

在 KubeSphere 中部署 RadonDB PostgreSQL

RadonDB PostgreSQL 是基于 PostgreSQL 的开源、云原生、高可用集群解决方案。

本教程演示如何从 KubeSphere 应用商店部署 RadonDB PostgreSQL。

准备工作

- 请确保已启用 OpenPitrix 系统。
- 您需要创建一个企业空间、一个项目和一个用户帐户 (project-regular) 供本教程操作使用。该帐户需要是平台普通用户，并邀请至项目中赋予 operator 角色作为项目操作员。本教程中，请以 project-regular 身份登录控制台，在企业空间 demo-workspace 中的 demo-project 项目中进行操作。有关更多信息，请参见[创建企业空间、项目、帐户和角色](#)。

动手实验

步骤 1：从应用商店中部署 RadonDB PostgreSQL

- 在 demo-project 项目的概览页面，点击左上角的应用商店。

The screenshot shows the KubeSphere Application Store interface. On the left, there is a sidebar with categories: '发现' (Discover), '最新上架' (Newest), '分类' (Categories), '全部' (All), 'Web 服务器' (Web Server), '消息队列' (Message Queue), '存储' (Storage), '网络' (Network), '镜像仓库' (Image Repository), '数据库和缓存' (Database and Cache) which is highlighted with a green border, and '数据导出器' (Data Exporter). In the main area, there is a search bar with the placeholder '查找应用' (Search application) and a 'KUBESPHERE' logo. Below the search bar, it says '共收录 8 款应用' (8 applications listed). There are eight application cards displayed:

- Memcached**: Free & open source, high-performance, distributed memory object caching system.
- Redis**: Redis is an open source (BSD licensed), in-memory data structure store.
- PostgreSQL**: PostgreSQL is a powerful, open source object-relational database system.
- MongoDB**: MongoDB is a general purpose, document-based, distributed database.
- RadonDB MySQL**: High Availability MySQL Cluster, Open Source.
- RadonDB PostgreSQL**: Chart for PostgreSQL with HA architecture. This card has a green border around its title and description, indicating it is the selected item.
- MySQL**: Fast, reliable, scalable, and easy to use open-source relational database.
- etcd**: etcd is a distributed reliable key-value store for the most critical data.

Each card includes developer information ('开发者: admin') and the latest version ('最新: [version]').

2. 找到 RadonDB PostgreSQL，点击应用信息页面上的部署。

The screenshot shows the KubeSphere application store interface. In the center, there is a card for the 'RadonDB PostgreSQL' chart. The card includes the following information:

- 应用商店** (Application Store) and **工作台** (Workstation) buttons.
- KUBESPHERE** logo at the top right.
- 返回** (Back) button.
- RadonDB PostgreSQL** title and subtitle: "Chart for PostgreSQL with HA architecture."
- 部署** (Deploy) button, which is highlighted with a green border.
- 应用信息** (Application Information) tab is selected.
- 应用描述** (Application Description) and **配置文件** (Configuration Files) tabs.
- What is RadonDB PostgreSQL?** section: A brief introduction stating it's an open-source, cloud-native, highly availability cluster solution based on PostgreSQL.
- RadonDB PostgreSQL supports Kubernetes or KubeSphere platforms.**
- Architecture** section: Lists features including automatic failover through repmgr, load-balancing I/O traffic through Pgpool-II, and keeping data consistency through PostgreSQL Streaming Replication.
- Features** section: Lists high availability PostgreSQL database features like leader-follower switching, asynchronous or synchronous streaming replication, and automatic failover.

3. 设置名称并选择应用版本。请确保将 RadonDB PostgreSQL 部署在 demo-project 中，点击下一步。

The screenshot shows the deployment configuration page for the RadonDB PostgreSQL application. It includes the following fields:

- 基本信息** (Basic Information) section:
 - 应用名称 ***: radondb-ehopmc
 - 应用版本 ***: 1.0.1 [11.11.0] (最新版本)
- 描述信息** (Description): A text input field with placeholder text: "描述信息不超过 256 个字符".
- 部署位置** (Deployment Location): A dropdown menu showing the deployment path: **demo-workspace** > **default** > **demo-project**.
- 下一步** (Next) button, which is highlighted with a green border.

4. 在应用配置页面，您可以使用默认配置，或者编辑 YAML 文件以自定义配置。点击部署继续。

The screenshot shows the KubeSphere application configuration interface for RadonDB PostgreSQL. At the top, there's a header with the KubeSphere logo and a 'project-regular' dropdown. Below the header, the title 'RadonDB PostgreSQL' is displayed, followed by a subtitle 'Chart for PostgreSQL with HA architecture.' There are two tabs at the bottom: '基本信息' (Basic Information) and '应用配置' (Application Configuration). A large code editor window titled '应用配置' contains a YAML configuration file:

```
1. ## Global Docker image parameters
2. ## Please, note that this will override the image parameters, including dependencies, configured to use the
   global value
3. ## Current available global Docker image parameters: imageRegistry, imagePullSecrets and storageClass
4. ##
5. # global:
6. #   imageRegistry: myRegistryName
7. #   imagePullSecrets:
8. #     - myRegistryKeySecretName
9. #   storageClass: myStorageClass
10. #   postgresql:
11. #     username: customuser
12. #     password: custompassword
13. #     database: customdatabase
14. #     repmgrUsername: repmgruser
15. #     repmgrPassword: repmgrpassword
16. #     repmgrDatabase: repmgrdatabase
17. #     existingSecret: myExistingSecret
18. #     ldap:
19. #       bindpw: bindpassword
20. #       existingSecret: myExistingSecret
21. #     pgpool:
22. #       adminUsername: adminuser
23. #       adminPassword: adminpassword
24. #       existingSecret: myExistingSecret
```

A green button labeled '部署' (Deploy) is located in the top right corner of the configuration area.

5. 稍等片刻待 RadonDB PostgreSQL 启动并运行。

The screenshot shows the KubeSphere project overview page for 'demo-project'. On the left, there's a sidebar with project navigation: '概览', '应用负载', '存储管理', '配置中心', '监控告警', and '项目设置'. The main content area is titled '应用' (Application) and includes sections for '应用模板' (Application Template) and '自制应用' (Custom Application). It also features a search bar and a table listing applications. One entry in the table is highlighted:

名称	状态	应用	版本	上次更新时间
radondb-vtkoy	正在创建	radondb-postgresql	1.0.1 [11.11.0]	2021-07-20 14:10:12

At the bottom of the table, it says '共 1 个条目' (1 item) and has navigation arrows. There are also buttons for '部署示例应用' (Deploy Example Application) and '部署新应用' (Deploy New Application).

步骤 2：查看 PostgreSQL 集群状态

1. 在 demo-project 项目的概览页面，可查看当前项目资源使用情况。

概览

demo-project
demo-project

demo-workspace
企业空间
project-admin
创建者

已部署应用(Helm)

R radondb-vtkoy9

资源状态

应用资源 物理资源

CPU 使用量 (最近 12 小时)

内存使用量 (最近 12 小时)

帮助信息

邀请其他成员到当前项目中?

如何设置项目网关?

资源用量排行 (Top5)

按 CPU 使用量排行

radondb-vtkoy9-radondb-postgresql 28 m
有状态副本集

radondb-vtkoy9-radondb-postgresql-pgpool 8 m
部署

2. 进入应用负载下的工作负载页面，点击有状态副本集，查看集群状态。

应用商店 工作台

KUBESPHERE

demo-project
demo-project

概览

应用负载

工作负载

部署 有状态副本集 (highlighted)

守护进程集

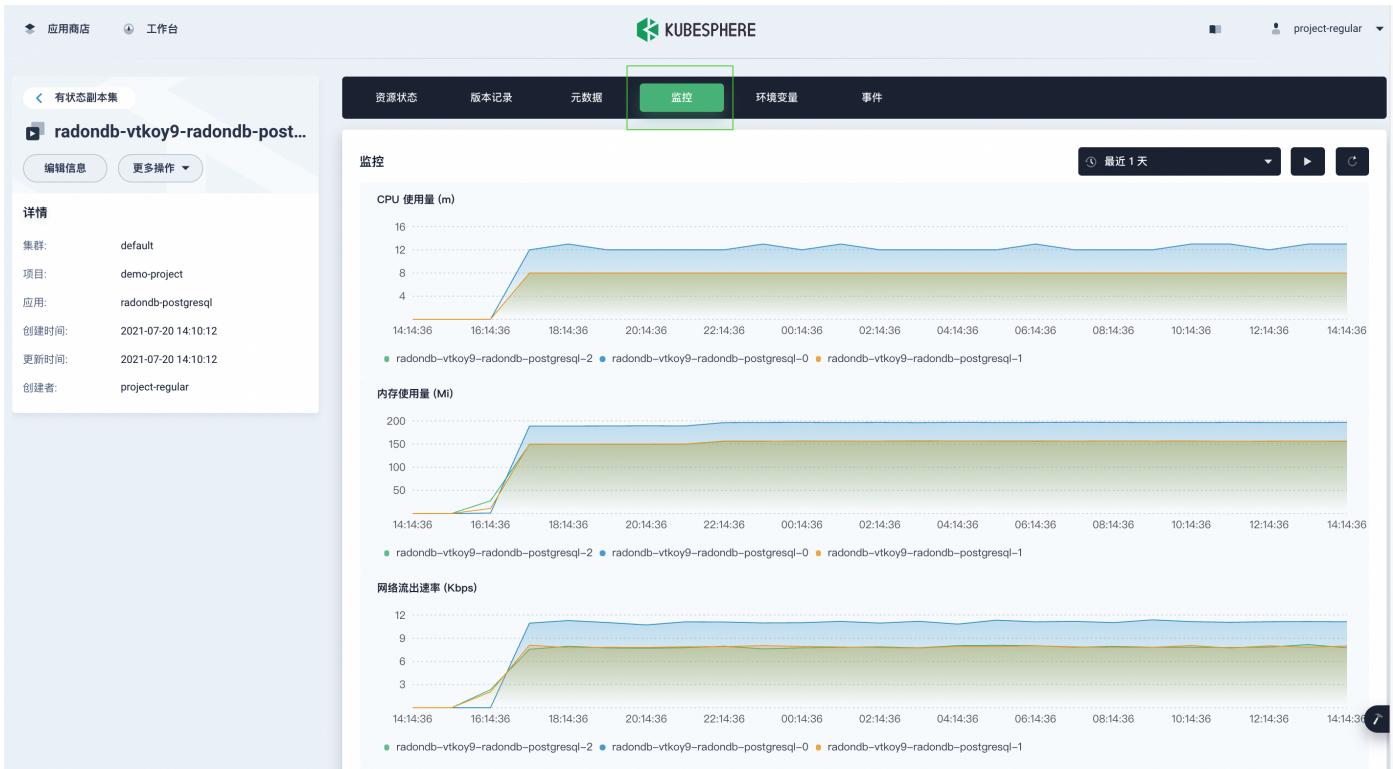
输入查询条件进行过滤

名称 状态 应用 创建时间

radondb-vtkoy9-radondb-postgresql
运行中 (3/3) radondb-postgresql 2021-07-20 14:10:12

共 1 个条目

进入一个有状态副本集群详情页面，点击监控标签页，可查看一定时间范围内的集群指标。



3. 进入应用负载下的容器组页面，可查看所有状态的容器。

The screenshot shows the KubeSphere application load balancer pod management interface for the project 'demo-project'. The left sidebar menu is expanded to show '应用负载' (Application Load Balancer) under '容器组' (Pods). The main content area displays a table of running pods, each with a status icon, name, node, IP address, application name, and update time. There are four pods listed: 'radondb-vtkoy9-radondb-postgresql-pgpool-864648d8c4-5' (status: 运行中), 'radondb-vtkoy9-radondb-postgresql-2' (status: 运行中), 'radondb-vtkoy9-radondb-postgresql-1' (status: 运行中), and 'radondb-vtkoy9-radondb-postgresql-0' (status: 运行中).

4. 进入存储管理下的存储卷页面，可查看存储卷，所有组件均使用了持久化存储。

存储卷

什么是存储卷类型?

什么是本地存储卷 (Local Volume)?

输入查询条件进行过滤

名称 状态 访问模式 挂载 创建时间

data-radondb-vtkoy9-radondb-postgresql-2
csi-ssd-enterprise
准备就绪 ReadWriteOnce 已挂载 2021-07-20 16:08

data-radondb-vtkoy9-radondb-postgresql-1
csi-ssd-enterprise
准备就绪 ReadWriteOnce 已挂载 2021-07-20 16:08

data-radondb-vtkoy9-radondb-postgresql-0
csi-ssd-enterprise
准备就绪 ReadWriteOnce 已挂载 2021-07-20 14:10

共 3 个条目

查看某个存储卷用量信息，以其中一个数据节点为例，可以看到当前存储的存储容量和剩余容量等监控数据。

资源状态 元数据 事件 快照信息

存储卷

Kubernetes 采集的是存储卷的设备用量数据，未挂载的存储卷暂时采集不到，并且对于如 OpenEBS/Local PV、NFS 等路径型存储卷通常与实际用量有一定出入。详见存储卷监控数据分析。

1.63% 9.61 Gi 9.78 Gi
剩余存储 存储卷容量

Inode 使用率 (%)

0.2 0.15 0.1 0.05
13:17:04 13:23:04 13:29:04 13:35:04 13:41:04 13:47:04 13:53:04 13:59:04 14:05:04 14:17:04

已挂载容器组

请输入关键字过滤

radondb-vtkoy9-radondb-postgresql-0 创建于 1 天前	worker-s002(192.168.0.26)	10.1 MiB	CPU 13 m	内存 196.97 Mi
---	---------------------------	----------	----------	--------------

共 1 个条目

步骤 3：访问 RadonDB PostgreSQL

1. 在 应用负载下的容器组页面，点击一个容器的名称，进入容器详情页面。
2. 在 资源状态页面，点击终端图标。

容器组

radondb-vtkoy9-radondb-post...

查看配置文件 剔除

集群: default
项目: demo-project
应用: radondb-postgresql
状态: 运行中
容器组 IP: 10.1
节点名称: worker-s002
节点 IP: 192.168.0.26
重启次数(总计): 5
QoS Class: Burstable
创建时间: 2021-07-20 16:11:46

资源状态 调度信息 元数据 监控 环境变量 事件

容器

终端 pgpool 镜像: docker.io/radondb/pgpool:4.2.2-debian-10-r1

运行中 状态 5 重启次数 5432/TCP 端口

存储设备

dshm 类型: 临时存储卷
pgpool /dev/shm (读写)

default-token-m96mz default-token-m96mz
类型: 密钥
pgpool /var/run/secrets/kubernetes.io/serviceaccount (只读)

3. 在弹出窗口中，向终端输入命令使用该应用。

```
psql -h <Pod name> -p 5432 -U postgres -d postgres
```

terminal

```
$ psql -h radondb-vtkoy9-radondb-postgresql-pgpool-864648d8c4-52cdh -p 5432 -U postgres -d postgres
Password for user postgres:
pgsql (10.16, server 11.11)
WARNING: psql major version 10, server major version 11.
          Some psql features might not work.
Type "help" for help.

postgres=# select version();
              version
-----
 PostgreSQL 11.11 on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
(1 row)

postgres=#

```

pgpool 镜像: docker.io/radondb/pgpool:4.2.2-debian-10-r1

基本信息

状态: 运行中
镜像: docker.io/radondb/pgpool:4.2.2-debian-10-r1
命令: -
资源预留: cpu: 1 / memory: 1Gi
资源限制: cpu: 2 / memory: 2Gi
重启次数: 5

4. 如果您想从集群外部访问 RadonDB PostgreSQL，详细信息请参见 [RadonDB PostgreSQL 开源项目](#)。