
                    #   fdatasync (default on Linux)
                    #   fsync
                    #   fsync_writethrough
                    #   open_sync
#full_page_writes = on          # recover from partial page
writes
#wal_compression = off          # enable compression of full-
page writes
#wal_log_hints = off            # also do full page writes of
non-critical updates
                    # (change requires restart)
#wal_init_zero = on         # zero-fill new WAL files
#wal_recycle = on           # recycle WAL files
#wal_buffers = -1           # min 32kB, -1 sets based on
shared_buffers
                    # (change requires restart)
#wal_writer_delay = 200ms       # 1-10000 milliseconds
#wal_writer_flush_after = 1MB       # measured in pages, 0
disables


#commit_delay = 0           # range 0-100000, in microseconds
#commit_siblings = 5            # range 1-1000


# - Checkpoints -


#checkpoint_timeout = 5min      # range 30s-1d
max_wal_size = 1GB
min_wal_size = 80MB
#checkpoint_completion_target = 0.5 # checkpoint target
duration, 0.0 - 1.0
#checkpoint_flush_after = 256kB     # measured in pages, 0
disables
#checkpoint_warning = 30s       # 0 disables


# - Archiving -
```

```
#archive_mode = off     # enables archiving; off, on, or always
                # (change requires restart)
#archive_command = ''       # command to use to archive a
logfile segment
                # placeholders: %p = path of file to archive
                #               %f = file name only
                # e.g. 'test ! -f /mnt/server/archivedir/%f &&
cp %p /mnt/server/archivedir/%f'
#archive_timeout = 0        # force a logfile segment switch
after this
                # number of seconds; 0 disables


# - Archive Recovery -


# These are only used in recovery mode.


#restore_command = ''       # command to use to restore an
archived logfile segment
                # placeholders: %p = path of file to restore
                #               %f = file name only
                # e.g. 'cp /mnt/server/archivedir/%f %p'
                # (change requires restart)
#archive_cleanup_command = ''   # command to execute at every
restartpoint
#recovery_end_command = ''  # command to execute at completion
of recovery


# - Recovery Target -


# Set these only when performing a targeted recovery.


#recovery_target = ''       # 'immediate' to end recovery as
soon as a
                                # consistent state is reached
                # (change requires restart)
#recovery_target_name = ''  # the named restore point to which
recovery will proceed
                # (change requires restart)
#recovery_target_time = ''  # the time stamp up to which
recovery will proceed
                # (change requires restart)
#recovery_target_xid = ''   # the transaction ID up to which
recovery will proceed
                # (change requires restart)
#recovery_target_lsn = ''   # the WAL LSN up to which recovery
will proceed
                # (change requires restart)
#recovery_target_inclusive = on # Specifies whether to stop:
                # just after the specified recovery target (on)
                # just before the recovery target (off)
```

```
                # (change requires restart)
#recovery_target_timeline = 'latest'    # 'current', 'latest',
or timeline ID
                # (change requires restart)
#recovery_target_action = 'pause'   # 'pause', 'promote',
'shutdown'
                # (change requires restart)




#------------------------------------------------------------
----------------
# REPLICATION
#------------------------------------------------------------
----------------


# - Sending Servers -


# Set these on the master and on any standby that will send
replication data.


#max_wal_senders = 10       # max number of walsender processes
                # (change requires restart)
#wal_keep_segments = 0      # in logfile segments; 0 disables
#wal_sender_timeout = 60s   # in milliseconds; 0 disables


#max_replication_slots = 10 # max number of replication slots
                # (change requires restart)
#track_commit_timestamp = off   # collect timestamp of
transaction commit
                # (change requires restart)


# - Master Server -


# These settings are ignored on a standby server.


#synchronous_standby_names = '' # standby servers that provide
sync rep
                # method to choose sync standbys, number of
sync standbys,
                # and comma-separated list of application_name
                # from standby(s); '*' = all
#vacuum_defer_cleanup_age = 0   # number of xacts by which
cleanup is delayed


# - Standby Servers -


# These settings are ignored on a master server.
```

```
#primary_conninfo = ''          # connection string to sending
server
                    # (change requires restart)
#primary_slot_name = ''         # replication slot on sending
server
                    # (change requires restart)
#promote_trigger_file = ''      # file name whose presence ends
recovery
#hot_standby = on           # "off" disallows queries during
recovery
                    # (change requires restart)
#max_standby_archive_delay = 30s    # max delay before
canceling queries
                    # when reading WAL from archive;
                    # -1 allows indefinite delay
#max_standby_streaming_delay = 30s  # max delay before
canceling queries
                    # when reading streaming WAL;
                    # -1 allows indefinite delay
#wal_receiver_status_interval = 10s # send replies at least
this often
                    # 0 disables
#hot_standby_feedback = off     # send info from standby to
prevent
                    # query conflicts
#wal_receiver_timeout = 60s     # time that receiver waits for
                    # communication from master
                    # in milliseconds; 0 disables
#wal_retrieve_retry_interval = 5s   # time to wait before
retrying to
                    # retrieve WAL after a failed attempt
#recovery_min_apply_delay = 0       # minimum delay for
applying changes during recovery


# - Subscribers -


# These settings are ignored on a publisher.


#max_logical_replication_workers = 4    # taken from
max_worker_processes
                    # (change requires restart)
#max_sync_workers_per_subscription = 2  # taken from
max_logical_replication_workers




#------------------------------------------------------------------
----------------
# QUERY TUNING
#------------------------------------------------------------------
----------------
```

```
# - Planner Method Configuration -


#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_indexonlyscan = on
#enable_material = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_parallel_append = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on
#enable_partitionwise_join = off
#enable_partitionwise_aggregate = off
#enable_parallel_hash = on
#enable_partition_pruning = on


# - Planner Cost Constants -


#seq_page_cost = 1.0            # measured on an arbitrary
scale
#random_page_cost = 4.0         # same scale as above
#cpu_tuple_cost = 0.01          # same scale as above
#cpu_index_tuple_cost = 0.005       # same scale as above
#cpu_operator_cost = 0.0025     # same scale as above
#parallel_tuple_cost = 0.1      # same scale as above
#parallel_setup_cost = 1000.0   # same scale as above


#jit_above_cost = 100000        # perform JIT compilation if
available
                    # and query more expensive than this;
                    # -1 disables
#jit_inline_above_cost = 500000     # inline small functions if
query is
                    # more expensive than this; -1 disables
#jit_optimize_above_cost = 500000   # use expensive JIT
optimizations if
                    # query is more expensive than this;
                    # -1 disables


#min_parallel_table_scan_size = 8MB
#min_parallel_index_scan_size = 512kB
#effective_cache_size = 4GB


# - Genetic Query Optimizer -


#geqo = on
#geqo_threshold = 12
```

```
#geqo_effort = 5            # range 1-10
#geqo_pool_size = 0         # selects default based on effort
#geqo_generations = 0           # selects default based on
effort
#geqo_selection_bias = 2.0      # range 1.5-2.0
#geqo_seed = 0.0            # range 0.0-1.0


# - Other Planner Options -


#default_statistics_target = 100    # range 1-10000
#constraint_exclusion = partition   # on, off, or partition
#cursor_tuple_fraction = 0.1        # range 0.0-1.0
#from_collapse_limit = 8
#join_collapse_limit = 8        # 1 disables collapsing of
explicit
                    # JOIN clauses
#force_parallel_mode = off
#jit = on               # allow JIT compilation
#plan_cache_mode = auto         # auto, force_generic_plan or
                    # force_custom_plan




#------------------------------------------------------------------
----------------
# REPORTING AND LOGGING
#------------------------------------------------------------------
----------------


# - Where to Log -


#log_destination = 'stderr'     # Valid values are combinations
of
                    # stderr, csvlog, syslog, and eventlog,
                    # depending on platform.  csvlog
                    # requires logging_collector to be on.


# This is used when logging to stderr:
#logging_collector = off        # Enable capturing of stderr
and csvlog
                    # into log files. Required to be on for
                    # csvlogs.
                    # (change requires restart)


# These are only used if logging_collector is on:
#log_directory = 'log'          # directory where log files are
written,
                    # can be absolute or relative to PGDATA
#log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'    # log file
name pattern,
                    # can include strftime() escapes
```

```
#log_file_mode = 0600           # creation mode for log files,
                    # begin with 0 to use octal notation
#log_truncate_on_rotation = off     # If on, an existing log
file with the
                    # same name as the new log file will be
                    # truncated rather than appended to.
                    # But such truncation only occurs on
                    # time-driven rotation, not on restarts
                    # or size-driven rotation.  Default is
                    # off, meaning append to existing files
                    # in all cases.
#log_rotation_age = 1d          # Automatic rotation of
logfiles will
                    # happen after that time.  0 disables.
#log_rotation_size = 10MB       # Automatic rotation of
logfiles will
                    # happen after that much log output.
                    # 0 disables.


# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'
#syslog_sequence_numbers = on
#syslog_split_messages = on


# This is only relevant when logging to eventlog (win32):
# (change requires restart)
#event_source = 'PostgreSQL'


# - When to Log -


#log_min_messages = warning     # values in order of decreasing
detail:
                    #   debug5
                    #   debug4
                    #   debug3
                    #   debug2
                    #   debug1
                    #   info
                    #   notice
                    #   warning
                    #   error
                    #   log
                    #   fatal
                    #   panic


#log_min_error_statement = error    # values in order of
decreasing detail:
                    #   debug5
                    #   debug4
                    #   debug3
                    #   debug2
                    #   debug1
```

```
                    #   info
                    #   notice
                    #   warning
                    #   error
                    #   log
                    #   fatal
                    #   panic (effectively off)


#log_min_duration_statement = -1    # -1 is disabled, 0 logs
all statements
                    # and their durations, > 0 logs only
                    # statements running at least this number
                    # of milliseconds


#log_transaction_sample_rate = 0.0  # Fraction of transactions
whose statements
                    # are logged regardless of their duration.
1.0 logs all
                    # statements from all transactions, 0.0
never logs.


# - What to Log -


#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_error_verbosity = default      # terse, default, or
verbose messages
#log_hostname = off
#log_line_prefix = '%m [%p] '       # special values:
                    #   %a = application name
                    #   %u = user name
                    #   %d = database name
                    #   %r = remote host and port
                    #   %h = remote host
                    #   %p = process ID
                    #   %t = timestamp without milliseconds
                    #   %m = timestamp with milliseconds
                    #   %n = timestamp with milliseconds (as a
Unix epoch)
                    #   %i = command tag
                    #   %e = SQL state
                    #   %c = session ID
                    #   %l = session line number
                    #   %s = session start timestamp
                    #   %v = virtual transaction ID
                    #   %x = transaction ID (0 if none)
                    #   %q = stop here in non-session
                    #        processes
```

```
                    #   %% = '%'
                    # e.g. '<%u%%%d> '
#log_lock_waits = off           # log lock waits >=
deadlock_timeout
#log_statement = 'none'         # none, ddl, mod, all
#log_replication_commands = off
#log_temp_files = -1            # log temporary files equal or
larger
                    # than the specified size in kilobytes;
                    # -1 disables, 0 logs all temp files
log_timezone = 'Etc/UTC'


#------------------------------------------------------------------
----------------
# PROCESS TITLE
#------------------------------------------------------------------
----------------


#cluster_name = ''          # added to process titles if
nonempty
                    # (change requires restart)
#update_process_title = on




#------------------------------------------------------------------
----------------
# STATISTICS
#------------------------------------------------------------------
----------------


# - Query and Index Statistics Collector -


#track_activities = on
#track_counts = on
#track_io_timing = off
#track_functions = none         # none, pl, all
#track_activity_query_size = 1024   # (change requires restart)
#stats_temp_directory = 'pg_stat_tmp'




# - Monitoring -


#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off
```

```
#------------------------------------------------------------------
----------------
# AUTOVACUUM
#------------------------------------------------------------------
----------------


#autovacuum = on            # Enable autovacuum subprocess?
'on'
                    # requires track_counts to also be on.
#log_autovacuum_min_duration = -1   # -1 disables, 0 logs all
actions and
                    # their durations, > 0 logs only
                    # actions running at least this number
                    # of milliseconds.
#autovacuum_max_workers = 3     # max number of autovacuum
subprocesses
                    # (change requires restart)
#autovacuum_naptime = 1min      # time between autovacuum runs
#autovacuum_vacuum_threshold = 50   # min number of row updates
before
                    # vacuum
#autovacuum_analyze_threshold = 50  # min number of row updates
before
                    # analyze
#autovacuum_vacuum_scale_factor = 0.2   # fraction of table
size before vacuum
#autovacuum_analyze_scale_factor = 0.1  # fraction of table
size before analyze
#autovacuum_freeze_max_age = 200000000  # maximum XID age
before forced vacuum
                    # (change requires restart)
#autovacuum_multixact_freeze_max_age = 400000000    # maximum
multixact age
                    # before forced vacuum
                    # (change requires restart)
#autovacuum_vacuum_cost_delay = 2ms # default vacuum cost delay
for
                    # autovacuum, in milliseconds;
                    # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1  # default vacuum cost limit
for
                    # autovacuum, -1 means use
                    # vacuum_cost_limit




#------------------------------------------------------------------
----------------
# CLIENT CONNECTION DEFAULTS
#------------------------------------------------------------------
----------------


# - Statement Behavior -
```

```
#client_min_messages = notice       # values in order of
decreasing detail:
                    #   debug5
                    #   debug4
                    #   debug3
                    #   debug2
                    #   debug1
                    #   log
                    #   notice
                    #   warning
                    #   error
#search_path = '"$user", public'  # schema names
#row_security = on
#default_tablespace = ''        # a tablespace name, '' uses
the default
#temp_tablespaces = ''          # a list of tablespace names,
'' uses
                    # only default tablespace
#default_table_access_method = 'heap'
#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#default_transaction_deferrable = off
#session_replication_role = 'origin'
#statement_timeout = 0          # in milliseconds, 0 is
disabled
#lock_timeout = 0           # in milliseconds, 0 is disabled
#idle_in_transaction_session_timeout = 0    # in milliseconds,
0 is disabled
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#vacuum_multixact_freeze_min_age = 5000000
#vacuum_multixact_freeze_table_age = 150000000
#vacuum_cleanup_index_scale_factor = 0.1    # fraction of total
number of tuples
                        # before index cleanup, 0 always
performs
                        # index cleanup
#bytea_output = 'hex'           # hex, escape
#xmlbinary = 'base64'
#xmloption = 'content'
#gin_fuzzy_search_limit = 0
#gin_pending_list_limit = 4MB


# - Locale and Formatting -


datestyle = 'iso, mdy'
#intervalstyle = 'postgres'
timezone = 'Etc/UTC'
#timezone_abbreviations = 'Default'     # Select the set of
available time zone
                    # abbreviations.  Currently, there are
                    #   Default
                    #   Australia (historical usage)
                    #   India
```

```
                    # You can create your own file in
                    # share/timezonesets/.
#extra_float_digits = 1         # min -15, max 3; any value >0
actually
                    # selects precise output mode
#client_encoding = sql_ascii        # actually, defaults to
database
                    # encoding


# These settings are initialized by initdb, but they can be
changed.
lc_messages = 'en_US.UTF-8'         # locale for system error
message
                    # strings
lc_monetary = 'en_US.UTF-8'         # locale for monetary
formatting
lc_numeric = 'en_US.UTF-8'          # locale for number
formatting
lc_time = 'en_US.UTF-8'             # locale for time
formatting


# default configuration for text search
default_text_search_config = 'pg_catalog.english'


# - Shared Library Preloading -


#shared_preload_libraries = ''  # (change requires restart)
#local_preload_libraries = ''
#session_preload_libraries = ''
#jit_provider = 'llvmjit'       # JIT library to use


# - Other Defaults -


#dynamic_library_path = '$libdir'




#------------------------------------------------------------------
----------------
# LOCK MANAGEMENT
#------------------------------------------------------------------
----------------


#deadlock_timeout = 1s
#max_locks_per_transaction = 64     # min 10
                    # (change requires restart)
#max_pred_locks_per_transaction = 64    # min 10
                    # (change requires restart)
#max_pred_locks_per_relation = -2   # negative values mean
                    # (max_pred_locks_per_transaction
```

```
                    #  / -max_pred_locks_per_relation) - 1
#max_pred_locks_per_page = 2            # min 0




#------------------------------------------------------------------
----------------
# VERSION AND PLATFORM COMPATIBILITY
#------------------------------------------------------------------
----------------


# - Previous PostgreSQL Versions -


#array_nulls = on
#backslash_quote = safe_encoding    # on, off, or safe_encoding
#escape_string_warning = on
#lo_compat_privileges = off
#operator_precedence_warning = off
#quote_all_identifiers = off
#standard_conforming_strings = on
#synchronize_seqscans = on


# - Other Platforms and Clients -


#transform_null_equals = off




#------------------------------------------------------------------
----------------
# ERROR HANDLING
#------------------------------------------------------------------
----------------


#exit_on_error = off            # terminate session on any
error?
#restart_after_crash = on       # reinitialize after backend
crash?
#data_sync_retry = off          # retry or panic on failure to
fsync
                    # data?
                    # (change requires restart)




#------------------------------------------------------------------
----------------
# CONFIG FILE INCLUDES
#------------------------------------------------------------------
----------------
```

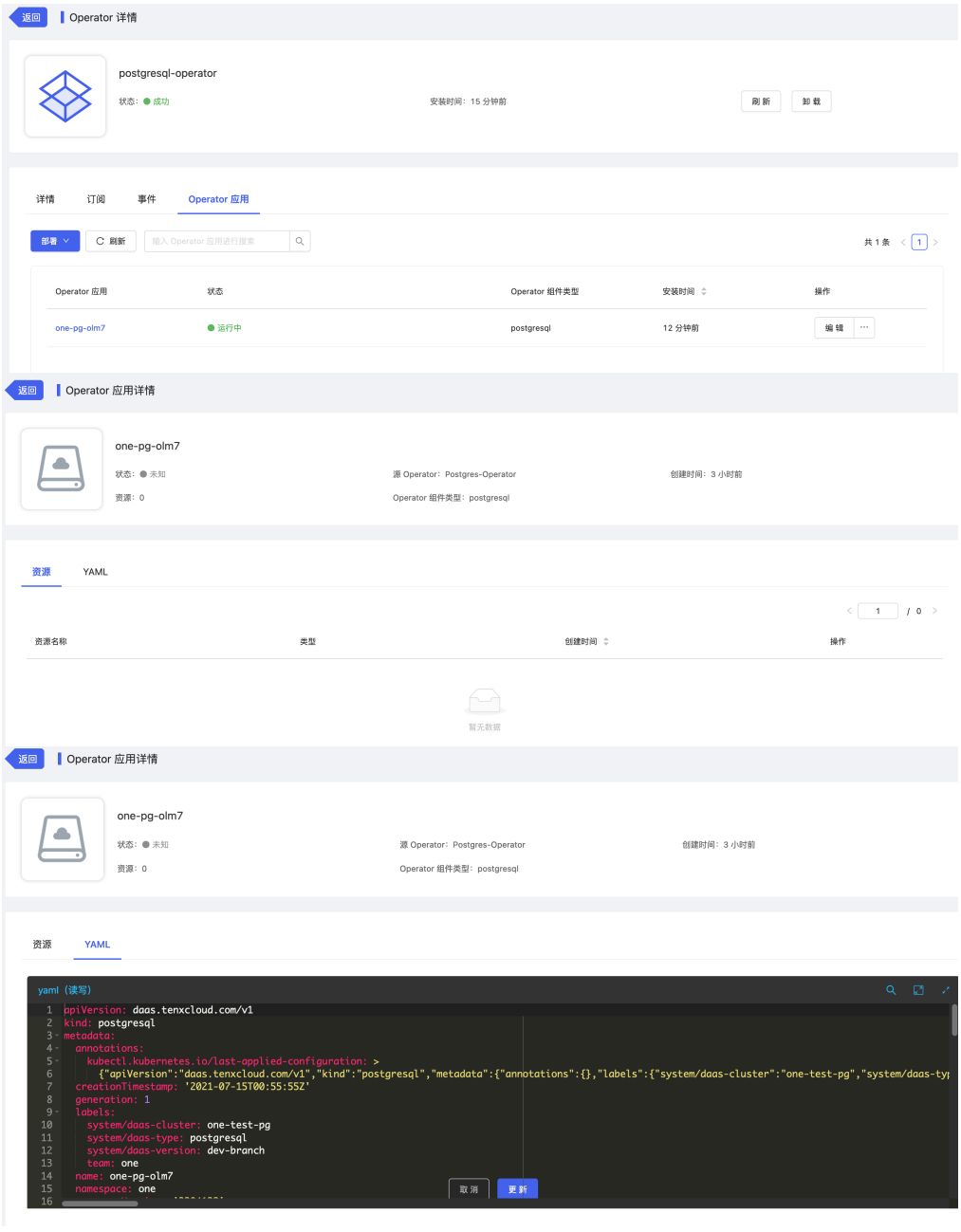
```
# These options allow settings to be loaded from files other
than the
# default postgresql.conf.  Note that these are directives, not
variable
# assignments, so they can usefully be given more than once.


#include_dir = '...'            # include files ending in
'.conf' from
                    # a directory, e.g., 'conf.d'
#include_if_exists = '...'      # include file only if it
exists
#include = '...'            # include file




#------------------------------------------------------------------
----------------
# CUSTOMIZED OPTIONS
#------------------------------------------------------------------
----------------


# Add settings for extensions here
```

- 创建完 configmap 配置文件后，点击部署 Operator 应用

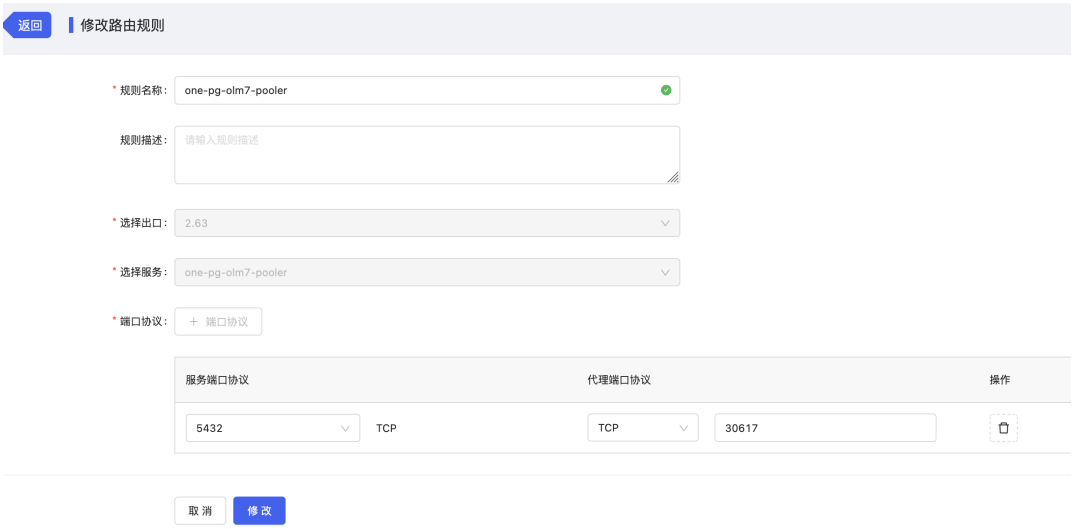
- metadata.labels.team: team 名称
- metadata.name: team 名称 + 自定义集群名称
- metadata.namespace: 命名空间
- spec.additionalVolumes.volumeSource.name: 刚才创建的 cm 名称
- spec.connectionPooler.numberOfInstances：pooler 实例的个数
- spec.connectionPooler.resources：pooler 实例的资源大小配置，推荐至少 500m/300Mi
- spec.numberOfInstances：pg 实例的个数
- spec.password：pg 实例的 password
- spec.postgresql.version: pg 实例的版本，推荐至少 12
- spec.resources：pg 实例的资源大小配置，推荐至少 1C/1024Mi
- spec.superUserPassword：pg 实例的 superUserPassword
- spec.teamId：pg 实例的 teamId，用于区分不同的团队，必填
- spec.volume.size：每个 pg Pod 挂载的存储大小，推荐至少 1G

- spec.volume.storageClass：集群使用的存储类名称，从“容器云--存储管理--存储卷--创建存储卷--存储类下拉列表”中可以查看到可以使用的存储类

- 创建完成后，在“容器云--容器应用--Operator 应用”中可以查看创建的 PostgreSQL 集群，点击集群名字可以查看详情

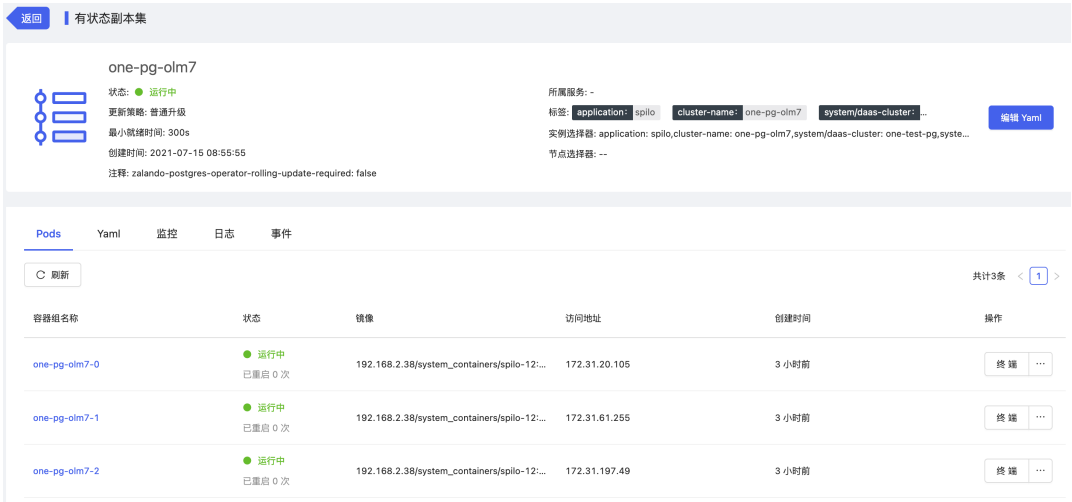


- **配置 PostgreSQL 集群访问**
  - 在“容器云--网络管理--应用路由”页面，点击“添加路由规则”，在“添加路由规则”页面，选择一个服务出口代理 PostgreSQL 集群的 Pooler 服务



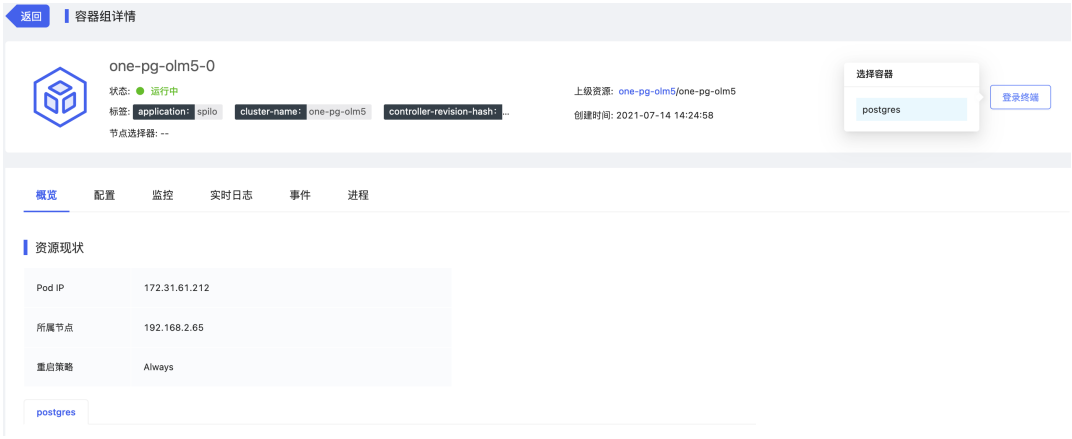
- **验证 PostgreSQL 集群访问**
  - 在“容器云--容器应用--容器服务”列表中，找到被代理的 PostgreSQL 集群的管理 pooler 服务（< PostgreSQL 集群名称>-pooler），点击“查看地址”，点击地址旁边的拷贝图标保存地址信息，用于后面访问验证。

    - 集群内：在 Kuberenetes 集群内访问 PostgreSQL 服务，使用这个地址
    - 服务代理-TCP：在 Kuberenetes 集群外访问 PostgreSQL 服务，使用这个地址
  - 在“容器云--工作负载--有状态副本集”列表中，找到 PostgreSQL 集群副本集，点击进入



  - 在 PostgreSQL 有状态副本集详情中，选择其中一个容器组，点击进入

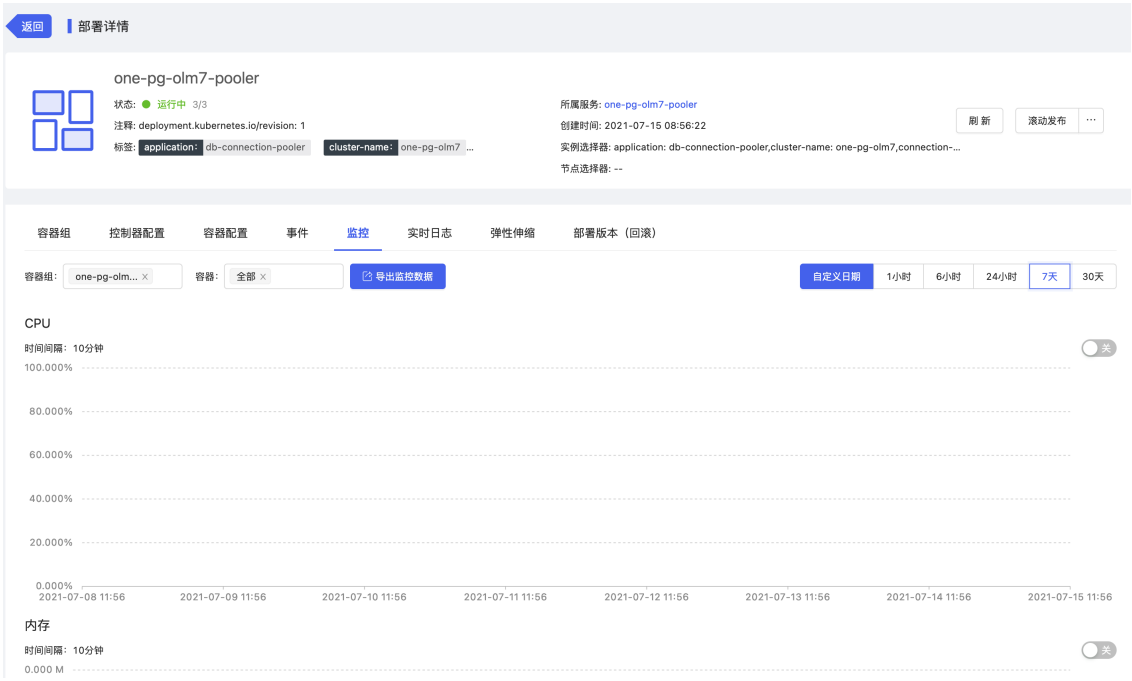
- 在 PostgreSQL 容器组详情中，点击登陆终端里的 PostgreSQL 容器



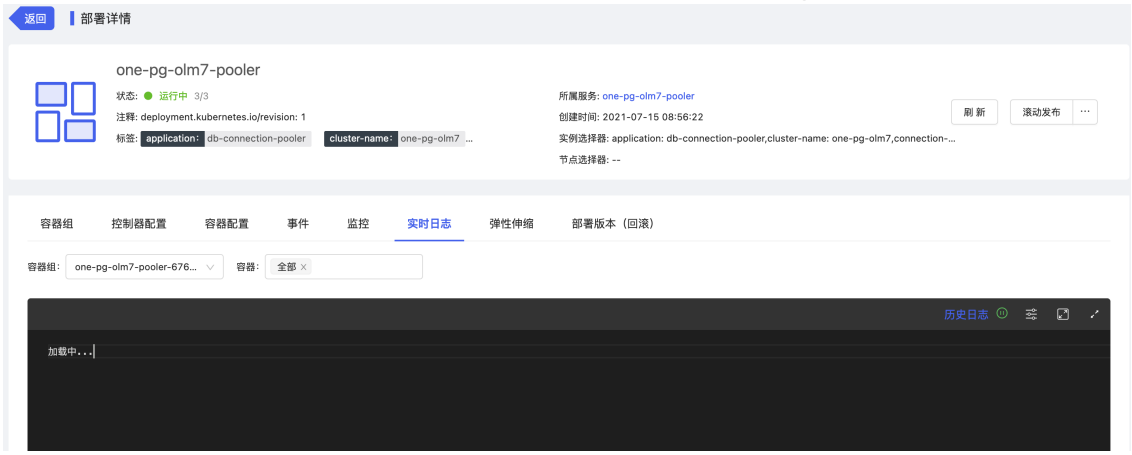
- 在弹出的终端交互窗口里，用前面保存的 PostgreSQL 地址信息，验证 PostgreSQL 能够被成功 访问

五、应用运维指南；（补充界面部署方式）

- **监控信息查看**：在“容器云--工作负载--容器组”，点击进入 PostgreSQL 容器组



- **日志信息查看**：在“容器云--工作负载--有状态副本集”，点击进入 PostgreSQL 容器组



- **事件信息查看**：在“容器云--工作负载--有状态副本集”，点击进入 PostgreSQL 容器组

- **审计信息查看**：在“安全和运维--平台运维--操作审计--审计记录”，选择“容器云/工作负载/容器组”、相应租户、项目后点击“立即查询”