

PostgreSQL数据库使用指南

1 商品说明

PostgreSQL 是一款开源对象-关系型数据库管理系统（ORDBMS），起源于 1986 年加州大学伯克利分校的 POSTGRES 项目，1996 年正式更名为 PostgreSQL。其以可靠性、功能完备性和高度兼容 SQL 标准著称，支持 ACID 事务、复杂查询、外键约束、触发器等核心特性。

本商品通过鲲鹏服务器+Euler2.0 部署。

2 商品购买

您可以在云商店搜索“PostgreSQL数据库”。

其中，地域、规格、推荐配置使用默认，购买方式根据您的需求选择按需/按月/按年，短期使用推荐按需，长期使用推荐按月/按年，确认配置后点击“立即购买”。

3 商品资源配置

商品支持ECS控制台配置，下面对资源配置的方式进行介绍。

3.1 ECS 控制台配置

3.1.1 准备工作

在使用ECS控制台配置前，需要您提前配置好安全组规则。

安全组规则的配置如下：

- 入方向规则放通端口5432，源地址内必须包含您的客户端ip，否则无法访问
- 入方向规则放通CloudShell连接实例使用的端口22，以便在控制台登录调试。
- 出方向规则一键放通

3.1.2 创建 ECS

前提工作准备好后，选择ECS控制台配置跳转到购买ECS页面，ECS资源的配置如下图所示：

购买弹性云服务器

1 购买配置 2 网络配置 3 高级配置 4 确认配置

区域: 华北-北京四

计费模式: 包年包月 按需计费 竞价计费

可用区: 可用区1 可用区2 可用区3 可用区7

实例规格: 通用计算型 业务场景型

CPU架构: x86 鲲鹏

规格: 最新系列 vCPUs 内存 规格名称

通用计算增强型 通用计算型 内存优化型 超大内存型 内存增强型 通用内存型 GPU加速型 AI加速型 通用入门型

规格: s6 s6 s6 收起规格

实例规格	规格名称	vCPUs	内存	CPU	基准/最大带宽	内网带宽	规格参考价格
通用计算型s6	s6 medium.4	1vCPUs	4GB	Intel Cascade Lake 2.6GHz	0.1 / 0.8 Gbit/s	10万PPS	¥0.34/小时
通用计算型s6	s6 large.2	2vCPUs	4GB	Intel Cascade Lake 2.6GHz	0.2 / 1.5 Gbit/s	15万PPS	¥0.36/小时
通用计算型s6	s6 large.4	2vCPUs	8GB	Intel Cascade Lake 2.6GHz	0.2 / 1.5 Gbit/s	15万PPS	¥0.68/小时

购买数量: 1 配置费用: ¥0.3988/小时 + 弹性公网IP流量费用: ¥0.80/Gb

2 网络配置 3 高级配置 4 确认配置

网络: vpc_kettle_app(192.168.10.0/24) C subnet_kettle_app(192.168.10.0/24) C 自动分配IP地址 可用私有IP数量: 49个

扩展网卡: 增加一块网卡 您还可以增加1块网卡

安全组: secgroup_kettle_app(4b014374-5380-42c7-a371-6456c1...) 新建安全组

安全组描述: 安全组默认防火墙功能。是一个逻辑上的分组，用于设置网络访问控制。请确保所选安全组已开放22端口 (Linux SSH登录)、3389端口 (Windows远程登录) 和 ICMP 协议 (Ping)。配置安全组规则

静默安全组规则

安全组名称	优先级	策略	协议端口	类型	源地址	描述
secgroup_kettle_app	1	允许	TCP 8080	IPv4	0.0.0.0	--

弹性公网IP: 现在购买 使用已有 暂不购买

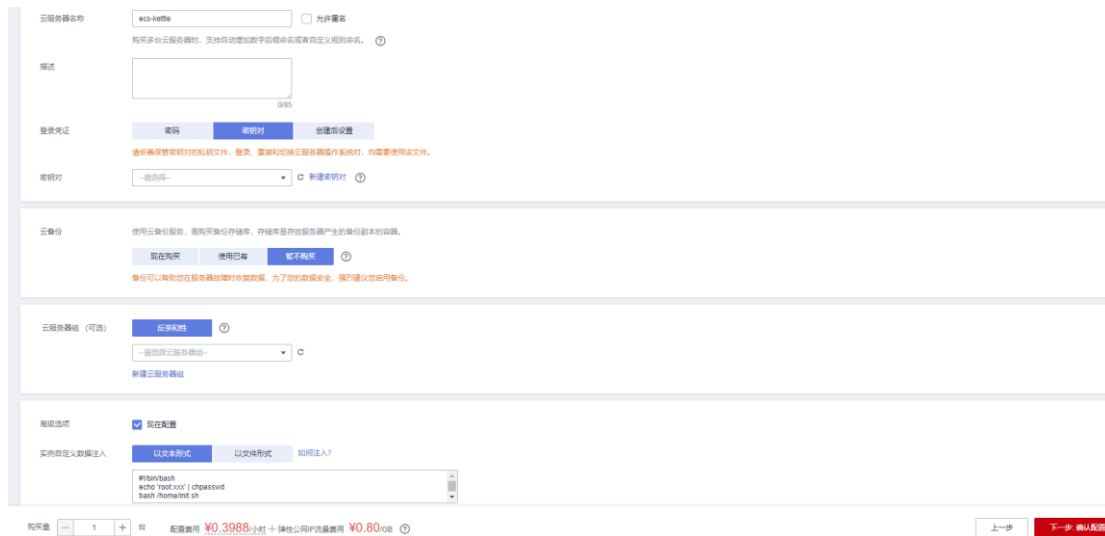
计费: 全局计费 静态BGP

公网带宽: 按带宽计费 按流量计费 加入流量套餐

带宽大小: 5 10 20 50 100 自定义 1-300 Mbit/s

购买数量: 1 配置费用: ¥0.3988/小时 + 弹性公网IP流量费用: ¥0.80/Gb

上一步 下一步 立即购买



值得注意的是：

- VPC您可以自行创建
- 安全组选择3.1.1章节中配置的安全组
- 弹性公网IP选择现在购买，推荐选择“按流量计费”，带宽大小可设置为5Mbit/s
- 其余默认或按规则填写即可。

4 商品使用

4.1 PostgreSQL 使用

4.1.1 容器内连接测试

执行以下命令，连接PostgreSQL

```
【docker exec -it postgres-arm psql -U postgres】
```

在PostgreSQL命令行内执行

```
【
```

```
// 创建测试数据库
```

```
CREATE DATABASE test_db;
```

```
// 切换数据库
```

```
\c test_db;
```

```
// 创建表
```

```
CREATE TABLE users (
```

```
id SERIAL PRIMARY KEY,  
name VARCHAR(50) NOT NULL,  
email VARCHAR(100) UNIQUE  
);  
  
// 插入数据  
  
INSERT INTO users (name, email) VALUES  
  
('张三', 'zhangsan@example.com'),  
  
('李四', 'lisi@example.com');  
  
// 查询测试数据  
  
SELECT * FROM users;
```

】

```
[root@postgresql opt]# docker exec -it postgres-arm psql -U postgres  
psql (14.18 (Debian 14.18-1.pgdg120+1))  
Type "help" for help.  
  
postgres=# CREATE DATABASE test_db;  
CREATE DATABASE  
postgres=# \c test_db;  
You are now connected to database "test_db" as user "postgres".  
test_db=# CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE  
);  
CREATE TABLE  
test_db=# INSERT INTO users (name, email) VALUES  
('张三', 'zhangsan@example.com'),  
('李四', 'lisi@example.com');  
INSERT 0 2  
test_db=# SELECT * FROM users;  
 id | name |      email  
-----+-----+-----  
  1 | 张三 | zhangsan@example.com  
  2 | 李四 | lisi@example.com  
(2 rows)
```

4.2 参考文档

- [PostgreSQL官网](#)