

HashData

目录

1 入门指南	1.1
2 步骤1: 准备工作	2.1
3 步骤2: 启动 HashData 数据仓库样例集群	3.1
4 步骤3: 授权连接样例集群	4.1
5 步骤4: 连接样例集群	5.1
6 步骤5: 将样例数据从对象存储加载到 HashData 数据仓库中	6.1
7 步骤6: 寻找额外的资料和重设你的环境	7.1

入门指南

欢迎来到 HashData 数据仓库指南。HashData 数据仓库是一个高性能，完全托管的 PB 级数据仓库服务。一个 HashData 数据仓库是由一组称之为节点的计算资源组成的集群。每个集群作为一个 HashData 数据仓库引擎，包含一个或多个数据库。

这个指南的目的是指导你创建一个 HashData 数据仓库样例集群。你可以通过这个样例集群来测试 HashData 数据仓库的功能。在这个教程中，你将执行如下步骤：

- 步骤1：准备工作
- 步骤2：启动 HashData 数据仓库样例集群
- 步骤3：授权连接样例集群
- 步骤4：连接样例集群
- 步骤5：将样例数据从对象存储加载到 HashData 数据仓库中
- 步骤6：寻找额外的资料和重设你的环境

这个教程的目的不是为了配置生产环境的，所以不会深入地讨论操作中的各种选项。当你完成了这个教程的所有步骤后，你能通过“额外的资料”章节找到关于集群计划、部署和维护，以及如何操作数据仓库中数据的更深入的信息。

步骤1: 准备工作

在你创建第一个 HashData 数据仓库集群前，确保在这个章节中完成如下准备工作：

- 注册青云账号
- 安装 SQL 客户端工具
- 确认网络连接
- 确认对象存储配额
- 创建 API 秘钥

注册青云账号

你首先需要注册一个青云账号。如果你已经拥有了一个青云账号，那么你可以跳过这个步骤，使用已经拥有的账号进行接下来的操作。

1. 打开链接 <http://qingcloud.com>，点击注册。
2. 按照页面提示完成注册流程。

安装 SQL 客户端工具

你可以使用任何 postgres 兼容的客户端程序连接到 HashData 数据仓库，比如 psql。此外，你还可以通过绝大部分使用标准数据库应用接口，如 JDBC，ODBC 的客户端程序连接到 HashData 数据仓库。最后，你还可以使用标准数据库应用接口开发自己的客户端程序来访问 HashData 数据仓库。由于 HashData 数据仓库基于 greenplum，而后者又是基于 postgres 而来，所以你可以直接使用 postgres 驱动访问 HashData 数据仓库。在这个教程中，我们将通过 psql 这个 postgres 的客户端程序演示如何连接到 HashData 数据仓库。

安装 psql

1. 如果你正在使用 Linux 操作系统，你可以使用以下命令安装 psql。

```
Redhat/Centos:  
# yum install postgresql  
  
Ubuntu:  
# apt-get install postgresql-client
```

2. 或者访问 [PostgreSQL 官方网站](#)，根据你的操作系统下载安装包。

确认网络连接

HashData 数据仓库使用 5432 作为服务的端口地址。

当你希望从云的外部访问 HashData 数据仓库服务时，你需要拥有一个公网 IP 地址，并将公网 IP 地址绑定在主节点上。同时，你还需要配置防火墙规则。HashData 集群默认会加入 [集群缺省防火墙](#)，你需要配置端口为 5432 协议为 TCP 的防火墙下行规则。

如果你的客户端程序运行在云的内部，那么你既不需要绑定公网 IP 地址，也不需要配置任何防火墙规则。

确认对象存储配额

对象存储是由云厂商提供的一种高可用低成本的存储系统，HashData 使用对象存储作为数据仓库的存储系统，极大的降低了用户数据存储的成本。为了使用对象存储，用户需要确认所使用的账户拥有创建对象存储 Bucket 的权限，并且拥有创建至少一个新 Bucket 的配额。

如果用户没有使用对象存储的权限，或者没有足够的 Bucket 配额，创建 HashData 数据仓库集群的操作将会失败。

创建 API 秘钥

HashData 数据仓库使用用户提供的 API 秘钥创建和访问对象存储，因此在创建 HashData 数据仓库集群之前，请提前申请 API 秘钥。我们建议用户为每个数据仓库集群申请单独的，独占使用的 API 秘钥。

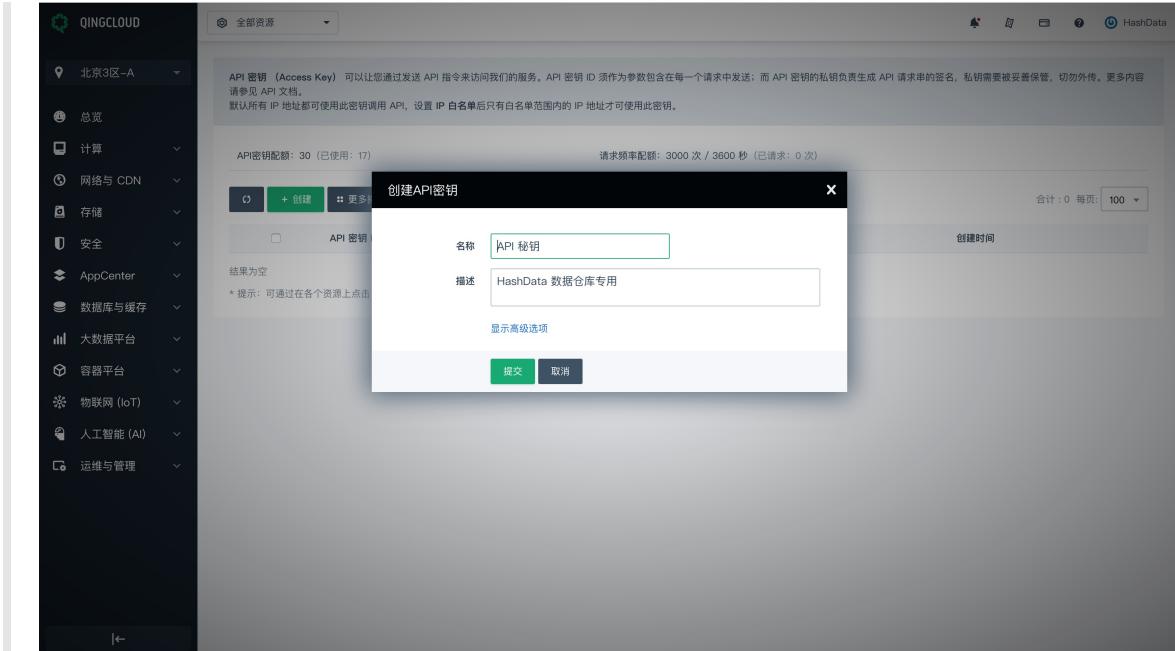
如果用户在创建 HashData 数据仓库时填写了错误的 API 秘钥，创建 HashData 数据仓库集群将会失败。

步骤2: 启动 HashData 数据仓库样例集群

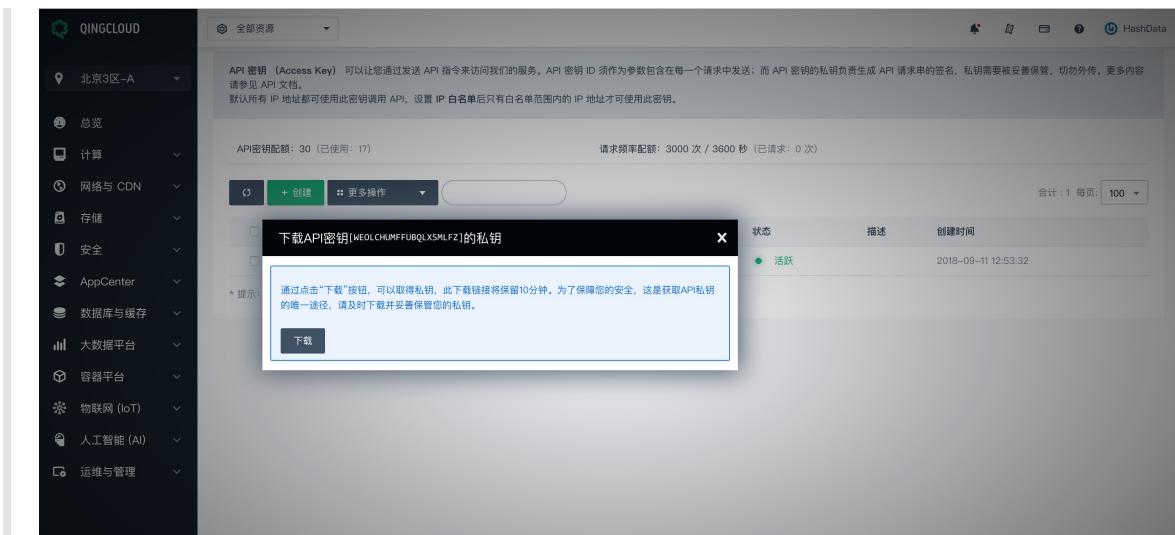
完成上面的前提步骤后，现在你能够开始启动一个 HashData 数据仓库集群。

启动 HashData 数据仓库集群

- 建议用户创建 HashData 数据仓库专用的 API 秘钥，如果您已经有创建好 API 秘钥，可以跳过此步，示例中我们设置一个 demo 密钥，注意 demo 只是演示的示例名称并不代表密钥本身。



通过点击“下载”按钮，可以取得私钥，此下载链接将保留10分钟。为了保障您的安全，这是获取API私钥的唯一途径，请及时下载并妥善保管您的私钥

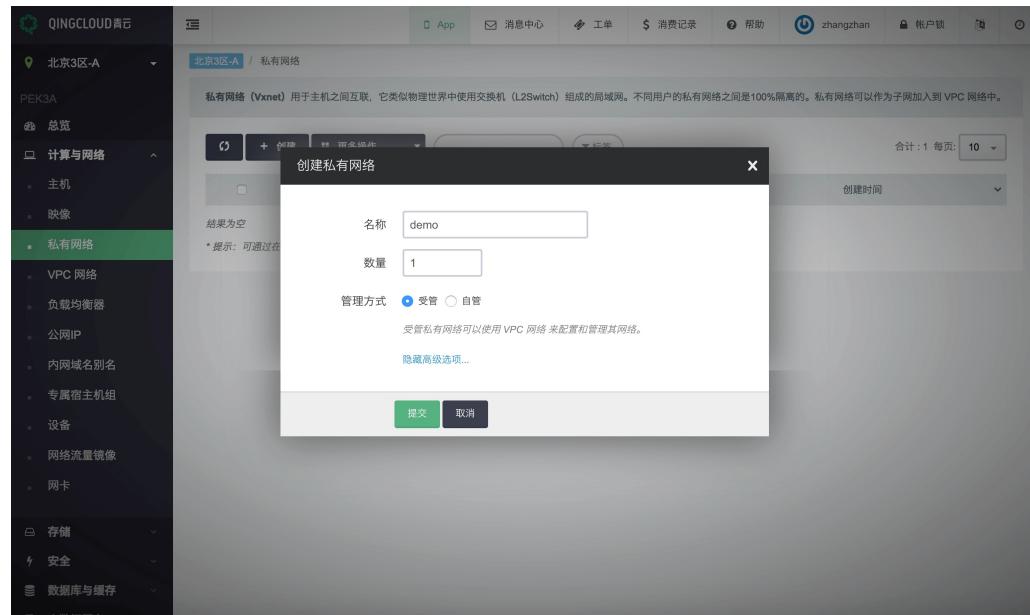


- 登陆青云并在应用中心找到对应的产品 [数据仓库\(HashData高性能MPP数据仓库\)](#)。
- 点击部署到控制台，您可以选择在哪个数据中心创建你的数据仓库集群。在这个教程中，我们选择了北京3区A。

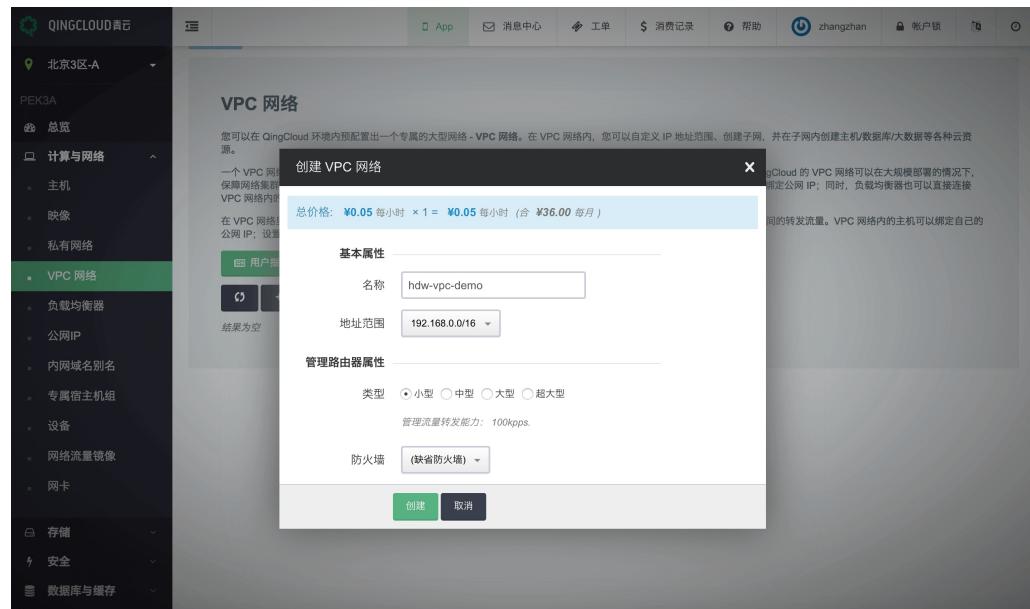
4. 填写基本信息，同时您可以根据实际需求选择资源配置类型，目前包括标准版，企业版，自定义版本。标准版定义为 HashData 数据仓库集群的最小配置，开发测试或者小规模部署可以选择标准版，企业版适用于大多数生产环境。另外用户可以按需进行自定义集群配置。

5. 创建依赖资源：你需要有一个已连接到 VPC 的私有网络。如果您没有创建好依赖资源，点击创建后，可以按照提示完成下面的步骤，如果你已经有一个可用的私有网络，可以跳过此步：

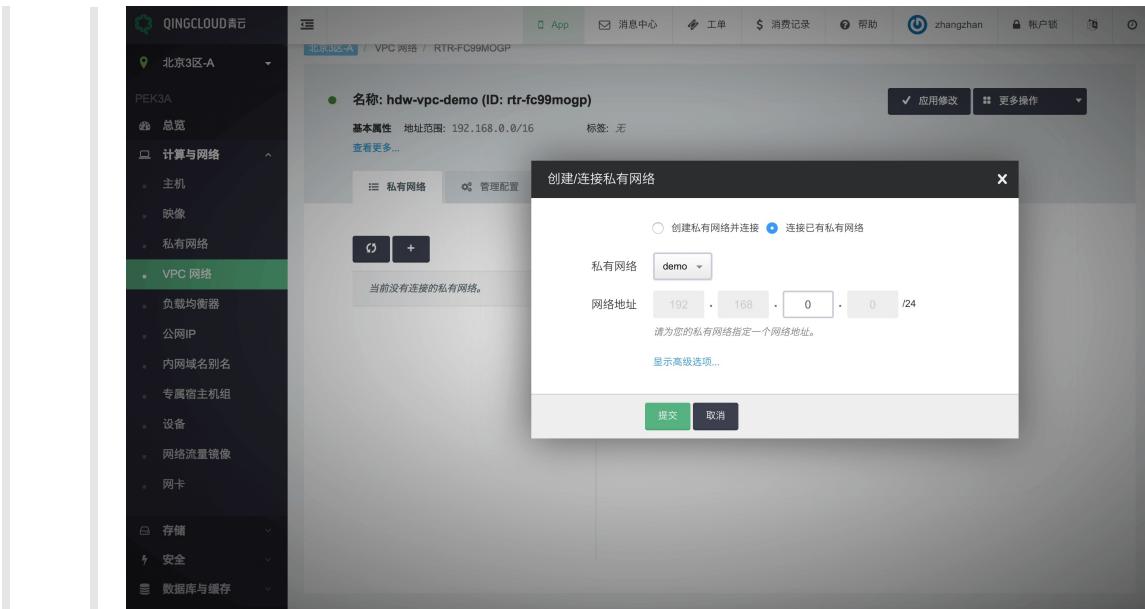
- 创建私有网络：计算机与网络 -> 私有网络，点击创建



- 创建 VPC 网络 : 计算机与网络 -> VPC网络 , 点击创建



- 连接私有网络到 VPC 网络 : 计算机与网络 -> VPC网络 , 点击创建完成的VPC网络 , 将上面创建完成的私有网络添加到VPC中



6. 设置私有网络，你可以选择具体的私有网络，同时根据实际情况选择自动分配和手动指定节点 IP，我们选择自动分配方式为您演示。当前缺省的私有网络名称为 vxnet0



7. 配置服务环境参数，你可以在青云设置自己的API密钥，密钥是访问对象存储期间使用的，结合实际情况创建对应数据库用户名，密码及其设置数据库名，最后完成提交操作。



8. 点击提交后，会显示如下信息内容，请耐心等待

在 AppCenter 控制面板中，选择新创建的集群并且查看集群状态信息。在你连接数据仓库之前，一定要确认集群的状态是可用的，并且数据库的健康状态是正常

The screenshot shows the QINGCLOUD AppCenter interface. On the left, there's a sidebar with various service icons and a navigation tree. The main area is titled '北京3区-A / APPCENTER / 管理 / 集群 / CL-FGDB10H1'. It displays the '基本属性' (Basic Properties) of a cluster named 'Data Warehouse on QingStor'. Key details include:

- ID: cl-fgdb10h1
- 状态: 活跃 (Active)
- 名称: Data Warehouse on QingStor
- 描述: 测试集群案例
- 应用: 数据仓库
- 版本: v2.1.1 对象存储
- 测试模式: 是
- 节点数量: 3
- 私有网络: vxnet0
- 授权提供商: 否
- 创建时间: 2018-09-07 15:57:59
- 创建于: 2分钟前

Below this, there's a section for '服务端口信息' (Service Port Information) with a client_port协议: tcp and 端口: 5432.

The right side of the screen shows a table titled '节点' (Nodes) with three entries:

节点	名称	角色	主机类型	节点状态	服务状态	配置	IP	防火墙	告警状态	监控
cln-wgnznrt1	无	主节点	性能型	活跃	获取中	2核 4G 10G	192.168.100.9	(cluster default security group)	无	<input checked="" type="radio"/>
cln-Sxzpsf6p	无	计算节点	性能型	活跃	获取中	2核 4G 10G	192.168.100.8 10.91.117.252	(cluster default security group)	无	<input type="radio"/>
cln-dvwywzefs	无	计算节点	性能型	活跃	获取中	2核 4G 10G	192.168.100.2 10.91.104.253	(cluster default security group)	无	<input type="radio"/>

* 提示: 可通过在各个资源上点击「右键」来进行常用操作, 以及「双击」来修改基本属性。

At the bottom, there are tabs for 资源 (Resources), 最近6小时 (Last 6 Hours), 最近一天 (Last Day), 最近两周 (Last Two Weeks), 最近一个月 (Last Month), and 最近6个月 (Last Six Months). A 'CPU' monitoring section is also present.

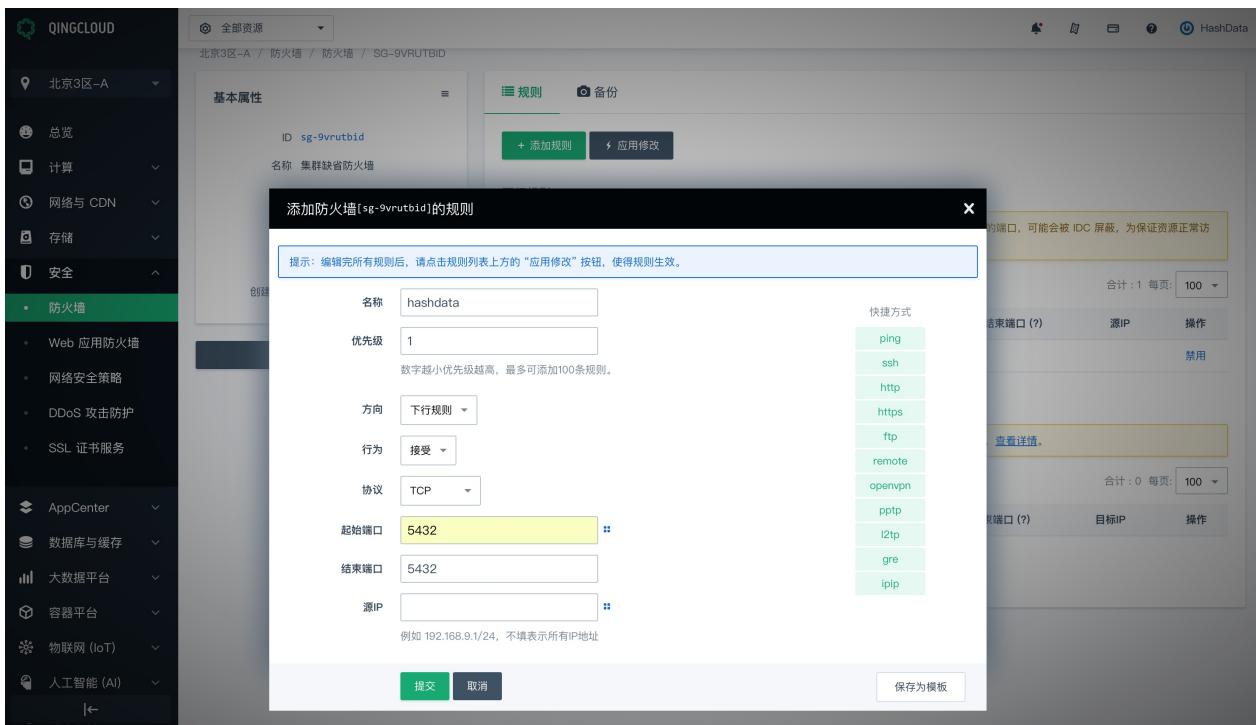
9. 在 AppCenter 控制面板对应配置参数位置，有您之前通过 API 密钥功能下载生成的密钥信息，以及对应的密钥私钥，请妥善保管密钥安全性，点击修改属性，您可以去修改对应 API 密钥参数属性信息

步骤3: 授权连接样例集群

在前面的步骤中，你已经创建启动了你的 HashData 数据仓库集群。在你连接到数据仓库集群之前，你需要对上一步骤中创建的路由器进行相应的配置。

防火墙配置

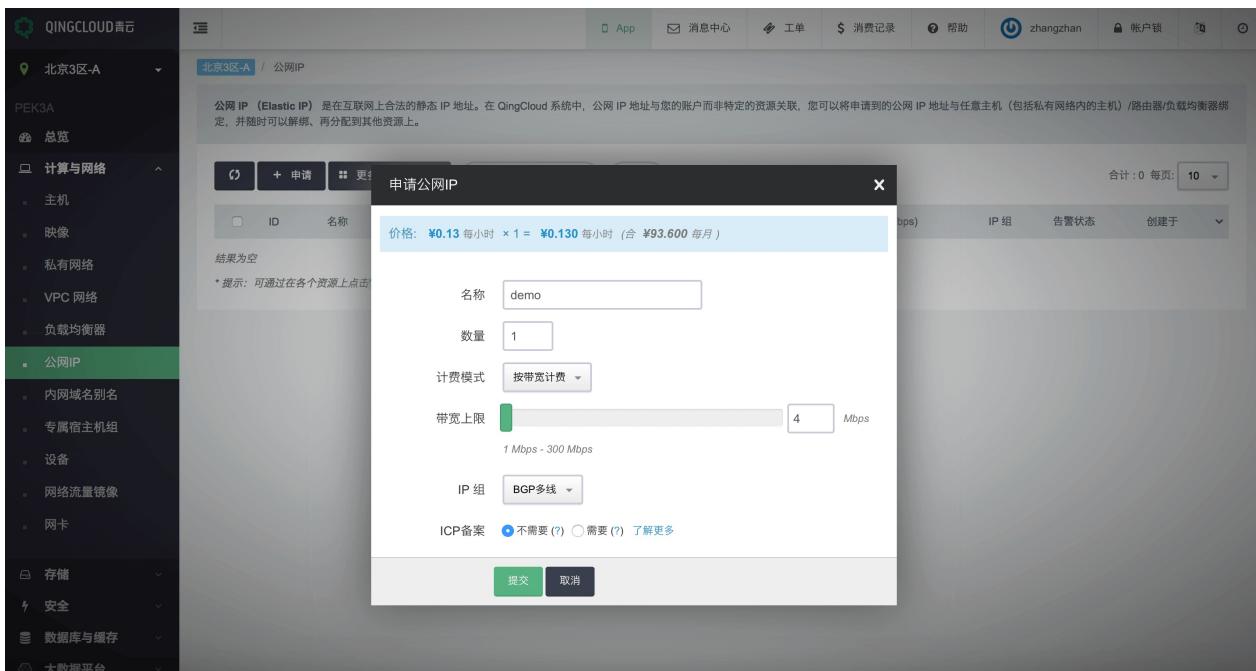
在路由器的详情页面，点击选用的防火墙进入其详情页面，添加一条打开 5432 端口的下行规则，如下所示：



这条下行规则允许你的 SQL 客户端工具能够访问路由器的 5432 端口。

配置公网 IP

如果你的 SQL 客户端不在青云的网络里，你还需要申请一个公网 IP 地址，并绑定到数据仓库集群的主节点。从云内部访问 HashData 数据仓库集群不需要绑定公网 IP。



将公网 IP 绑定到数据仓库：

The screenshot shows the 'Nodes' configuration page. It lists three nodes: 'cln-xxa1' (主节点, 性能型), 'cln-baravq5y' (外部 IP, 点型), and 'cln-horv3b66' (计算节点, 性能型). The 'cln-xxa1' row has a context menu open over its 'IP' column, with options '+ 绑定' (Bind) and 'x 解绑' (Unbind). The 'IP' column for this node shows '192.168.100.2' and '10.91.89.253'. Other columns include 节点状态 (Status), 服务状态 (Service Status), 配置 (Configuration), 防火墙 (Firewall), 告警状态 (Alert Status), and 监控 (Monitoring). A note at the bottom says: '* 提示：可通过在各个资源上点击「右键」来进行常用操作，以及「双击」来修改基本属性。'

步骤4: 连接样例集群

现在你可以通过 SQL 客户端工具连接到你的数据仓库集群，并且跑一条简单的查询语句来测试连接。你能够使用几乎所有与 postgres 兼容的 SQL 客户端工具。在这个教程中，你将使用在准备工作中安装的 postgres 自带的 psql 客户端。

确定连接 IP 地址和端口

数据仓库集群的 IP 地址可由配置公网 IP 步骤中确定。在剩下的教程中，我们用 121.201.25.29 作为例子。在集群主控制台，选择 examplecluster 进入详情页面。从详情页面中，你能看到端口：5432。

使用 psql 连接到集群

你可以通过下面命令连接到集群：

```
psql -d postgres -h 121.201.25.29 -p 5432 -U admin
```

然后根据提示输入登陆密码。

简单测试查询

登陆数据仓库后，你可以运行如下命令做一些简单的测试查询：

```
postgres=# CREATE TABLE foo (a INT, b INT);
NOTICE: Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'a' as the Greenplum Database data distribution key for this table.
HINT: The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.
CREATE TABLE
postgres=# INSERT INTO foo (SELECT i, i + 1 FROM generate_series(1, 10000) AS i);
INSERT 0 10000
postgres=# SELECT COUNT(*) FROM foo;

## count

10000
(1 row)

postgres=# SELECT SUM(a) FROM foo;

## sum

50005000
(1 row)
```

步骤5：将样例数据从对象存储加载到 HashData 数据仓库中

现在你已经有了一个名为 postgres 的数据库，并且你已经成功地连接上它了。接下来你可以在数据库中创建一些新表，然后加载数据到这些表中，并尝试一些查询语句。为了方便你的测试，我们准备了一些 TPC-H 的样例数据存储在青云对象存储中。

1. 创建表

拷贝并执行下面的建表语句在 postgres 数据库中创建相应的表对象。你可以通过 HashData 数据仓库[开发指南](#)查看更详细的建表语法。

其中定义的外部表（READABLE EXTERNAL TABLE）用来访问青云对象存储上面的数据。我们提供了 1GB、10GB、100GB 的 TPC-H 公共测试数据集，在此示例中我们使用 1GB 的 TPC-H 数据集。

```
CREATE TABLE NATION (
    N_NATIONKEY  INTEGER NOT NULL,
    N_NAME        CHAR(25) NOT NULL,
    N_REGIONKEY   INTEGER NOT NULL,
    N_COMMENT     VARCHAR(152));

CREATE TABLE REGION (
    R_REGIONKEY  INTEGER NOT NULL,
    R_NAME        CHAR(25) NOT NULL,
    R_COMMENT     VARCHAR(152));

CREATE TABLE PART (
    P_PARTKEY      INTEGER NOT NULL,
    P_NAME         VARCHAR(55) NOT NULL,
    P_MFGR          CHAR(25) NOT NULL,
    P_BRAND         CHAR(10) NOT NULL,
    P_TYPE          VARCHAR(25) NOT NULL,
    P_SIZE          INTEGER NOT NULL,
    P_CONTAINER     CHAR(10) NOT NULL,
    P_RETAILPRICE  DECIMAL(15,2) NOT NULL,
    P_COMMENT       VARCHAR(23) NOT NULL );

CREATE TABLE SUPPLIER (
    S_SUPPKEY      INTEGER NOT NULL,
    S_NAME         CHAR(25) NOT NULL,
    S_ADDRESS      VARCHAR(40) NOT NULL,
    S_NATIONKEY    INTEGER NOT NULL,
    S_PHONE        CHAR(15) NOT NULL,
    S_ACCTBAL     DECIMAL(15,2) NOT NULL,
    S_COMMENT      VARCHAR(101) NOT NULL);

CREATE TABLE PARTSUPP (
    PS_PARTKEY    INTEGER NOT NULL,
    PS_SUPPKEY    INTEGER NOT NULL,
    PS_AVAILQTY   INTEGER NOT NULL,
    PS_SUPPLYCOST DECIMAL(15,2) NOT NULL,
    PS_COMMENT    VARCHAR(199) NOT NULL );

CREATE TABLE CUSTOMER (
```

```

C_CUSTKEY      INTEGER NOT NULL,
C_NAME         VARCHAR(25) NOT NULL,
C_ADDRESS       VARCHAR(40) NOT NULL,
C_NATIONKEY    INTEGER NOT NULL,
C_PHONE        CHAR(15) NOT NULL,
C_ACCTBAL     DECIMAL(15,2) NOT NULL,
C_MKTSEGMENT   CHAR(10) NOT NULL,
C_COMMENT      VARCHAR(117) NOT NULL);

CREATE TABLE ORDERS (
O_ORDERKEY      INT8 NOT NULL,
O_CUSTKEY      INTEGER NOT NULL,
O_ORDERSTATUS   CHAR(1) NOT NULL,
O_TOTALPRICE   DECIMAL(15,2) NOT NULL,
O_ORDERDATE    DATE NOT NULL,
O_ORDERPRIORITY CHAR(15) NOT NULL,
O_CLERK        CHAR(15) NOT NULL,
O_SHIPPRIORITY INTEGER NOT NULL,
O_COMMENT      VARCHAR(79) NOT NULL);

CREATE TABLE LINEITEM (
L_ORDERKEY      INT8 NOT NULL,
L_PARTKEY      INTEGER NOT NULL,
L_SUPPKEY      INTEGER NOT NULL,
L_LINENUMBER   INTEGER NOT NULL,
L_QUANTITY     DECIMAL(15,2) NOT NULL,
L_EXTENDEDPRICE DECIMAL(15,2) NOT NULL,
L_DISCOUNT     DECIMAL(15,2) NOT NULL,
L_TAX          DECIMAL(15,2) NOT NULL,
L_RETURNFLAG   CHAR(1) NOT NULL,
L_LINESUPPLY   CHAR(1) NOT NULL,
L_SHIPDATE     DATE NOT NULL,
L_COMMITDATE   DATE NOT NULL,
L_RECEIPTDATE  DATE NOT NULL,
L_SHIPINSTRUCT CHAR(25) NOT NULL,
L_SHIPMODE     CHAR(10) NOT NULL,
L_COMMENT      VARCHAR(44) NOT NULL);

CREATE READABLE EXTERNAL TABLE e_NATION (LIKE NATION)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/nation/') FORMAT 'csv';

CREATE READABLE EXTERNAL TABLE e_REGION (LIKE REGION)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/region/') FORMAT 'csv';

CREATE READABLE EXTERNAL TABLE e_PART (LIKE PART)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/part/') FORMAT 'csv';

CREATE READABLE EXTERNAL TABLE e_SUPPLIER (LIKE SUPPLIER)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/supplier/') FORMAT 'csv';

CREATE READABLE EXTERNAL TABLE e_PARTSUPP (LIKE PARTSUPP)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/partsupp/') FORMAT 'csv';

CREATE READABLE EXTERNAL TABLE e_CUSTOMER (LIKE CUSTOMER)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/customer/') FORMAT 'csv';

```

```
CREATE READABLE EXTERNAL TABLE e_ORDERS (LIKE ORDERS)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/orders/') FORMAT 'csv';

CREATE READABLE EXTERNAL TABLE e_LINEITEM (LIKE LINEITEM)
LOCATION ('qs://hashdata-public.pek3a.qingstor.com/tpch/1g/lineitem/') FORMAT 'csv';
```

2. 执行如下命令将保存在对象存储上面的 TPC-H 数据拷贝插入到数据仓库表中。

```
INSERT INTO NATION SELECT * FROM e_NATION;
INSERT INTO REGION SELECT * FROM e_REGION;
INSERT INTO PART SELECT * FROM e_PART;
INSERT INTO SUPPLIER SELECT * FROM e_SUPPLIER;
INSERT INTO PARTSUPP SELECT * FROM e_PARTSUPP;
INSERT INTO CUSTOMER SELECT * FROM e_CUSTOMER;
INSERT INTO ORDERS SELECT * FROM e_ORDERS;
INSERT INTO LINEITEM SELECT * FROM e_LINEITEM;
```

3. 现在可以开始运行样例查询了。

这里所采用的数据集和查询是商业智能计算测试 TPC-H。TPC-H 是美国交易处理效益委员会组织制定的用来模拟决策支持类应用的一个测试集。TPC-H 实现了一个数据仓库，共包含 8 个基本表，其数据量可以设定从 1G 到 3T 不等。在这个样例中，我们选择了 1G 的数据集。TPC-H 基准测试包括 22 个查询，其主要评价指标是各个查询的响应时间，即从提交查询到结果返回所需时间。这里只提供了前三条查询语句。关于 TPC-H 完整 22 条查询语句以及详细介绍可参考 [TPC-H 主页](#)。

```
-- This query reports the amount of business that was billed, shipped, and returned.

select
    l_returnflag,
    l_linenstatus,
    sum(l_quantity) as sum_qty,
    sum(l_extendedprice) as sum_base_price,
    sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
    sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
    avg(l_quantity) as avg_qty,
    avg(l_extendedprice) as avg_price,
    avg(l_discount) as avg_disc,
    count(*) as count_order
from
    lineitem
where
    l_shipdate <= '1998-12-01'
group by
    l_returnflag,
    l_linenstatus
order by
    l_returnflag,
    l_linenstatus;

-- This query finds which supplier should be selected to place an order for a given part in a given region.

select
```

```

s.s_acctbal,
s.s_name,
n.n_name,
p.p_partkey,
p.p_mfgr,
s.s_address,
s.s_phone,
s.s_comment
from
supplier s,
partsupp ps,
nation n,
region r,
part p,
(select p_partkey, min(ps_supplycost) as min_ps_cost
from
part,
partsupp ,
supplier,
nation,
region
where
p_partkey=ps_partkey
and s_suppkey = ps_suppkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'EUROPE'
group by p_partkey ) g
where
p.p_partkey = ps.ps_partkey
and g.p_partkey = p.p_partkey
and g. min_ps_cost = ps.ps_supplycost
and s.s_suppkey = ps.ps_suppkey
and p.p_size = 45
and p.p_type like '%NICKEL'
and s.s_nationkey = n.n_nationkey
and n.n_regionkey = r.r_regionkey
and r.r_name = 'EUROPE'
order by
s.s_acctbal desc,
n.n_name,
s.s_name,
p.p_partkey
LIMIT 100;

-- This query retrieves the 10 unshipped orders with the highest value.

select
l_orderkey,
sum(l_extendedprice * (1 - l_discount)) as revenue,
o_orderdate,
o_shipppriority
from
customer,
orders,
lineitem

```

```
where
    c_mktsegment = 'MACHINERY'
    and c_custkey = o_custkey
    and l_orderkey = o_orderkey
    and o_orderdate < '1995-03-15'
    and l_shipdate > '1995-03-15'
group by
    l_orderkey,
    o_orderdate,
    o_shippriority
order by
    revenue desc,
    o_orderdate
LIMIT 10;
```

步骤6: 寻找额外的资料和重设你的环境

完成这个入门教程后，你可以寻找更多额外的资料来学习和理解这个教程中介绍的概念，或者你可以将你的环境重置回原来的状态。如果你想尝试其他学习资料中提到的任务，你可以让集群一直运行着。不过，需要注意的是，只要集群还运行着，你将一直被收取费用。

额外学习资料

我们建议您可以通过如下资料来学习和理解这个教程中介绍的概念：

- HashData 数据仓库 [管理指南](#)
- HashData 数据仓库 [开发指南](#)

重置你的环境

当你完成了这个入门指南，你可以通过如下步骤重置你的环境：

- 删除样例集群。
- 删除步骤 3 中添加的防火墙规则和公网 IP。