

北京海量数据技术股份有限公司

**Vastbase G100 V2.2**

**(Build 10)**

**快速入门**

【版权声明】

©2007-2022 北京海量数据技术股份有限公司 版权所有

本文档著作权归 **北京海量数据技术股份有限公司**（简称“海量数据”）所有，未经海量数据事先书面许可，任何主体不得以任何形式复制、 修改、抄袭、传播全部或部分本文档内容。

北京海量数据技术股份有限公司保留所有的权利。

【服务声明】

本文档意在向客户介绍海量数据全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的产品、服务的种类、服务标准等应由您与海量数据之间的商业合同约定，除非双方另有约定，否则，海量数据对本文档内容不做任何明示或模式的承诺或保证。

**目录**

[1. 产品概述 3](#_Toc9132)

[2. 新手指引 3](#_Toc32641)

[3. 安装数据库 3](#_Toc32210)

[3.1. 安装概述 3](#_Toc17446)

[3.2. 准备环境 3](#_Toc2375)

[3.2.1. 防火墙设置 5](#_Toc7611)

[3.2.2. 设置网卡MTU值（可选） 6](#_Toc28744)

[3.2.3. 设置IPC参数（首次需配置） 6](#_Toc22170)

[3.2.4. 设置内核参数 7](#_Toc8360)

[3.2.5. 创建用户与目录 11](#_Toc18080)

[3.2.6. 修改资源限制 11](#_Toc5828)

[3.2.7. 设置Core\_Pattern 12](#_Toc13535)

[3.2.8. 预装依赖 12](#_Toc28902)

[3.2.9. 添加排序规则（可选） 13](#_Toc29687)

[3.3. 安装数据库 14](#_Toc30224)

[3.3.1. 字符安装（推荐） 14](#_Toc24069)

[3.3.2. 图形化安装 25](#_Toc15819)

[3.3.3. 镜像安装 38](#_Toc7159)

[3.3.4. 安装卸载FAQ 41](#_Toc31659)

[4. 升级说明 42](#_Toc21971)

[4.1. 使用VPatch进行版本控制 42](#_Toc25534)

[5. 启停数据库 53](#_Toc3141)

[6. 使用数据库 55](#_Toc25997)

[6.1. 使用vsql连接数据库 55](#_Toc17425)

[6.2. 数据库管理 57](#_Toc160)

[6.2.1. 创建数据库 57](#_Toc15631)

[6.2.2. 创建表 59](#_Toc541)

[6.2.3. 管理表 63](#_Toc4563)

[6.2.4. 查看对象 67](#_Toc7232)

[6.2.5. 配置运行参数 70](#_Toc3512)

# **产品概述**

Vastbase数据库是海量数据基于开源openGauss内核开发的企业级关系型数据库。融合多年对各行业应用场景的深入理解，Vastbase除了具有openGauss极致的性能以外，还在原有功能基础上增加了大量Oracle兼容特性，提供了极佳的业务连续性保障，并遵循EAL4+标准进行了大幅度的安全增强，同时整合GIS空间组件等一系列专业应用领域的企业级功能。

# 新手指引

本文通过以下模块，指导您安装和使用数据库：



* **安装数据库**：提供了字符安装（推荐）、图形化安装和镜像安装方式，您可以根据自身业务情况任选一种。
* **启停数据库**：启动、停止和重启数据库。
* **使用数据库**：了解如何连接和管理数据库，助您轻松上手Vastbase数据库。

# 安装数据库

## 安装概述

Vastbase支持单机部署和流复制集群两种方式。单机部署时，可在一个主机部署多个数据库实例，但为了数据安全，不建议您这样部署。流复制集群部署支持一台主机和最少一台备机。

## 准备环境

* 软件环境要求：

|  |  |  |
| --- | --- | --- |
| **Linux操作系统** | **处理器** | **版本** |
| Red Hat Enterprise Linux | x86\_64 | 7.3及以上 |
| CentOS | ARM、x86\_64 | 7.3及以上 |
| Oracle Enterprise Linux | ARM | 7.3及以上 |
| Kylin | ARM | 10及以上 |
| Linx（凝思） | x86\_64 | 6.0.80 |
| OpenEuler | ARM | 20.30 |

* 硬件环境要求：
* 开发测试环境：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **CPU** | **内存** | **本地存储** | **网络** | **机器数量** |
| Vastbase | 2核+ | 4GB+ | SAS，100G | 千兆 | 1 |

* 生产环境：Vastbase数据库服务器在生产环境中应具备的最低硬件配置要求如下。

|  |  |
| --- | --- |
| **项目** | **配置描述** |
| 最小内存 | * 功能调试32GB以上。 * 性能测试和商业部署时，单实例部署建议128GB以上。 * 复杂的查询对内存的需求量比较高，在高并发场景下，可能出现内存不足。该情况建议您使用大内存的机器，或使用负载管理限制系统的并发量。 * 集群搭建一主两备时，服务器内存配置推荐为8G及以上，否则会出现shared\_memory不够的情况。 |
| CPU | * 功能调试最小1×8 核 2.0GHz。 * 性能测试和商业部署时，单实例部署建议1×16核 2.0GHz。 * CPU超线程和非超线程两种模式都支持。但Vastbase各节点的设置需保持一致。 |
| 硬盘 | * 用于安装Vastbase的硬盘需最少满足如下要求： * 至少1GB用于安装Vastbase的应用程序包。 * 每个主机需大约300MB用于元数据存储。 * 预留70%以上的磁盘剩余空间用于数据存储。 * 建议系统盘配置为Raid1，数据盘配置为Raid5，且规划4组Raid5数据盘用于安装Vastbase。有关Raid的配置请参考硬件厂家的手册或互联网上的方法进行配置，其中Disk Cache Policy一项需要设置为Disabled，否则机器异常掉电后有数据丢失的风险。 * Vastbase支持使用SSD（固态硬盘）作为数据库的主存储设备，支持SAS接口和NVMe协议的SSD盘，以Raid的方式部署使用。 |

注：在实际产品中，硬件配置的规划需要考虑您的数据规模以及期望的数据库响应速度，请根据实际情况进行规划。

安装前准备

### 防火墙设置

为了确保Vastbase在防火墙开启的状态下能够正常使用，您需要开放数据库监听端口或者将Vastbase相关的服务、协议、IP和端口添加到各主机的防火墙白名单中。添加方法如下：

**步骤1** 以root用户登录操作系统。

**步骤2** 关闭防火墙和SELinux。

* 执行如下命令关闭防火墙。

systemctl status firewalld.service

systemctl stop firewalld.service

systemctl disable firewalld

* 执行如下命令关闭selinux。

#查看selinux是否关闭（diabled表示关闭；enforcing表示开启）

getenforce

#临时关闭

setenforce 0

**步骤3** 执行vi /etc/hosts命令，修改hosts文件，添加实际的IP地址和主机名。

IP地址 主机名

注：如果您想永久关闭SELinux，可参考如下方法：

1、执行如下命令打开config文件。

vim /etc/selinux/config

2、将SELINUX=enforcing修改为SELINUX=disabled。

3、执行如下命令重启系统。

reboot

### 设置网卡MTU值（可选）

将各数据库节点和交换机的网卡MTU值（最大传输单元）设置为相同大小（MTU值≥1500），推荐值：8192（MTU值可根据需要自行修改）。需要连同交换机一起修改（修改方法请咨询交换机厂商）。

**步骤1** 以root用户登录操作系统。

**步骤2** 执行如下命令，设置网卡MTU值。

方法一：

#格式：ifconfig 网卡编号 mtu 值；网卡编号可通过ip a命令查看。

ifconfig eth0 mtu 8192

方法二：

cat /sys/class/net/eth0/mtu

echo "8192" > /sys/class/net/eth0/mtu

### 设置IPC参数（首次需配置）

**步骤1** 以root用户登录操作系统。

**步骤2** 进入/etc/systemd/logind.conf文件，查看是否已经设置了RemoveIPC=no，如果没有则执行步骤3，否则跳过。

vi /etc/systemd/logind.conf

**步骤3** （可选）修改/etc/systemd/logind.conf文件中RemoveIPC=no。

echo "RemoveIPC=no" >> /etc/systemd/logind.conf

**步骤4** 进入/usr/lib/systemd/system/systemd-logind.service文件，查看是否已经设置了RemoveIPC=no，如果没有设置则执行步骤5，否则跳过。

vi /usr/lib/systemd/system/systemd-logind.service

**步骤5** （可选）添加RemoveIPC=no到/usr/lib/systemd/system/systemd-logind.service文件。

echo "RemoveIPC=no" >> /usr/lib/systemd/system/systemd-logind.service

systemctl daemon-reload

systemctl stop systemd-logind

systemctl start systemd-logind

### 设置内核参数

**前提条件**：当您需要性能测试时可设置内核参数。

**步骤1** 以root用户登录操作系统。

**步骤2** （可选）设置内核参数（逐一设置）。

**注**：您还可以直接执行步骤3，进行批量设置内核参数。

设置fs.aio-max-nr参数，该参数表示可以拥有的异步IO请求数目，推荐值：1048576。

sysctl fs.aio-max-nr='1048576'

* 设置file-max参数，该参数表示每次登录会话可以打开的文件数。

sysctl fs.file-max='76724600'

* 设置kernel.sem参数，该参数包含4个参数（排序为SEMMSL、SEMMNS、SEMOPM和MNI）。
* SEMMSL：每个信号量set中信号量最大个数，最小250。
* SEMMNS：Linux系统中信号量最大个数，至少32000，等于SEMMSL\*SEMMNI。
* SEMOPM：SEMOP系统调用允许的信号量最大个数，至少100；或者等于SEMMSL。
* SEMMNI：Linux系统信号量set最大个数，最少128。

sysctl kernel.sem='4096 2097152000 4096 512000'

* 设置kernel.shmall参数，该参数用于控制共享内存页数，等于系统内存（建议设置为80%，单位：byte）/PAGE\_SIZE（getconf PAGE\_SIZE获取），该参数设置太小有可能导致数据库启动报错。示例：物理内存为128GB，PAGE大小为4096，则kernel.shmall=1281024102410240.8/4096=26843545。

sysctl kernel.shmall='26843545'

* 设置kernel.shmmax参数，该参数表示最大单个共享内存段大小(建议为大于shared\_buffer值)， 等于系统内存0.5，单位：byte。示例：物理内存为128GB，则：kernel.shmmax=128102410241024\*0.5=68719476736。

sysctl kernel.shmmax='68719476736'

* 设置kernel.shmmni参数，该参数系统范围内共享内存段的最大数量，默认值： 4096。

sysctl kernel.shmmni='4096'

* 设置net.core.netdev\_max\_backlog参数，该参数表示允许送到队列的数据包的最大数目。

sysctl net.core.netdev\_max\_backlog='10000'

* 设置net.core.rmem\_default参数，该参数表示预留用于接收缓冲的内存默认值，单位：byte。

sysctl net.core.rmem\_default='262144'

* 设置net.core.rmem\_max参数，该参数表示预留用于接收缓冲的内存最大值，单位：byte。

sysctl net.core.rmem\_max='4194304'

* 设置net.core.wmem\_default参数，该参数表示预留用于发送缓冲的内存默认值，单位：byte。

sysctl net.core.wmem\_default='262144'

* 设置net.core.wmem\_max参数，该参数表示预留用于发送缓冲的内存最大值，单位：byte。

sysctl net.core.wmem\_max='4194304'

* 设置net.core.somaxconn参数，该参数表示socket监听的backlog（队列）上限，默认值：128。

sysctl net.core.somaxconn='4096'

* 设置net.ipv4.tcp\_fin\_timeout 参数，该参数表示FIN\_WAIT\_2状态的超时时长，单位：秒。

sysctl net.ipv4.tcp\_fin\_timeout='5'

* 设置vm.dirty\_background\_bytes参数，该参数表示触发回刷的脏页数据量，超过该参数，脏页刷到磁盘，单位：btye。

sysctl vm.dirty\_background\_bytes='409600000'

* 设置vm.dirty\_expire\_centisecs参数，该参数表示脏数据的过期时间，超过该时间系统会将脏数据回写到磁盘上，单位：百分之一秒，以下命令中3000代表30秒。

sysctl vm.dirty\_expire\_centisecs='3000'

* 设置vm.dirty\_ratio参数，该参数表示脏数据百分比，超过这个百分比，新的IO请求将会被阻挡，直到脏数据被写进磁盘。

sysctl vm.dirty\_ratio='80'

* 设置vm.dirty\_writeback\_centisecs参数，该参数表示多长时间，后台刷脏页进程会被唤醒一次，检查是否有缓存需要清理。单位：百分之一秒，以下命令中100代表1秒。

sysctl vm.dirty\_writeback\_centisecs='100'

* 设置vm.overcommit\_memory参数，该参数表示内存分配策略，可选值：0、1和2。
* 0：表示内核将检查是否有足够的可用内存供应用进程使用，如果有足够的可用内存，内存申请允许；否则，内存申请失败，并把错误返回给应用进程。
* 1：表示内核允许分配所有的物理内存，而不管当前的内存状态如何。
* 2：表示内核允许分配超过所有物理内存和交换空间总和的内存。

sysctl vm.overcommit\_memory='0'

* 设置vm.swappiness参数，该参数表示激活交换之前可用内存的百分比。数值越低，使用的交换越少，并且物理内存中保留的内存页越多，默认值：60。

sysctl vm.swappiness='60'

* 设置net.ipv4.ip\_local\_port\_range参数，该参数可以设置本地动态端口分配范围，防止占用监听端口。

sysctl net.ipv4.ip\_local\_port\_range='40000 65535'

* 设置fs.nr\_open参数，该参数表示单个进程可分配的最大文件数，单位：个。对于有很多对象（表、视图、索引、序列和物化视图等）的PostgreSQL数据库，建议设置为2000万。

sysctl fs.nr\_open='20000000'

**步骤3** 设置内核参数（批量设置），执行 vi /etc/sysctl.conf 将以下内核信息写入sysctl.conf文件末尾，保存退出。

fs.aio-max-nr=1048576

fs.file-max= 76724600

kernel.sem=4096 2097152000 4096 512000

kernel.shmall=26843545

kernel.shmmax=68719476736

kernel.shmmni = 4096

net.core.netdev\_max\_backlog = 10000

net.core.rmem\_default = 262144

net.core.rmem\_max = 4194304

net.core.wmem\_default = 262144

net.core.wmem\_max = 4194304

net.core.somaxconn = 4096

net.ipv4.tcp\_fin\_timeout = 5

vm.dirty\_background\_bytes = 409600000

vm.dirty\_expire\_centisecs = 3000

vm.dirty\_ratio = 80

vm.dirty\_writeback\_centisecs = 100

vm.overcommit\_memory = 0

vm.swappiness= 60

net.ipv4.ip\_local\_port\_range = 40000 65535

fs.nr\_open = 20000000

**步骤4** 重载配置，使其生效。

sysctl -p

### 创建用户与目录

**步骤1** 以root用户登录操作系统。

**步骤2** 创建数据库用户（可自定义），设定初始密码（需要重复输入2次且完全一致）。

useradd -m vastbase

passwd vastbase

注：自定义的用户名可用于安装数据库，文件属组和属主需要进行相应替换。

**步骤3** 创建数据库coredump目录。

mkdir -p /home/vastbase/data/vdb\_coredump

chmod 770 /home/vastbase/data

chown vastbase.vastbase /home/vastbase/data

**步骤4** （可选）创建数据库数据目录（可自定义）。

mkdir -p /home/vastbase/data/vastbase

chmod 700 /home/vastbase/data/vastbase

chown -R vastbase.vastbase /home/vastbase/data/vastbase

**步骤5** （可选）创建数据库软件目录（可自定义）

mkdir -p /home/vastbase/local/vastbase

chown -R vastbase.vastbase /home/vastbase

注：步骤4和步骤5不影响数据库安装。如不执行则安装数据库时将采用默认安装目录。

### 修改资源限制

**步骤1** 以root用户登录操作系统。

**步骤2** 执行 vi /etc/security/limits.conf ，在文件末尾添加如下内容，保存退出。

vastbase soft nproc unlimited

vastbase hard nproc unlimited

vastbase soft stack unlimited

vastbase hard stack unlimited

vastbase soft core unlimited

vastbase hard core unlimited

vastbase soft memlock unlimited

vastbase hard memlock unlimited

vastbase soft nofile 10240000

vastbase hard nofile 10240000

### 设置Core\_Pattern

为记录数据库异常停机信息，执行如下命令，设定Core\_pattern路径为Vastbase组用户可写的路径。

**步骤1** 以root用户登录操作系统。

**步骤2** 执行如下操作完成设置。

echo "/home/vastbase/data/vdb\_coredump/core-%e-%p-%t" > /proc/sys/kernel/core\_pattern

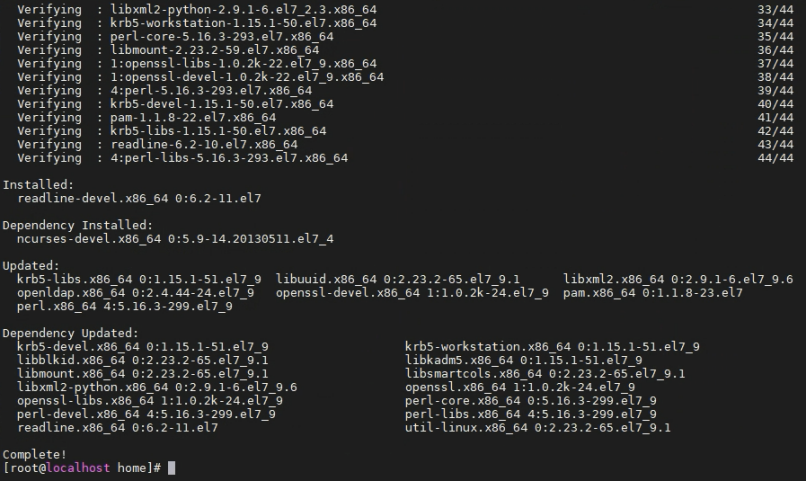
### 预装依赖

**步骤1** 以root用户登录操作系统。

**步骤2** 执行如下操作预装依赖（以CentOS 7为例）。

yum install -y readline python libicu cracklib libxslt tcl perl openldap pam systemd-libs bzip2 gettext libaio ncurses-libs

结果如下表示成功：



### 添加排序规则（可选）

**注意事项**：凝思系统需添加排序规则，其他系统不需要。

**背景信息**：在初始化实例时，数据库会将当前操作系统所有collation语言排序，并且将规则信息（通过命令"locale -a"查看）加入到pg\_collation表中。如果语言排序规则是在初始化实例之后才加入系统的，则不能自动加入到pg\_collation系统表中，在创建表等对象时如果指定了该排序规则，则会报该排序规则不存在。

例如：Linux操作系统默认安装时只有C、POSIX、zh\_CN.utf8等语言排序规则，如果需要数据库支持en\_US.utf8等其他语言排序规则，需要在初始化实例之前安装其语言排序规则。如果在初始化实例后再安装语言排序规则，则需要通过创建一个新的数据库，指定lc\_collate、lc\_ctype来使数据库支持新增的语言排序规则。

* Linux操作系统中，添加语言排序规则方法如下：
* **步骤1** 以root用户登录操作系统。
* **步骤2** 执行如下命令安装locales包。

apt-get install locales

* **步骤3** 执行如下命令，安装新的语言排序规则。

dpkg-reconfigure locales

* 注：也可使用如下命令安装en\_US.utf8。

localedef -v -c -i en\_US -f UTF-8 en\_US.UTF-8

* 凝思系统中，添加语言排序规则方法如下：
* **步骤1** 以root用户登录操作系统。
* **步骤2** 执行如下命令，勾选（按空格）中文en\_US.UTF-8编码。

sudo dpkg-reconfigure locales

* **步骤3** 执行如下命令，添加排序规则。

sudo apt-get install locales

* **步骤4** 执行如下命令，显示包含en\_US.UTF-8。

locale -a

* **步骤5** 重新启动Vastbase，即可创建包含en\_US.UTF-8数据库。

## 安装数据库

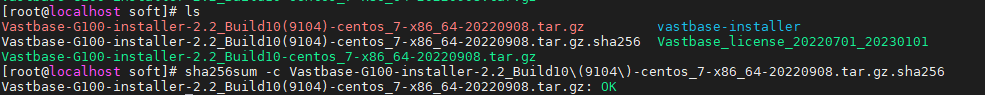
### 字符安装（推荐）

#### 校验安装包完整性

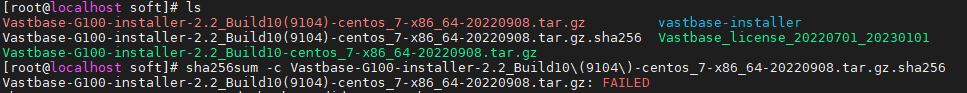
用户可使用安装包相应的sha256文件对安装包进行sha256校验码校验，安装包文件和安装包名.sha256需在同一文件夹里，执行以下命令：

sha256sum -c 安装包名.sha256

若校验成功，输出结果如下：



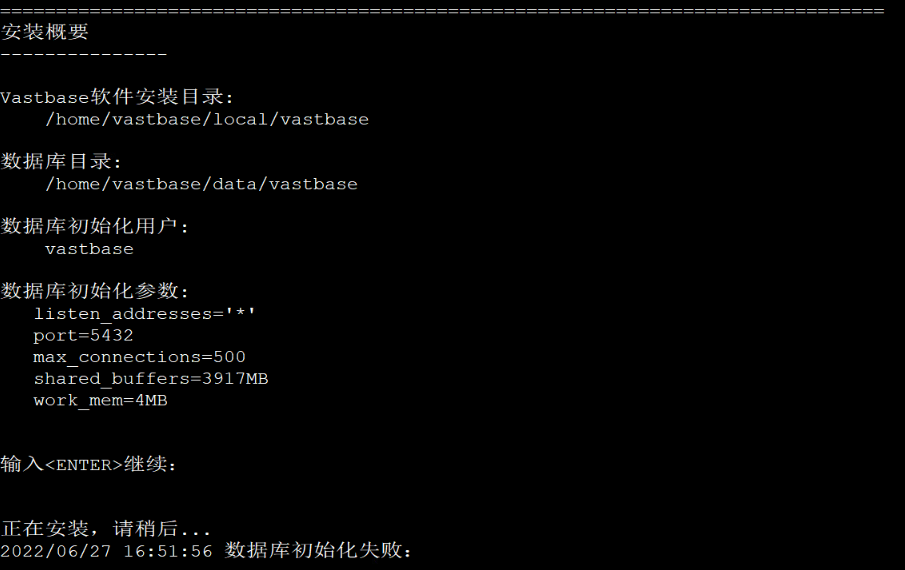
若校验失败，输出结果如下：



#### 实例化数据库安装

**注意事项：**

* 二进制安装程序与安装包需放于同一路径。
* 安装程序不能由操作系统超级用户执行。
* 若初始化失败且提示如下错误，请参见安装卸载FAQ章节。



安装步骤

**步骤1** 以root用户登录操作系统。

**步骤2** 创建目录（/soft/vb），解压安装包，以root用户赋予vastbase用户操作的权限。

注：将获取的安装包（本安装以Vastbase-G100-installer-2.2\_Build10-centos\_7-x86\_64-20220908.tar.gz

为例，实际安装以获取安装包名称为准）和license文件上传到/soft/vb（路径可自定义，本安装步骤以/soft/vb为例）。

mkdir -p /soft/vb

cd /soft/vb

tar -vxf Vastbase-G100-installer-2.2\_Build10-centos\_7-x86\_64-20220908.tar.gz

#vastbase是用户名，可自定义

chown -R vastbase:vastbase /soft/vb/

chmod -R 775 /soft/vb/

注：root用户对soft目录有755权限，否则切换到其他用户后，无法访问soft目录。

**步骤3** 切换到vastbase用户。

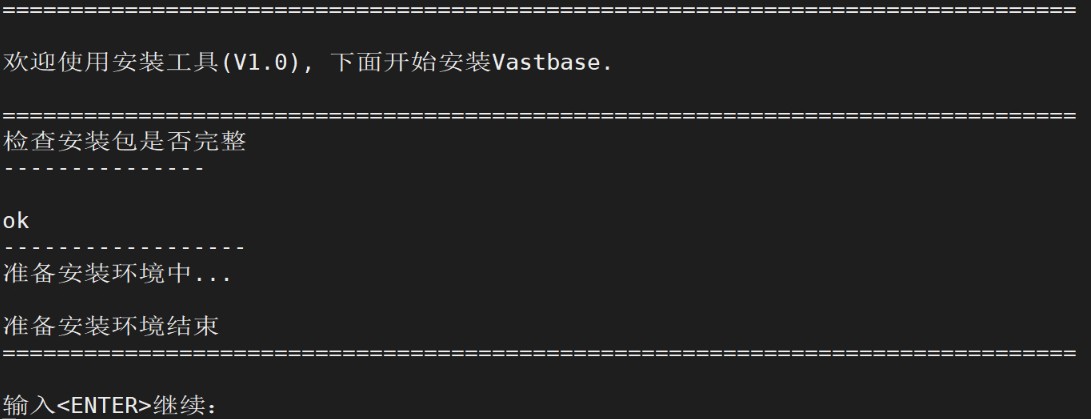
su - vastbase

**步骤4** 执行安装程序。

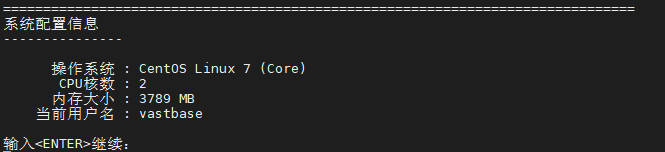
cd /soft/vb/vastbase-installer/

./vastbase\_installer

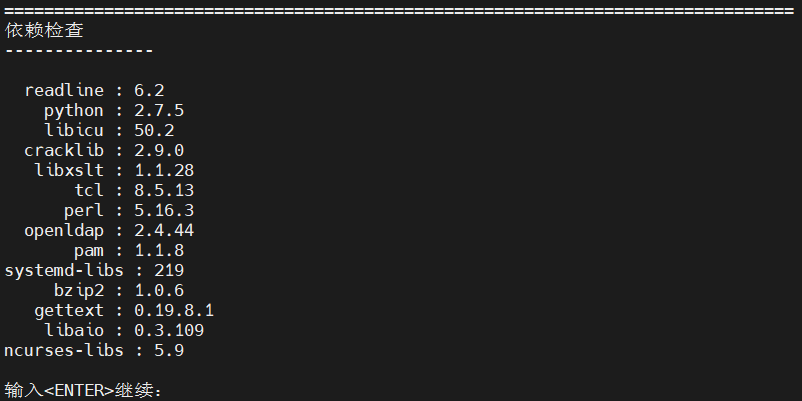
1. 安装环境检查。



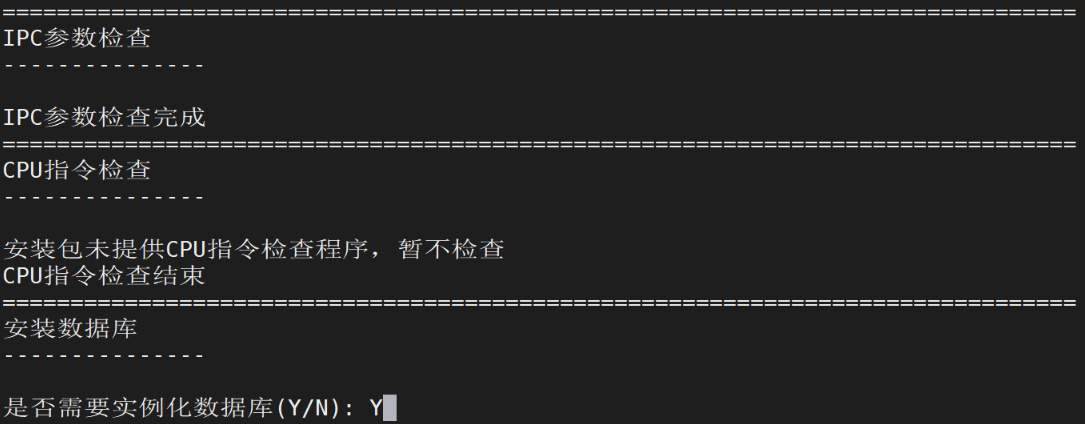
1. 系统配置信息。



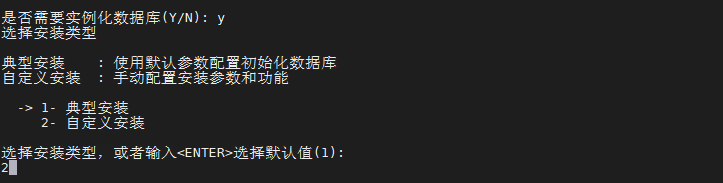
1. 依赖检查（检查服务器是否已经安装需要的依赖包）。



1. IPC参数检查（若检查通过，自动跳转下一步，否则根据提示进行设置即可）。
2. 安装数据库选项（选Y）。

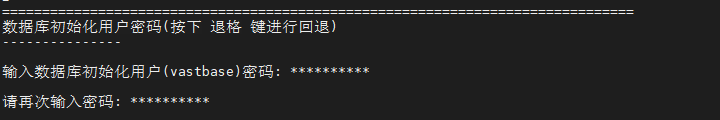


1. 选择安装类型（选2）。

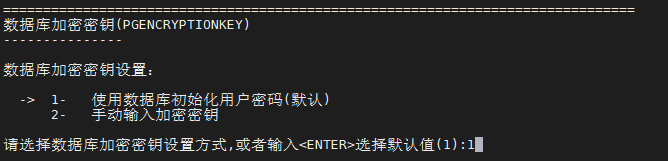


* 典型安装：使用默认参数配置初始化数据库。
* 自定义安装：手动配置安装参数和功能。

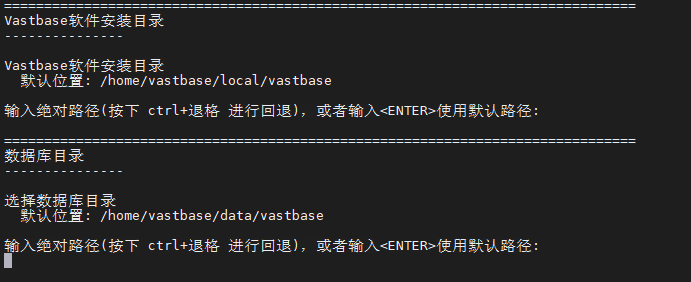
1. 设置超级管理员密码，需要输入密码，并再次输入密码确认（密码由大小写字母加数字组成，例如：aA123\*\*\*\*）



1. 设置密钥（选1）。



1. 设置数据库安装路径（输入步骤2创建的数据库安装路径（/soft/vb），或者输入<回车>使用默认路径（默认路径：/home/vastbase/local/vastbase）。



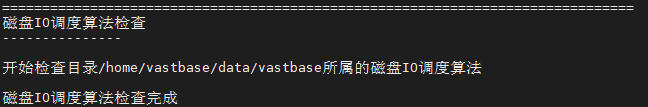
注：此处软件安装目录不能能数据库目录相同，目录相同时会自动创建子目录用于分开存放软件和数据

1. 参数配置

* 监听端口：默认5432。
* 最大连接数：默认500。
* 共享内存：默认为系统内存的1/4。

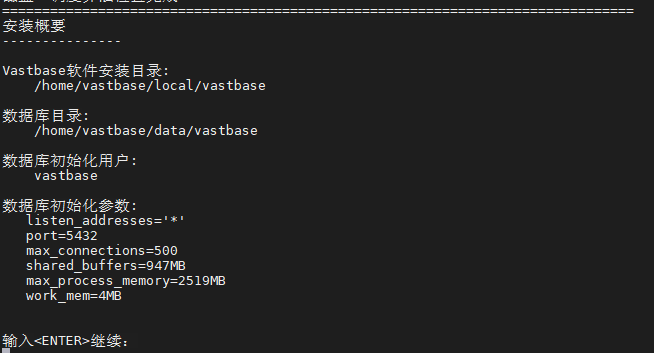


1. 磁盘IO调度算法检查。

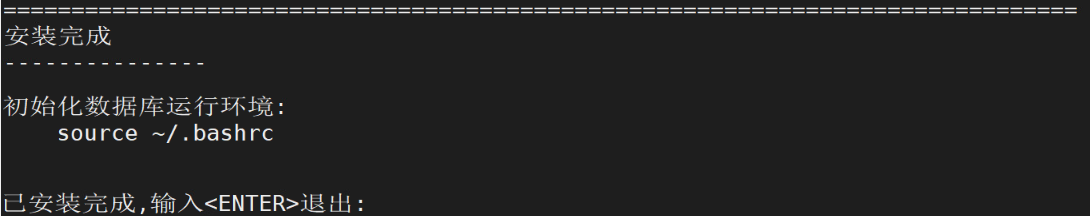


注：为避免影响数据库性能，生产环境建议采用deadline策略，若检查出非deadline策略，可按提示进行修改，再重新安装，也可以直接跳过继续安装。

1. 安装概要查看，核对信息。



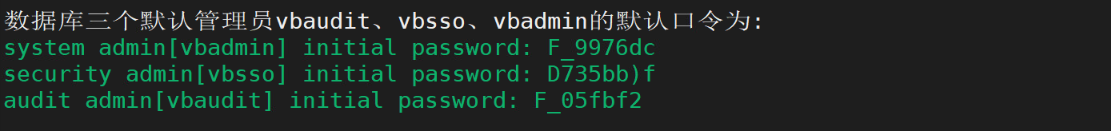
1. 安装数据库（提示安装完成）。



注：

* 如果您的数据库版本大于2.2（Build 5.75）（版本可通过安装包installer和Build后面的数字按顺序组合后得出版本号。示例：安装包名称：Vastbase-G100-installer-2.2\_Build\_5.39-centos\_7-x86\_64-2021\*\*\*\*.tar.gz，则版本号：2.2（Build 5.39））。
* 安装数据库过程会生成随机口令作为管理员初始口令，当您首次登录数据库执行sql命令时系统会强制要求您修改用户密码，修改方法请参见：《[VASTDATA]Vastbase G100 V2.2 管理员指南》->1.数据库使用->1.6连接数据库->1.6.2使用vsql连接章节。

显示信息如下：



**初始化数据库运行环境**

**步骤1** 以vastbase用户登录操作系统。

**步骤2** 执行如下命令初始化数据库。

source ~/.bashrc

**步骤3** 将license路径写入postgresql.conf配置文件，以确定license文件的位置（license文件路径为绝对路径，且需要指定到文件名级别，可自定义路径，但需要确保vastbase用户对license文件目录有读权限）。

echo "license\_path='/soft/Vastbase\_license\_20220701\_20230101'" >> $PGDATA/postgresql.conf

**步骤4** （可选）如数据库为2.2（Build 5.75）及以上版本，执行如下命令。

echo "password\_force\_alter=off" >> $PGDATA/postgresql.conf

#### 非实例化数据库安装

**注意事项**

* 二进制安装程序与安装包需放于同一路径。
* 安装程序不能由操作系统超级用户执行。
* 若后续初始化失败且提示如下错误，请参见安装卸载FAQ章节。



**安装步骤**

**步骤1** 以root用户登录操作系统。

**步骤2** 创建目录（/soft/vb），解压安装包，以root用户赋予vastbase用户操作的权限。

注：将获取的安装包（本安装以Vastbase-G100-installer-2.2\_Build\_5.39-centos\_7-x86\_64-2021\*\*\*\*.tar.gz为例，实际安装以获取安装包名称为准）和license文件上传到/soft/vb（可自定义，本安装步骤以/soft/vb为例）。

mkdir -p /soft/vb

cd /soft/vb

tar vxf Vastbase-G100-installer-2.2\_Build10-centos\_7-x86\_64-20220908.tar.gz

#需按照实际情况更换安装包名称

chown -R vastbase:vastbase /soft/vb/

chmod -R 775 /soft/vb/

**步骤3** 切换到vastbase用户。

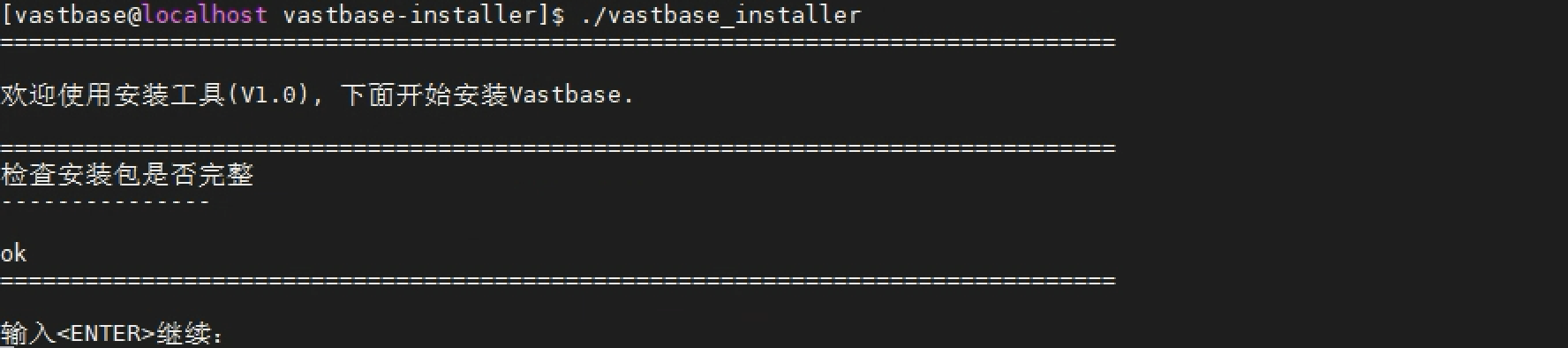
su - vastbase

**步骤4** 执行安装程序。

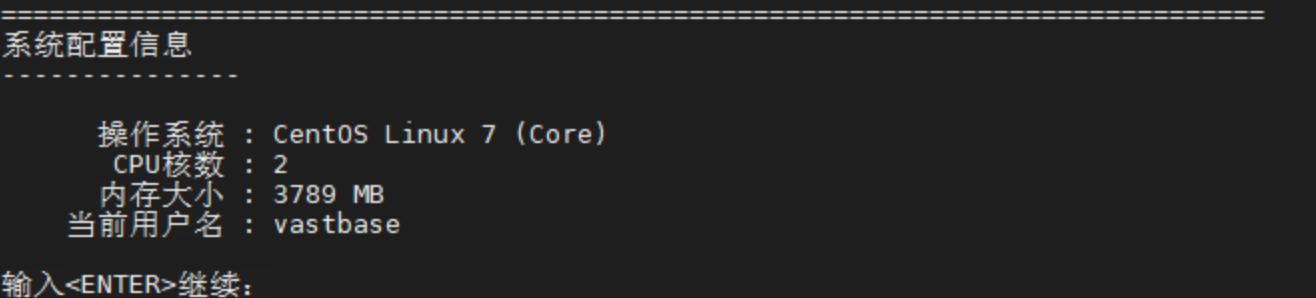
cd /soft/vb/vastbase-installer/

./vastbase\_installer

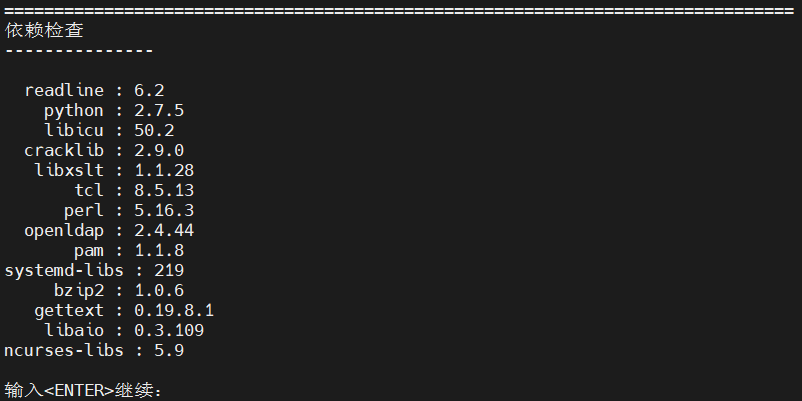
1. 安装环境检查。



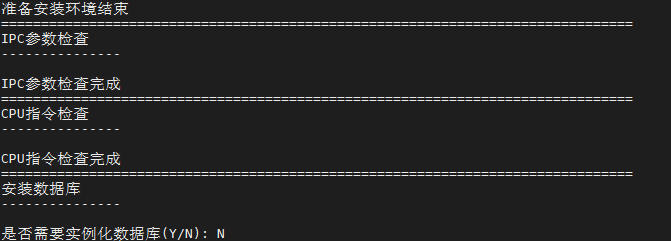
1. 系统配置信息。



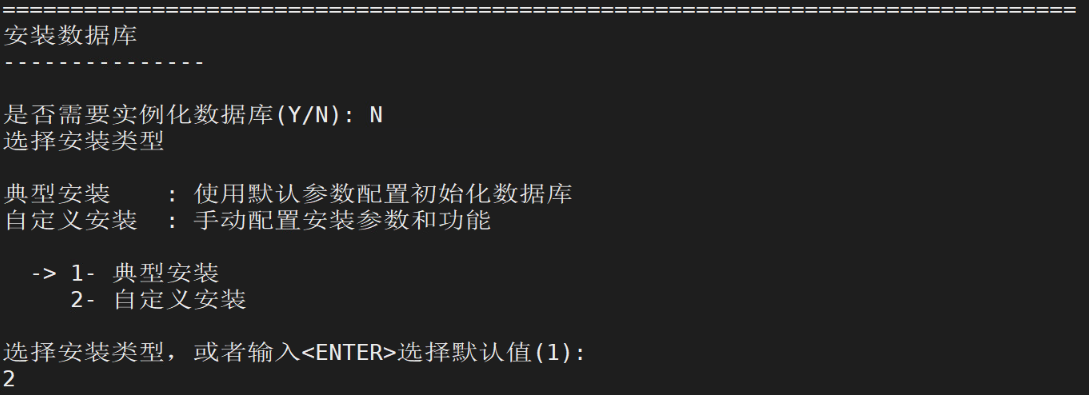
1. 依赖检查（检查服务器是否已经安装需要的依赖包）。



1. IPC参数检查（若检查通过，自动跳转下一步，否则根据提示进行设置即可）。
2. 安装数据库选项（选N）。

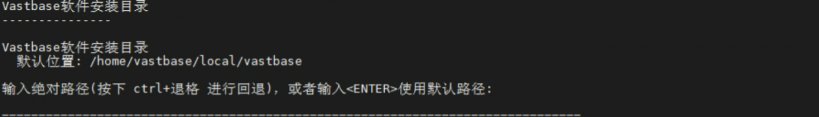


1. 选择安装类型（选2）。

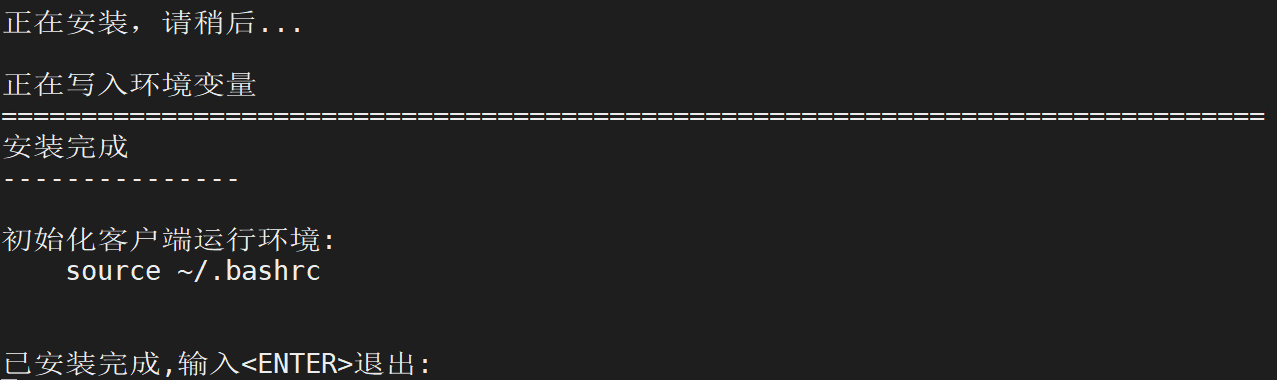


* 典型安装：使用默认参数配置初始化数据库。
* 自定义安装：手动配置安装参数和功能。

1. 设置数据库安装路径（输入步骤2创建的数据库安装路径（/soft/vb），或者输入<回车>使用默认路径（默认路径：/home/vastbase/local/vastbase））。



1. 安装（提示安装完成）。



**初始化数据库运行环境**

**步骤1** 以vastbase用户登录操作系统。

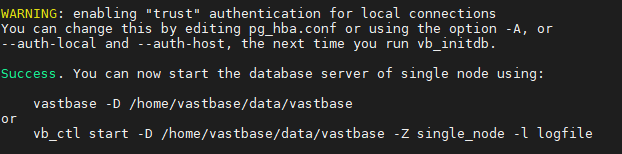
**步骤2** 执行如下命令初始化数据库。

source ~/.bashrc

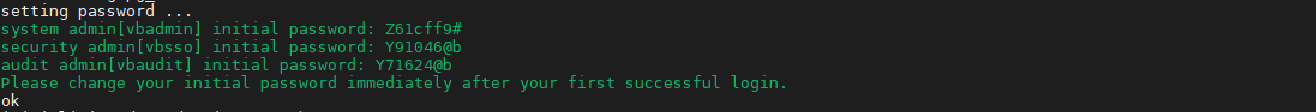
**步骤3** 初始化数据库。

vb\_initdb -D /home/vastbase/data/vastbase --nodename vastbase -w vbase@123

注：/home/vastbase/data/vastbase为数据库目录，可自定义，需存在且在vastbase用户下，需要0700权限。提示如下信息，表示正确。



注：如果您的数据库版本大于2.2（Build 5.75），初始化数据库时生成的管理员随机口令，当您首次登录数据库执行sql命令时系统会强制要求您修改用户密码，修改方法请参见：《[VASTDATA]Vastbase G100 V2.2 管理员指南》->1.数据库使用->1.6连接数据库->1.6.2使用vsql连接章节。

显示信息如下：

**步骤4** 将license路径写入postgresql.conf配置文件，以确定license文件的位置（license文件路径为绝对路径，且需要指定到文件名级别，可自定义路径，但需要确保vastbase用户对license文件目录有读权限）。

echo "license\_path='/soft/Vastbase\_license\_20220701\_20230101'" >> $PGDATA/postgresql.conf

**步骤5** （可选）如数据库为2.2（Build 5.75）及以上版本，执行如下命令。

echo "password\_force\_alter= off" >> $PGDATA/postgresql.conf

#### 卸载数据库

卸载Vastbase的过程包含卸载Vastbase和对Vastbase服务器的环境清理。

**前提条件：**关闭数据库。

**卸载Vastbase**

**步骤1** 以操作系统用户vastbase登录数据库节点。

su - vastbase

**步骤2** 进入安装程序所在目录。

cd /soft/vb/vastbase-installer/

**步骤3** 执行卸载命令。

./vastbase\_installer --uninstall

注：卸载完成后，已安装的Vastbase程序（即$GAUSSHOME对应目录）和Vastbase相关环境变量将被删除，仅保留数据库实例。

### 图形化安装

#### 安装DB Installer系统

**支持的环境：**Vastbase集群支持在x86\_64和ARM平台搭建，目前支持的操作系统如下：

|  |  |
| --- | --- |
| **平台** | **操作系统** |
| x86\_64 | CentOS 7.x |
| ARM | Kylin V10 SP2 |

**软件环境准备：**

* **安装java环境。**

**步骤1** 以root用户登录操作系统。

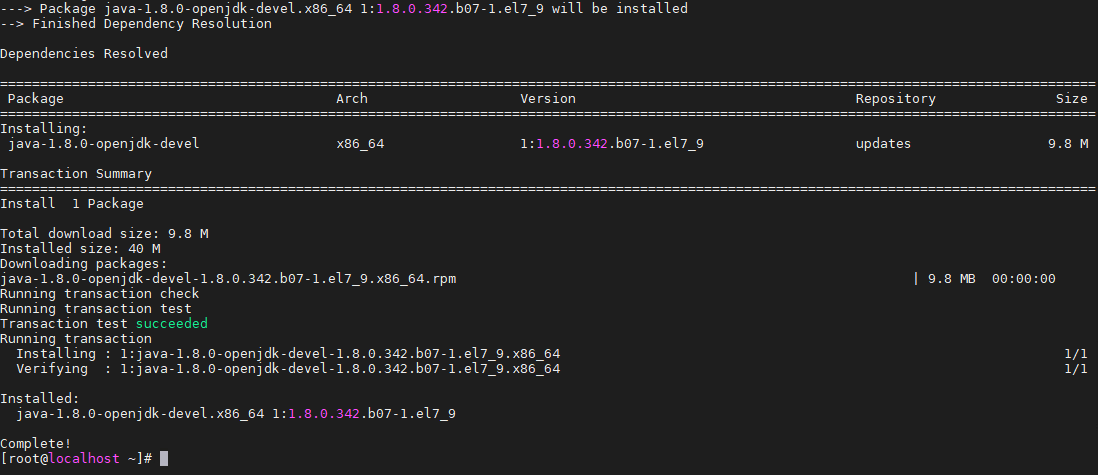
**步骤2** 执行如下命令，查找系统资源库中所有可以安装的java包。

yum list | grep java

**步骤3** 安装java运行环境（版本号需要≥1.8.0，以下安装以java-1.8.0-openjdk-devel.x86\_64为例）。

yum install -y java-1.8.0-openjdk-devel.x86\_64

安装成功显示如下：



**步骤4** 在/etc/profile文件中添加如下信息，设置环境变量。

#JAVA\_HOME为安装java的路径

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.342.b07-1.el7\_9.x86\_64

export PATH=$PATH:$JAVA\_HOME/bin

**步骤5** 执行如下命令使环境变量生效。

source /etc/profile

* 安装单机数据库需要安装的依赖包括：

zlib、libuuid、readline、krb5-libs、cracklib、openldap、pam、systemd-libs、libxml2、libicu、libxslt、tcl、perl、bzip2、gettext、openssl、libaio、gcc

* 安装集群数据库需要安装的依赖包括：

Zlib、libuuid、readline、krb5-libs、cracklib、openldap、pam、systemd-libs、libxml2、libicu、libxslt、tcl、perl、bzip2、gettext、openssl、libaio、ntpdate、unzip、gcc、postgresql-devel

* **安装设置ntp服务。**

集群节点将会配置时间同步主机，需要预先在集群机器外单独配置一个时间同步主机，该主机需要预先安装ntp服务。

**步骤1** 以root用户登录操作系统。

**步骤2** 安装ntp。

yum install ntp

**步骤3** 配置ntp服务，执行“vi /etc/ntp.conf”添加如下信息，以本地时间作为时间服务。

server 127.127.1.0

fudge 127.127.1.0 stratum 10

**步骤4** 启动ntp服务。

systemctl enable ntpd

systemctl start ntpd

**步骤5 （可选）**开放udp 123端口，该步骤在防火墙关闭的情况下可直接跳过。

ntp服务主机需要开放udp 123端口（ntp的默认端口）。执行该步骤需要需要启动防火墙 FirewallD服务。

sudo firewall-cmd --zone=public --add-port=123/udp --permanent

**步骤6** 重载防火墙。

sudo firewall-cmd --reload

**部署实施DB Installer：**

**步骤1** 将DB Installer安装包上传至任意目录（以/root为例）。安装包格式为：vastbase\_cluster\_\*\*\*\*.tar.gz，其中\*\*\*\*代表版本号。例如vastbase\_cluster\_trunk-centos\_x86\_64\_2022092117.tar.gz。

**步骤2** 进入上传DB Installer安装包路径。

cd /root

**步骤3** 解压安装包。

tar -zxvf vastbase\_cluster\_trunk-centos\_x86\_64\_2022092117.tar.gz

**步骤4** 进入应用jar包所在目录。

cd /root/hainstaller/installer

**步骤5** 启动db installer后台服务。

nohup java -jar ha\_installer-1.1-SNAPSHOT.jar >> run.log 2>&1 &

**步骤6** 查看日志确定访问端口。

vi /root/hainstaller/installer/run.log

日志中如下信息显示端口为7070。

29

**步骤7** 上传license，上传路径为：/root/hainstaller/installer/lic/（注意：license文件上传之后名称需更改为“vastbaselicense”）。

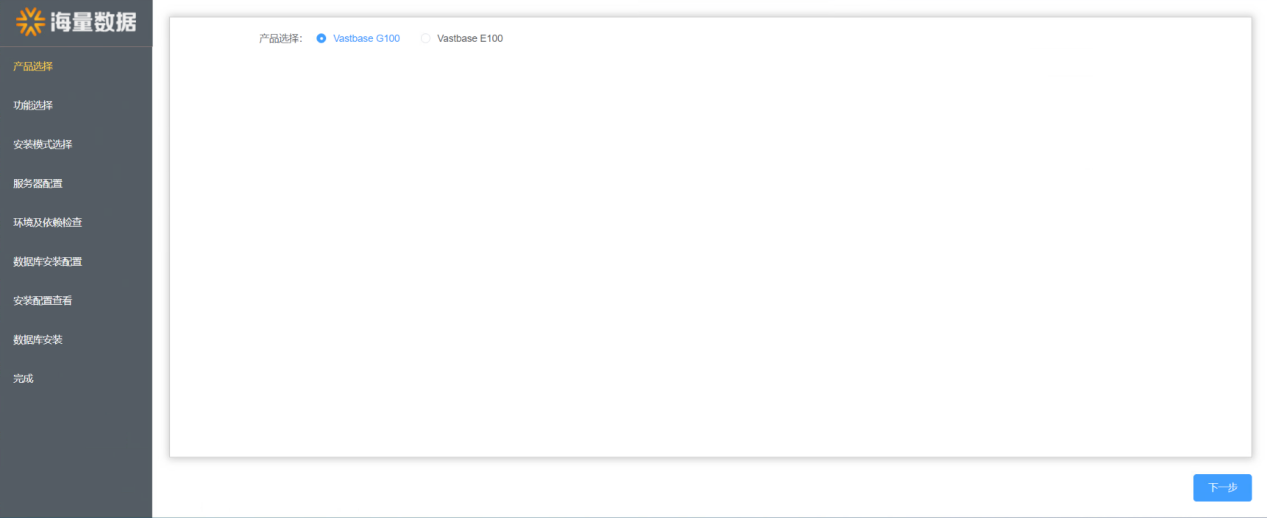
**步骤8** 访问DB Installer系统。

IP（服务器IP）:访问端口（日志中显示的端口）

例如：172.16.101.219:7070。

#### 单机数据库安装

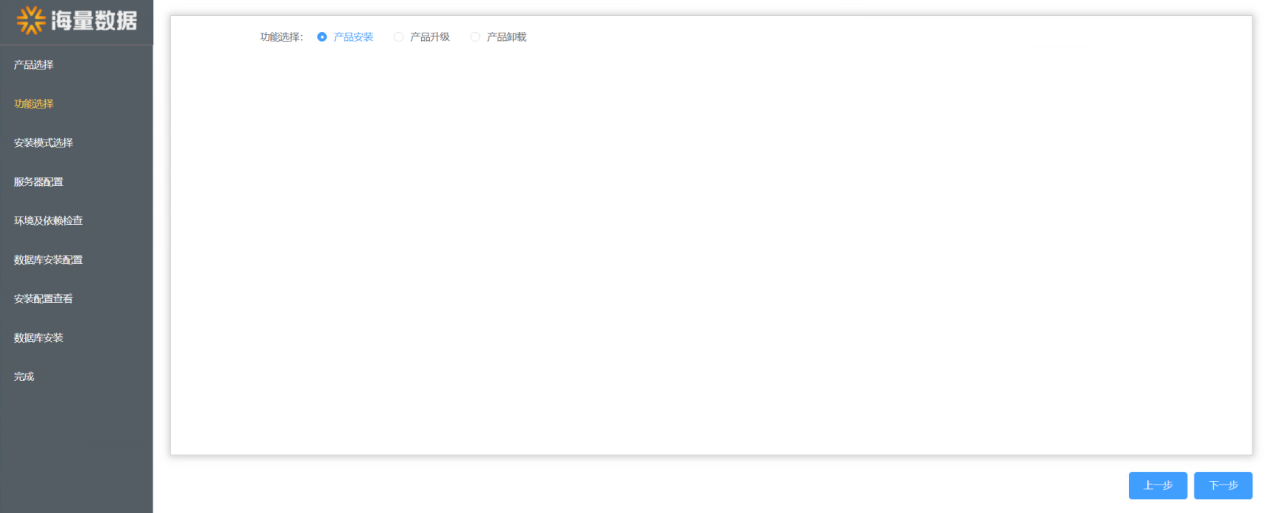
**步骤1** 左侧产品选择页，选择Vastbase G100，单击下一步。



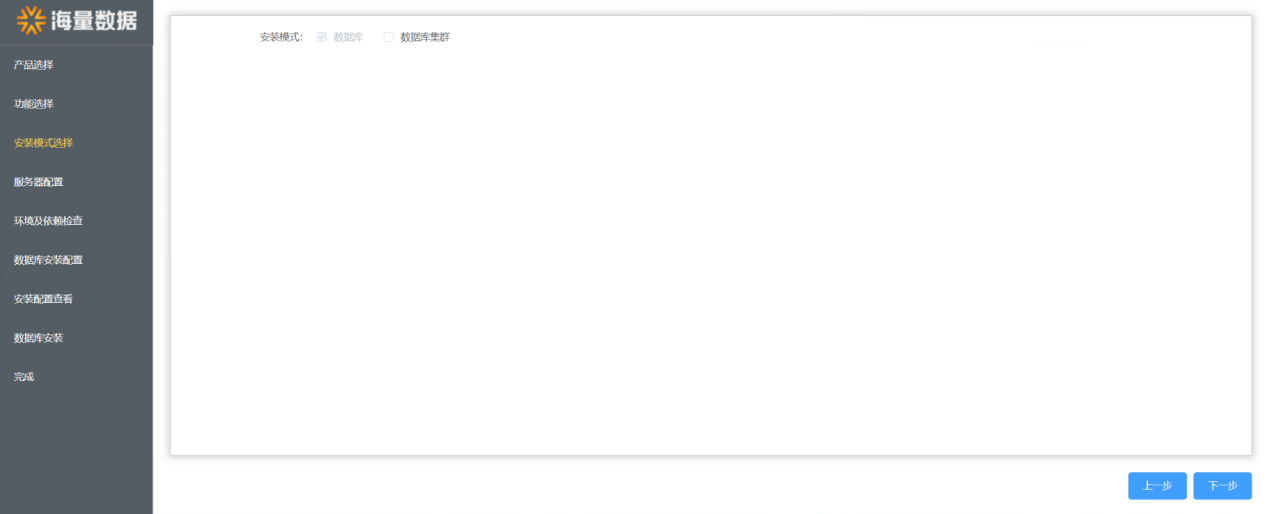
注：Vastbase G100支持安装、卸载和升级。

**步骤2** 左侧功能选择页，选择产品安装，单击下一步。

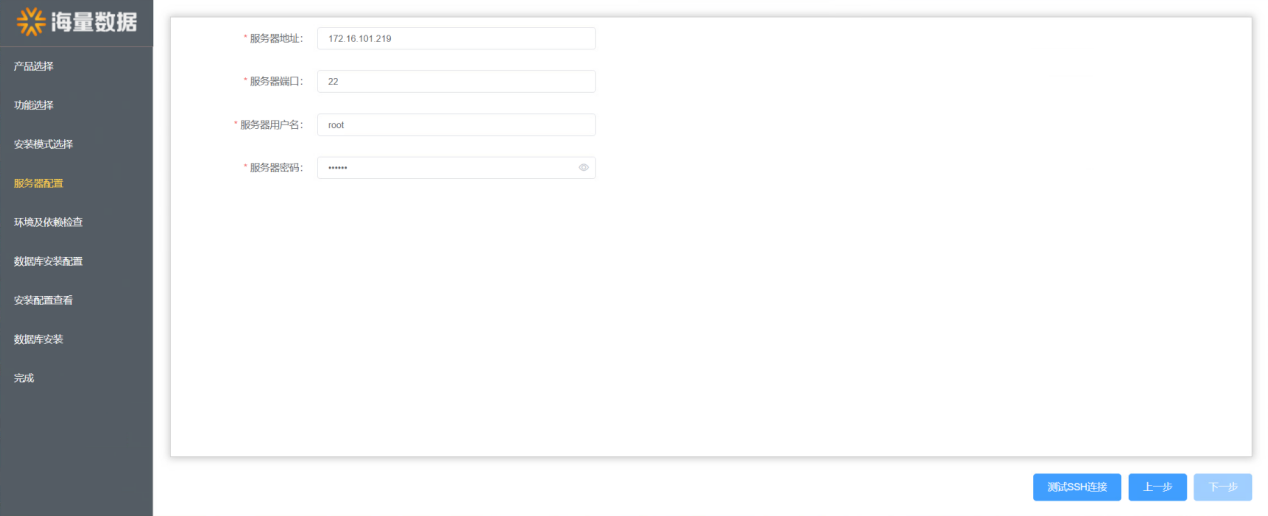
* 产品安装：用于安装数据库。
* 产品升级：用于升级数据库。
* 产品卸载：用于卸载数据库。



**步骤3** 左侧安装模式选择页，选择安装模式为数据库（默认模式），单击下一步。

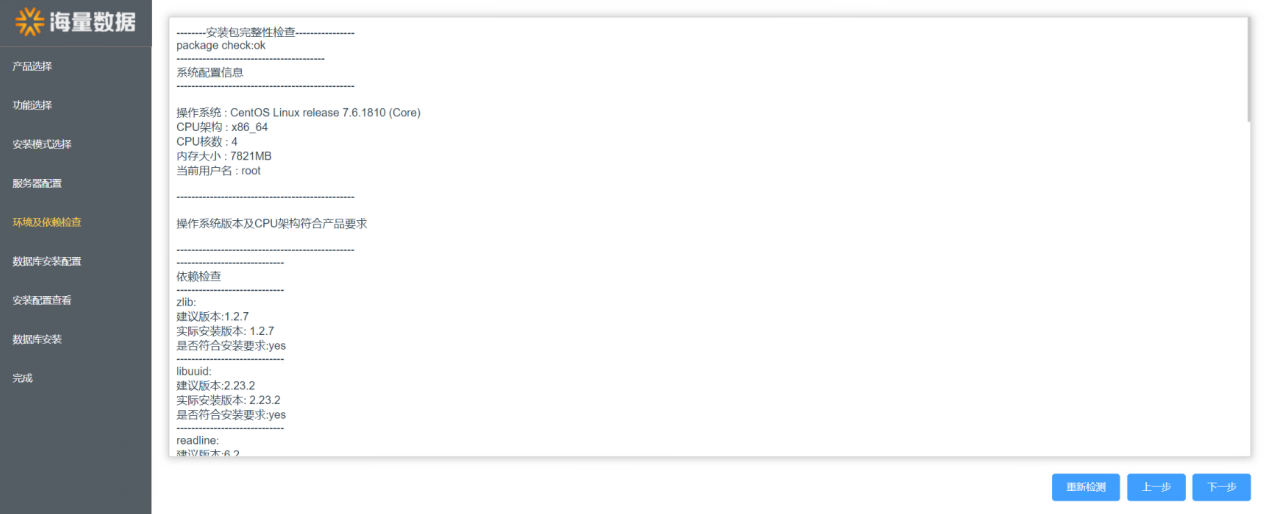


**步骤4** 左侧服务器配置页，填写下表中信息后，单击测试SSH连接，显示成功弹窗后单击下一步。

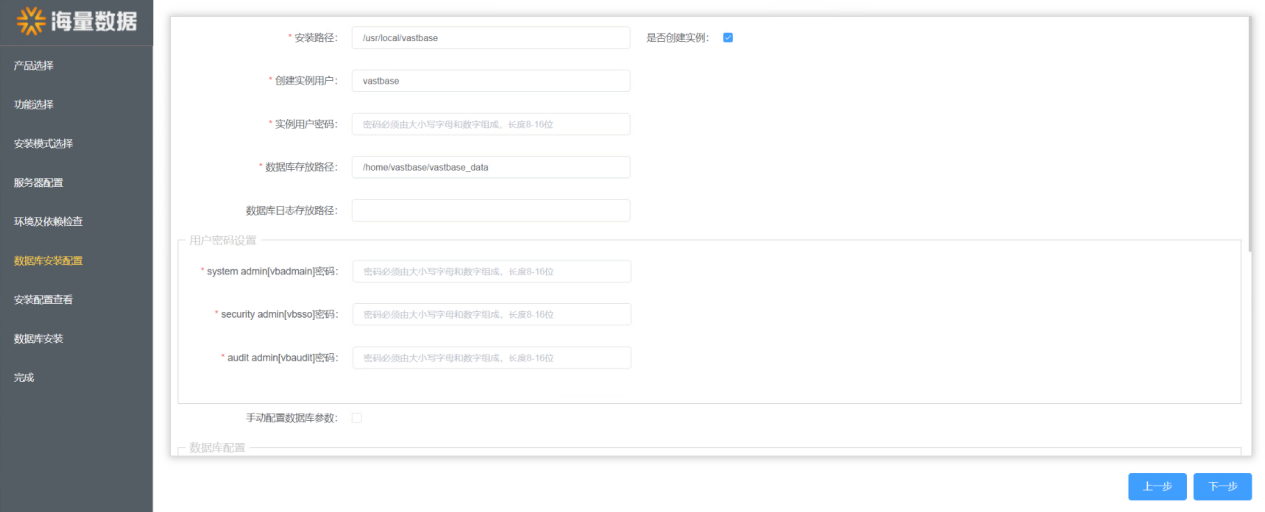


|  |  |  |
| --- | --- | --- |
| **参数** | **描述** | **样例** |
| 服务器地址 | 安装数据库的服务器地址。 | 172.16.101.219 |
| 服务器端口 | 安装数据库的端口。 | 22 |
| 服务器用户名 | 连接安装数据库服务器的用户名（需要使用root等超级用户）。 | root |
| 服务器密码 | 连接安装数据库服务器的密码。 | \*\*\*\* |

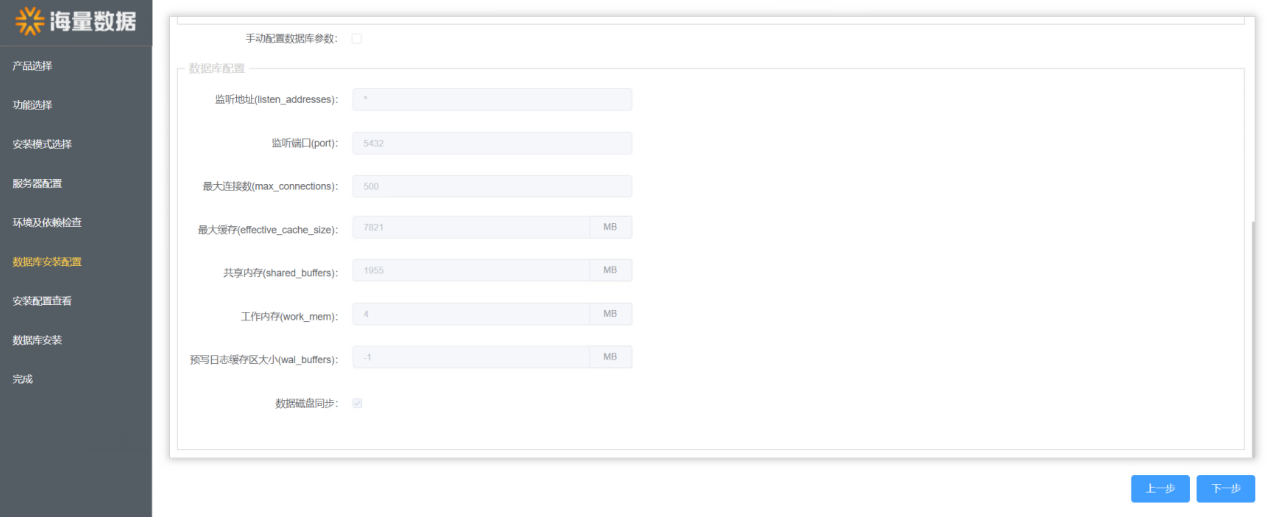
**步骤5** 左侧环境依赖性检查页，检查通过后（您还可以通过右侧滚动条来查看完整的检查结果），单击下一步。



**步骤6** 左侧数据库安装配置页，配置完成后单击下一步。



注：图形安装界面的密码由大小写字母和数字组成，长度为8-16位，目前图形安装不支持特殊字符，而字符界面是支持的。

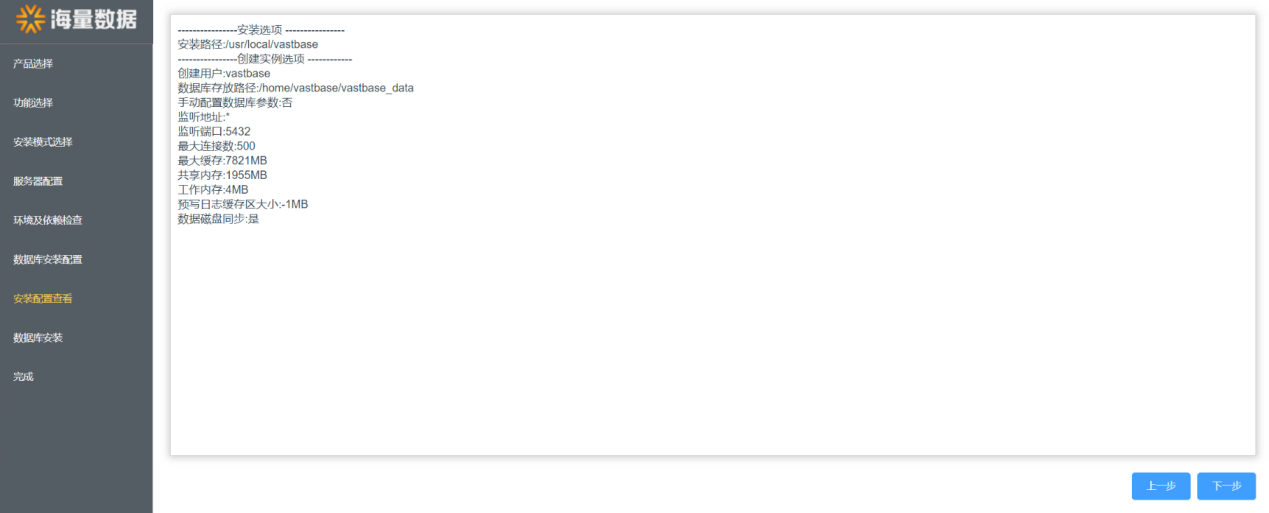


|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **描述** | **样例** | **是否必填** |
| 安装路径 | 安装数据库路径 | /home/\* | 是 |
| 是否创建实例 | 是否创建实例 | / | 否 |
| 创建实例用户 | 实例用户名 | vastbase | 是 |
| 实例用户密码 | 实例密码 | \*\*\*\* | 是 |
| 数据库存放路径 | 数据库存放路径 | /home/\*/\*\_data | 是 |
| 数据库日志存放路径 | 数据库日志存放路径 | / | 否 |
| system admin密码 | system admin密码 | / | 是 |
| security admin密码 | security admin密码 | / | 是 |
| audit admin密码 | audit admin密码 | / | 是 |
| 监听地址 | 监听地址 | \* | 否 |
| 监听端口 | 监听端口 | 57432 | 否 |
| 最大连接数 | 最大连接数 | 500 | 否 |
| 最大缓存 | 最大缓存 | 1838 | 否 |
| 共享内存 | 共享内存 | 300 | 否 |
| 工作内存 | 工作内存 | 4 | 否 |
| 预写日志缓存区大小 | 预写日志缓存区大小 | -1 | 否 |
| 数据磁盘同步 | 数据磁盘同步 | 是 | 否 |

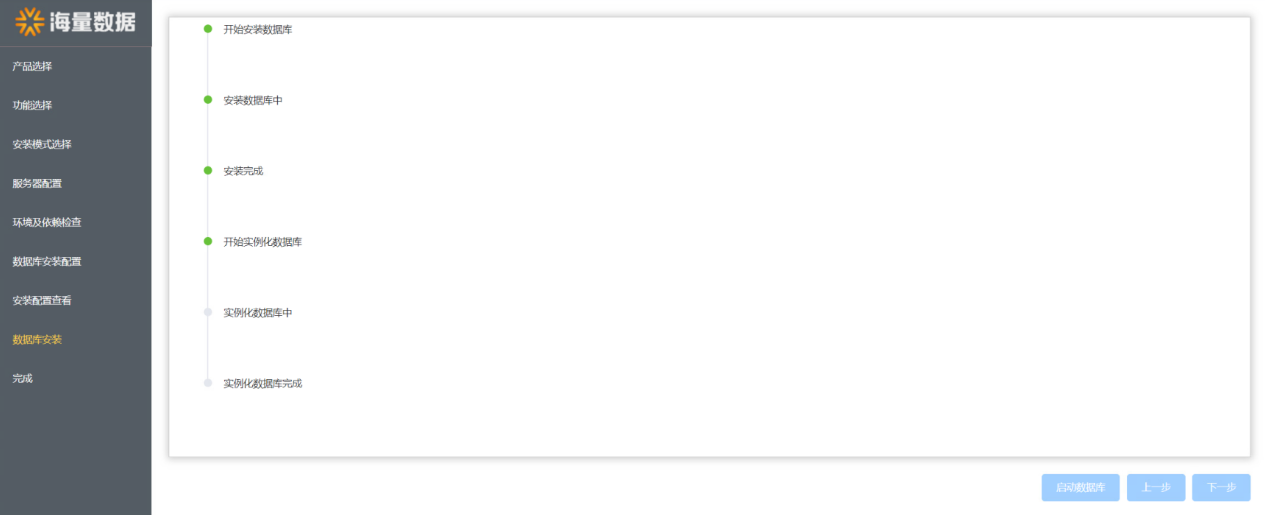
注：

* 单机数据库安装时，可填写\*号，表示使用默认值。
* 密码必须包含大小字母及数字，长度8~16位。
* 勾选“手动配置数据库参数”选项才可以修改数据库配置。

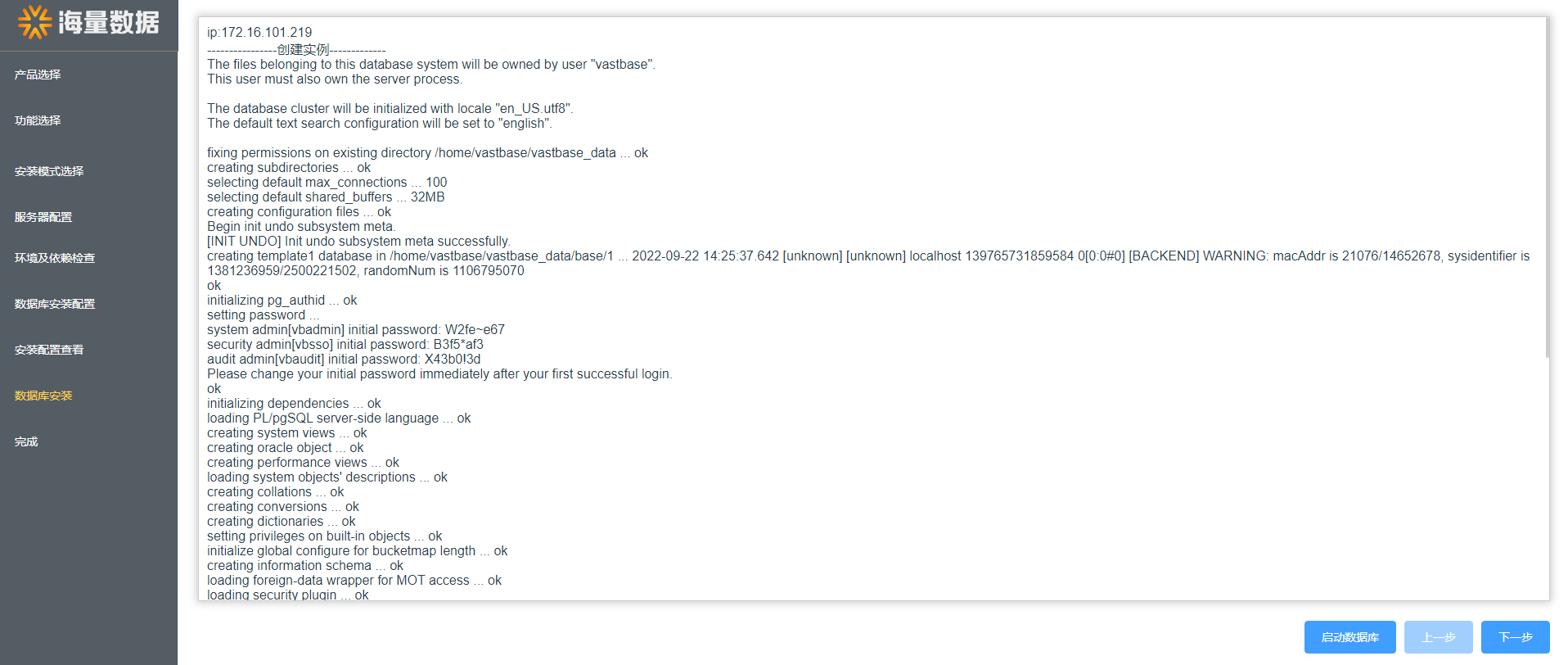
**步骤7** 左侧安装配置查看页，确认显示信息无误后单击下一步。



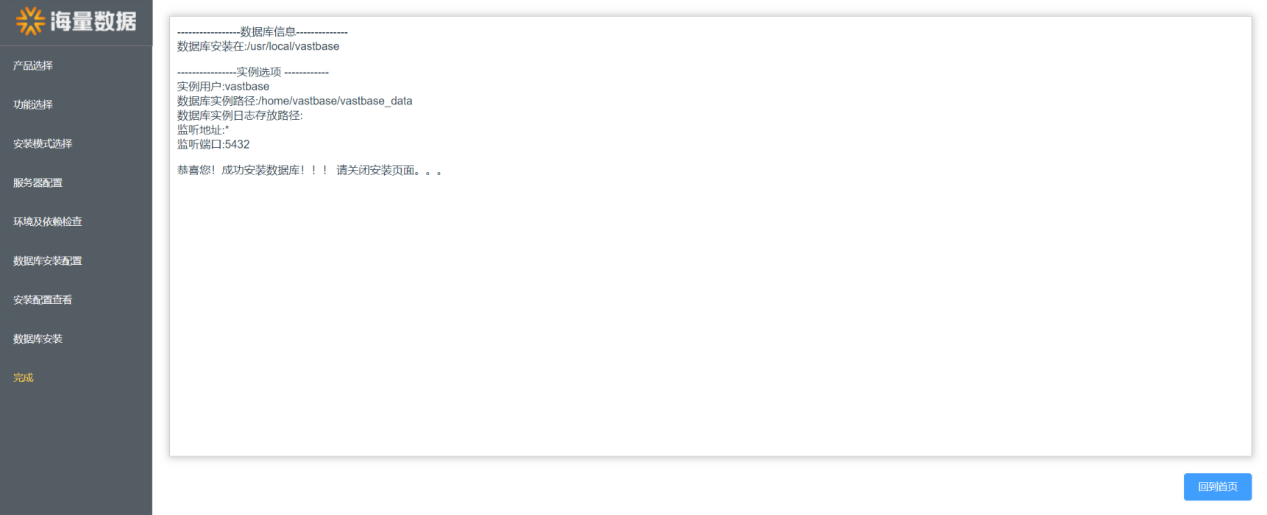
数据库安装过程显示如下：



**步骤8** 左侧数据库安装页，完成数据库实例化后显示如下信息（您还可以通过右侧滚动条来查看完整安装信息），单击启动数据库或者单击下一步。



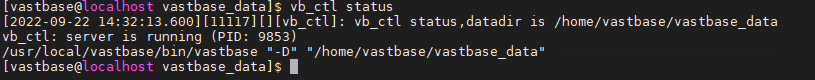
**步骤9**左侧完成页，单击回到首页，完成安装。



图形界面点击启动数据库界面返回命令行查看数据库状态。

vb\_ctl status

返回结果如下：



如下命令分别为启动/停止/重启Vastbase数据库的命令。

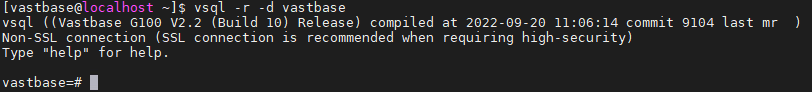
vb\_ctl <start/stop/restart>

**登录数据库**

实例化数据库过程中会默认创建数据库超级用户vastbase，执行命令登录数据库。

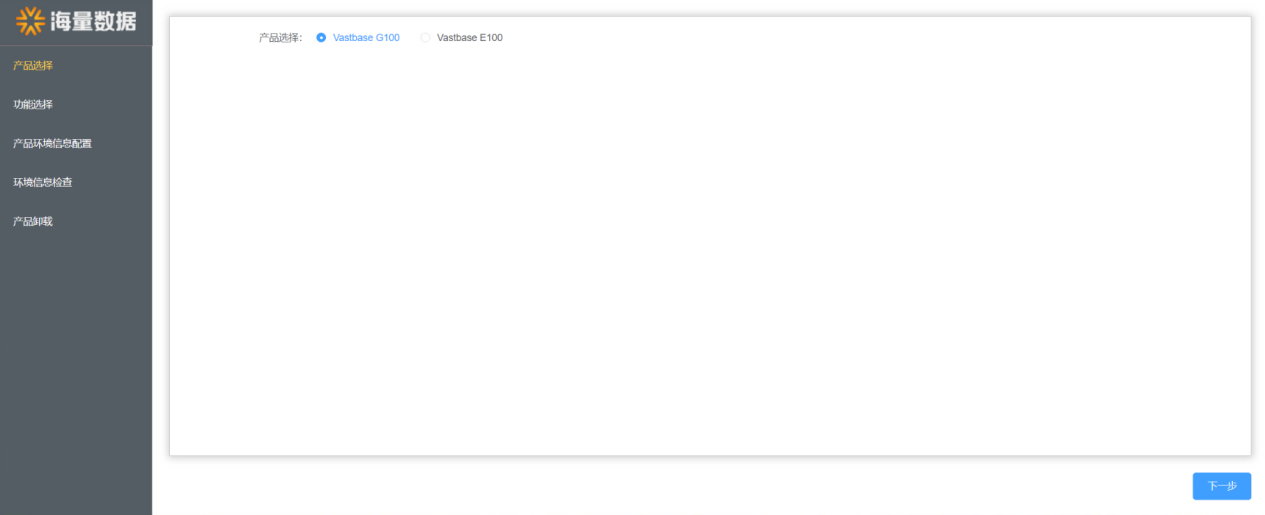
vsql -r -d vastbase

结果显示如下：

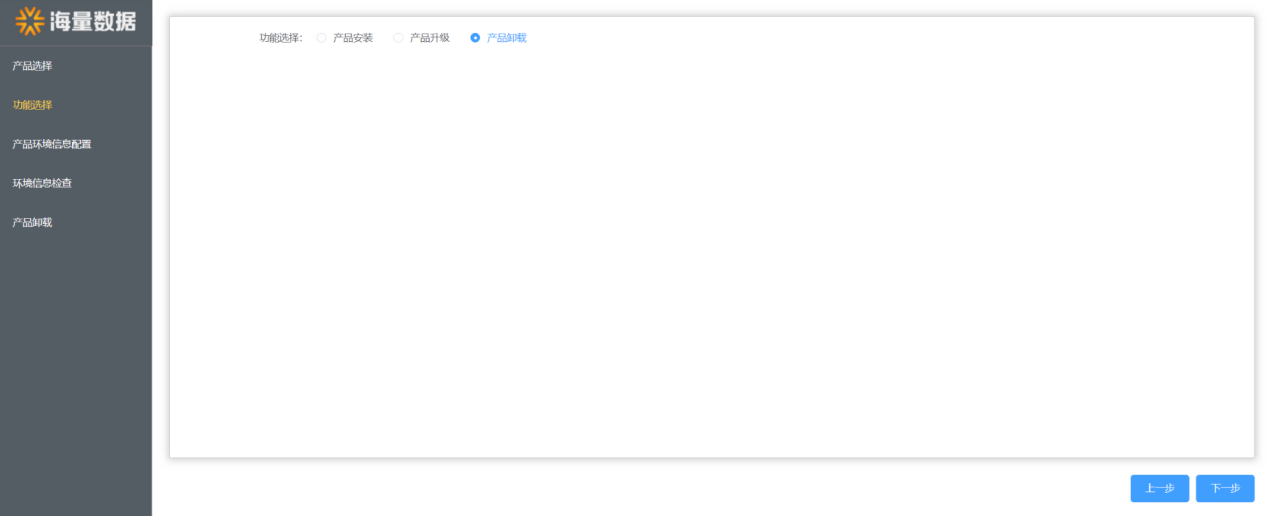


#### 卸载数据库

**步骤1** 左侧产品选择页，选择Vastbase G100（对应您需要卸载的数据库），单击下一步。

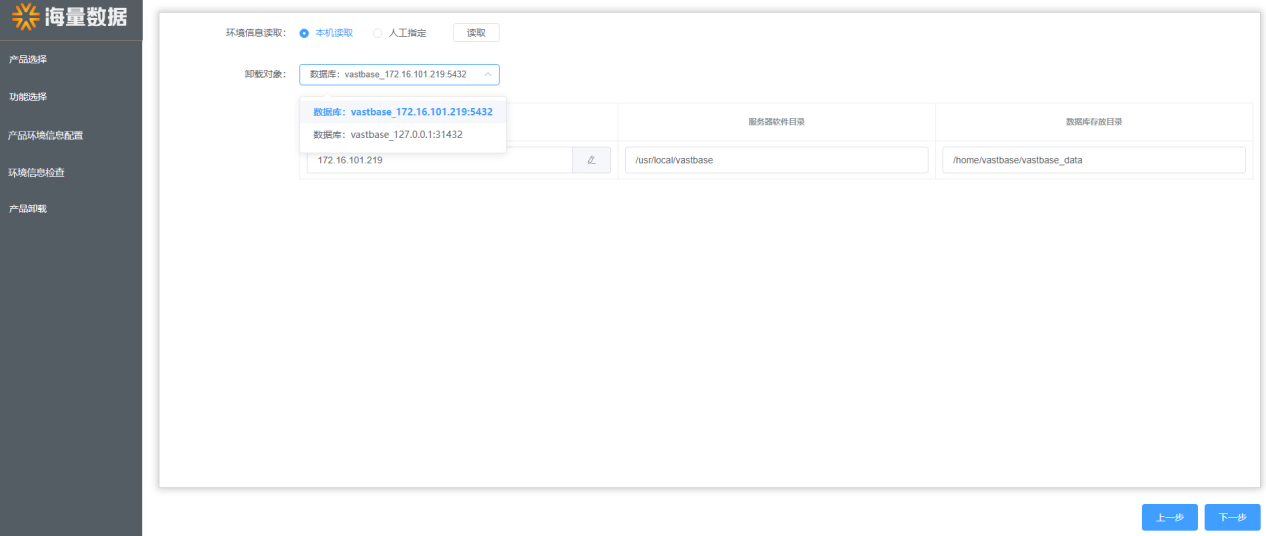


**步骤2** 左侧功能选择页，选择产品卸载，单击下一步。

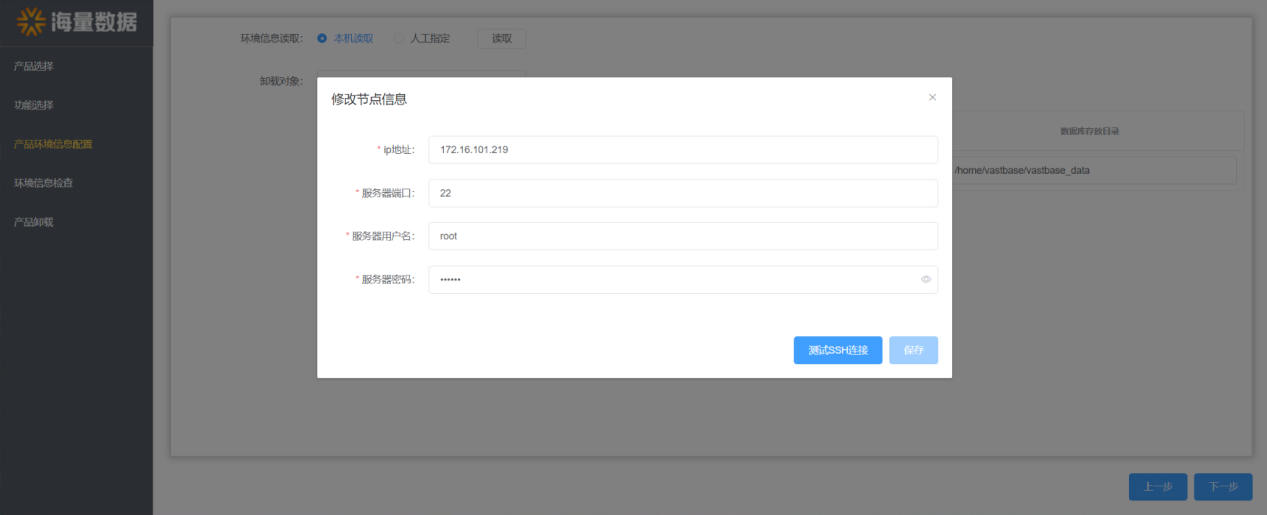


**步骤3** 左侧产品环境信息配置页，分为本机读取和人工指定。

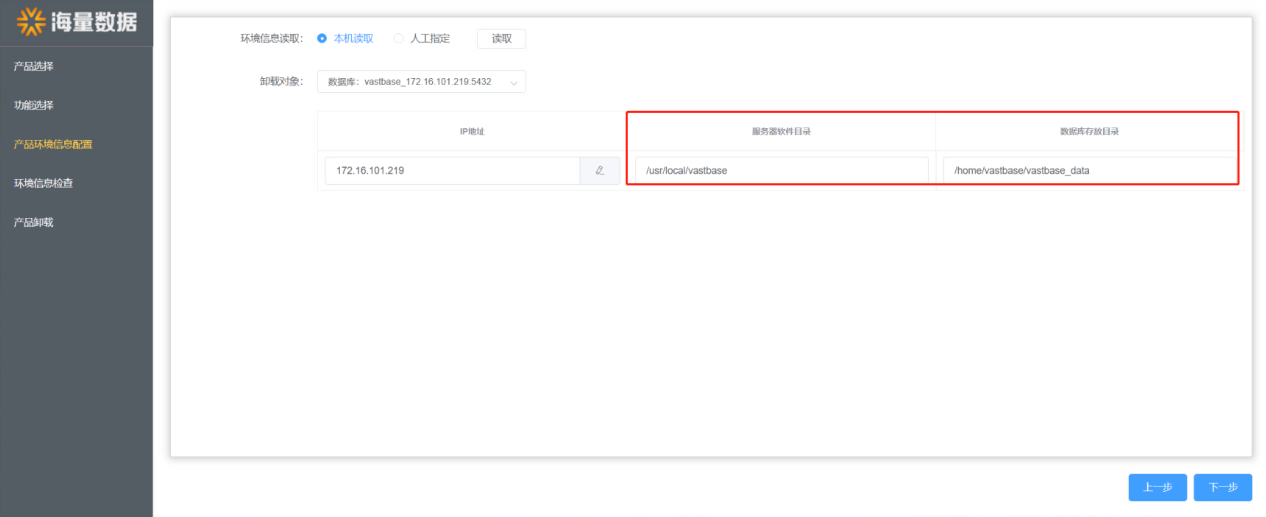
* **本机读取**
* **卸载对象：**单击读取，通过下拉框，展示当前安装程序所安装的数据库或集群信息列表。如下图：



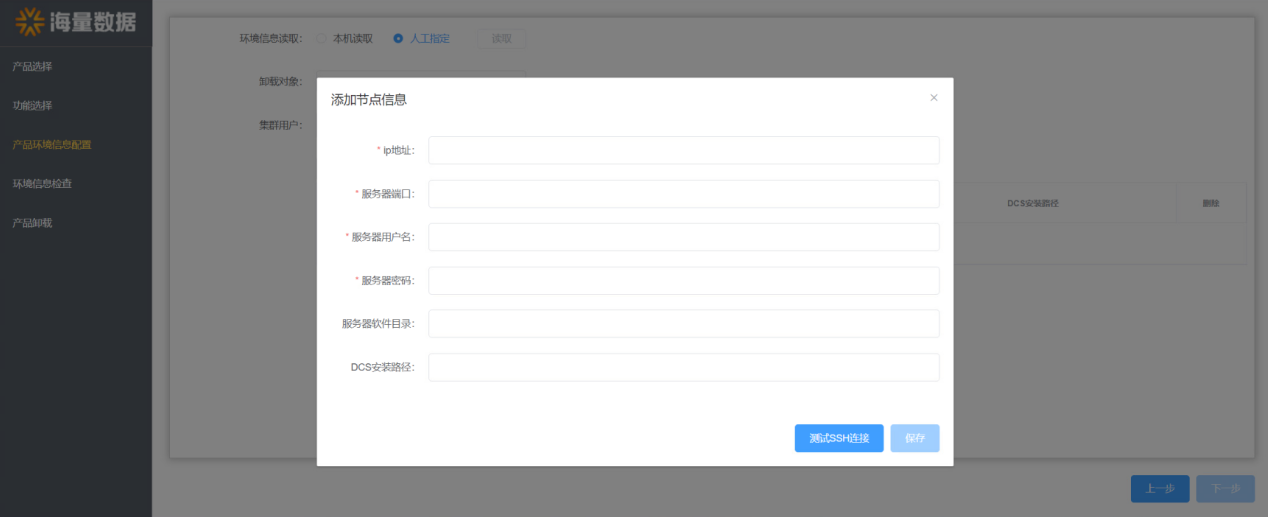
* **IP地址：**单击C:\Users\HP\Desktop\9.png9对本机读取的ip地址和服务器端口等信息进行修改，不允许为空。如下图：



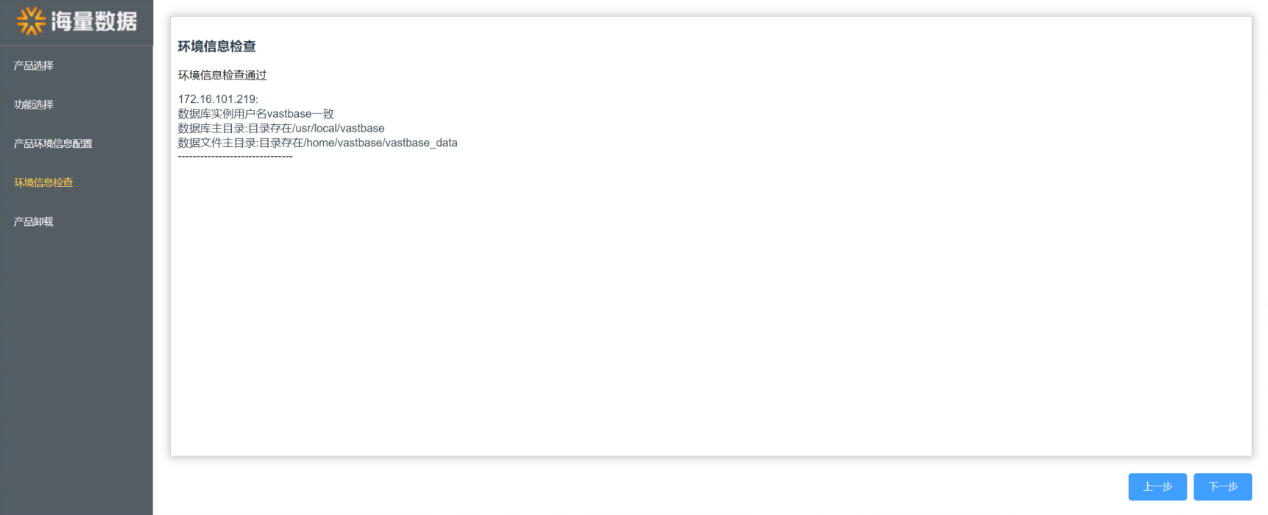
* 单击服务器软件目录和数据库存放目录，对读取的结果进行修改，不允许为空。如下图：



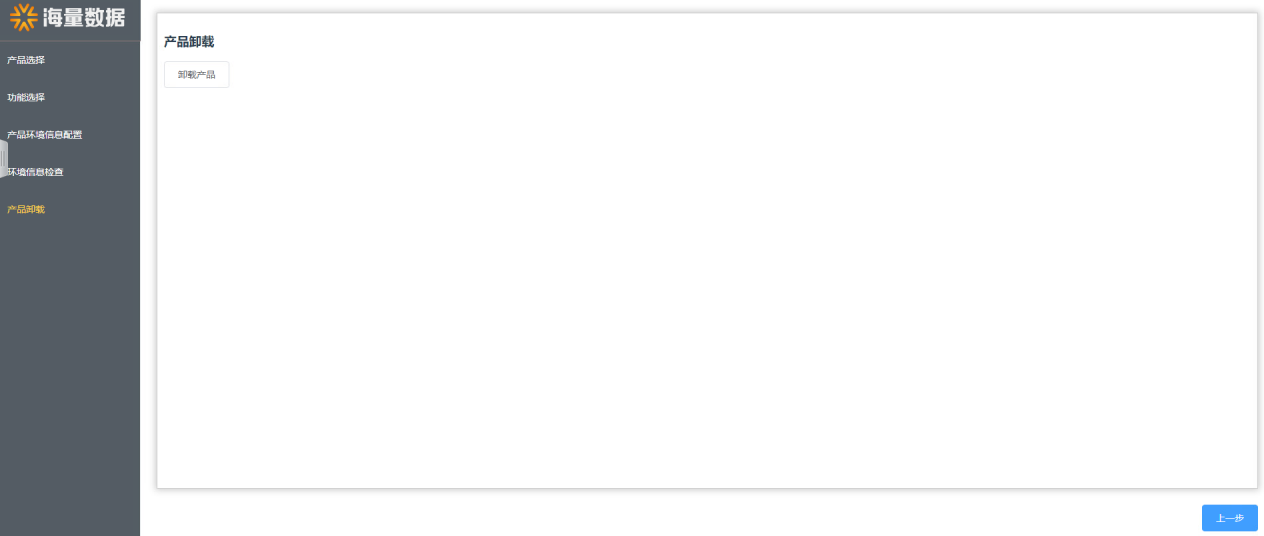
* **人工指定。**
* 添加节点信息：通过下拉菜单选择卸载对象，单击添加，输入相关信息（左侧带“\*”为必填信息），单击测试SSH连接，通过后单击保存。如下图：



**步骤4** 在环境信息检查页，检查通过后，单击下一步。

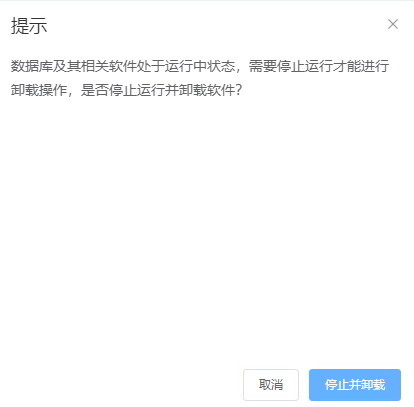


**步骤5** 在产品卸载页，单击卸载产品，完成卸载。



注：

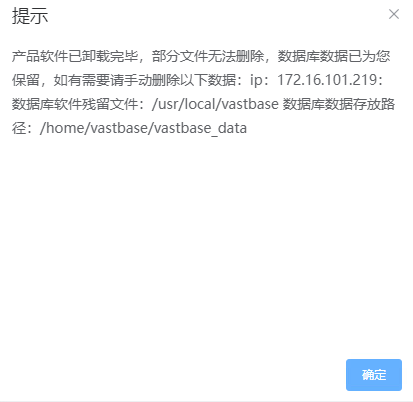
* 如果此时数据库处于运行状态，会弹窗提示确认框，单击停止并卸载，完成卸载。



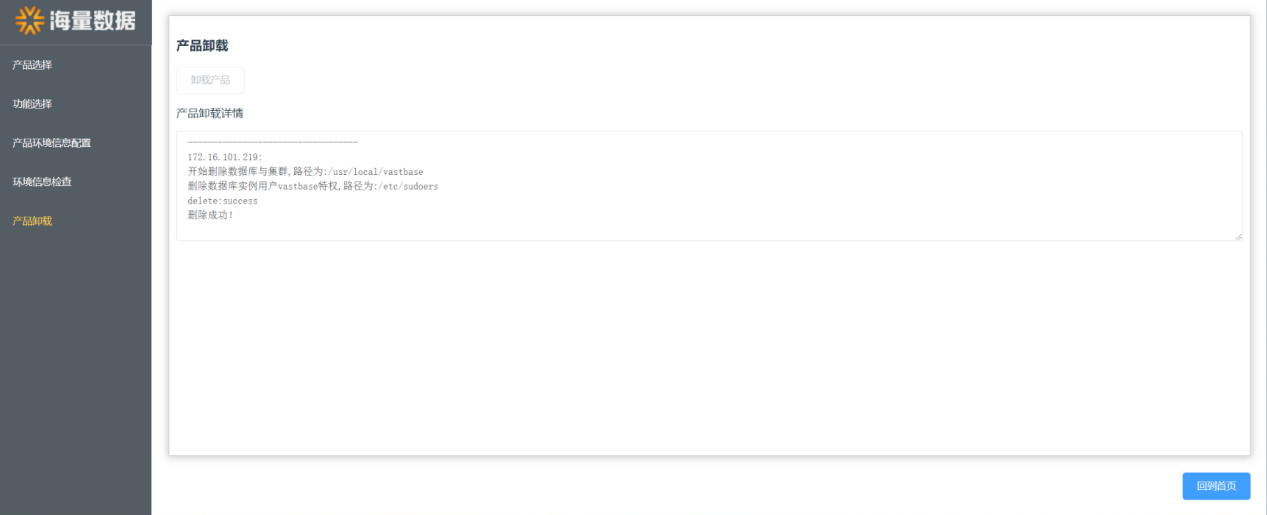
* 如果此时数据库并未运行，会弹窗卸载确认框，单击确认，完成卸载。



**步骤6** 卸载完成后，会弹框提示信息，查看后单击确认。



**步骤7** 完成卸载，单击回到首页，可继续其他操作。



### 镜像安装

镜像包含了Vastbase运行需要的基本环境、已经完成安装和初始化过的数据库，您可以快速使用数据库。

**前置条件**

需要提前安装好Docker，操作步骤如下：

**步骤1** 以root用户登录操作系统。

**步骤2** 执行如下命令安装Docker。

yum install docker

步骤3 启动Docker。

systemctl start docker

**导入（拉取）镜像**

**方式一：导入镜像（用户无外网访问权限）。**

**步骤1** 以root用户登录操作系统。

**步骤2** 进入镜像目录，导入镜像（镜像从工程师处获取）。G100\_Ver2\_2\_x86.tar为镜像名称。

docker load -i G100\_Ver2\_2\_x86.tar

**方式二：拉取镜像（用户有外网访问权限）。**

docker pull 17313185255/kunpeng-kylin-v10:G100\_Ver2\_2

**安装数据库**

**步骤1** 以root用户登录操作系统。

**步骤2** 创建数据目录，并授予权限。

mkdir -p /data/vb\_data

chmod 777 /data/vb\_data

**步骤3** 查看镜像，获取IMAGE ID。

docker images

显示结果如下：

**C:\Users\HP\Desktop\19.png19**

**步骤4** 初始化容器。

docker run -it --privileged=true --dns=8.8.8.8 -v /data/vb\_data:/data -p 192.168.1.205:5434:5432 d18a7ab34f71

注:上面命令中根据实际需要修改的部分：

|  |  |
| --- | --- |
| **序号** | **可修改部分** |
| 1 | 宿主机ip地址及端口 192.168.1.205:5432，5432为宿主机的端口，容器外用户可通过宿主IP和该端口访问数据库。 |
| 2 | DNS对应的ip |
| 3 | 宿主机固化的目录。即形如/data/vb\_data:/data 中的/data/vb\_data。其中/data是容器中的目录不能修改。 |
| 4 | 18a7ab34f71为查询到的镜像的IMAGE ID。 |

docker run ：创建一个新的容器并运行一个命令，语法如下：

docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

OPTIONS说明：

* -a stdin：指定标准输入输出内容类型，可选 STDIN/STDOUT/STDERR三项。
* -d：后台运行容器，并返回容器ID。
* -i：以交互模式运行容器，通常与 -t 同时使用。
* -P：随机端口映射，容器内部端口随机映射到主机的端口。
* -p：指定端口映射，格式为：主机(宿主)端口:容器端口。
* -t：为容器重新分配一个伪输入终端，通常与 -i 同时使用。
* --name="vastbase"：为容器指定一个名称。
* --dns 8.8.8.8：指定容器使用的DNS服务器，默认和宿主一致。
* --dns-search example.com:指定容器DNS搜索域名，默认和宿主一致。
* -h "mars"：指定容器的hostname。
* -e username="ritchie"：设置环境变量。
* --env-file=[]：从指定文件读入环境变量。
* --cpuset="0-2" or --cpuset="0,1,2"：绑定容器到指定CPU运行。
* -m :设置容器使用内存最大值。
* --net="bridge"：指定容器的网络连接类型，支持bridge、host、none和container四种类型。
* --link=[]：添加链接到另一个容器。
* --expose=[]：开放一个端口或一组端口。
* --volume，-v：绑定一个卷。

**步骤5** 宿主机上登录数据库。

su – vastbase

vsql -r -h 192.168.1.205 -U testuser -W Happy1074 -p 5432 postgres

注：可以连接到宿主机IP的远程客户端均可访问该数据库（容器会自动生成初始密码，可联系现场工程师获取）。

**注意事项**

通过vb\_ctl start/stop启动或停止数据库，但是需要指定PGDATA变量。我们提供了/home/vastbase/.Vastbase文件用于配置vastbase相关的环境变量，并在~/.bashrc文件中source，建议将相关环境变量写入.Vastbase文件。

### 安装卸载FAQ

**FAQ1：**如果安装失败，且无日志输出，则请执行非实化例数据库安装部署，在后续实例化过程中，若出现缺少libreadline.so.8或libreadline.so.7错误。

**解决方案：**

**步骤1** 切换到root用户。

su - root

**步骤2** 执行如下操作。

find / -name libreadline.so.8

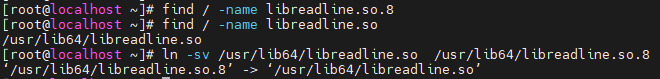
**步骤3** 如步骤2无回显，则查找是否有libreadline.so。

find / -name libreadline.so

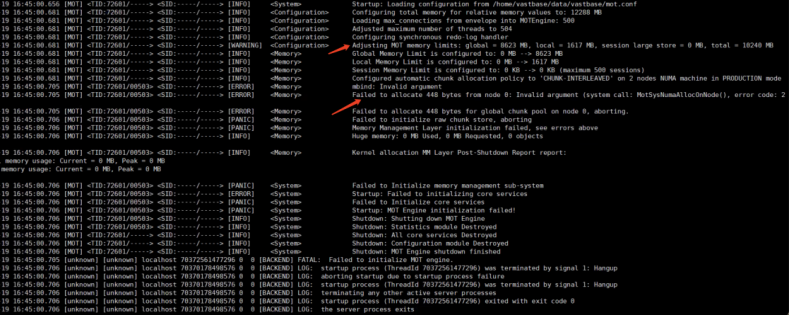
**步骤4** 执行如下软连接。

ln -sv /usr/lib64/libreadline.so /usr/lib64/libreadline.so.8

显示如下：



**FAQ2：**如果启动过程中提示启动失败，请查看$PGDATA/pg\_log中的日志。



以上提示由于CPU不支持MOT所致，MOT需要CPU特殊指令集的支持：

* X86\_64架构上，MOT依赖rdtscp指令。
* ARM架构上，MOT依赖isb、mrs、cntvct\_el0等指令。

**解决方案：**使用lscpu命令输出的Flags项提前判断CPU是否支持MOT，如果不支持请安装NOMOT版本（工程师提供安装包）。

# 升级说明

## 使用VPatch进行版本控制

操作前请做好原始产品和原始用户数据的备份！

**功能描述**

VPatch是用于Vastbase版本控制的工具可以实现在单机环境下的升级和回退操作，具体功能如下：

* 升级环境检查，补丁冲突检查等。
* PSU、PSR补丁的升级、回退及升级或回退失败时的还原。
* 升级过程中记录相关日志，步骤文件。
* 升级过程中管理备份文件。

**前提条件**

需在python3.7环境下运行，即执行python3 --version，返回版本为3.7.\* 。

**语法格式**

* 配置文件如下(ini格式)：

[vpatch]

userName=用户名user1

userGroup=用户组group1

appPath=产品软件目录

dataPath=集簇的绝对路径

port=端口

tmpPath=升级过程临时文件目录

* 执行升级。

vpatch --upgrade

* 执行回退，即回退掉升级记录中对应操作记录号(patch\_record\_id)所升级的补丁，操作记录号被记录在升级历史信息中。

vpatch --rollback=patch\_record\_id

* 执行连续回退。倒叙遍历升级历史信息，逐个回退升级补丁，直到回退了指定操作记录号的补丁为止。注意此处指定的升级操作记录号(patch\_record\_id)也在回退范围内。

vpatch --rollbackTo=patch\_record\_id

* 查看vpatch版本信息。

vpatch -V | --version

* 查看vpatch帮助信息。

vpatch -? | --help

* -I|--information：显示vpatch同目录下升级补丁的信息，在rollback时显示回退的升级补丁的信息，在rollback to时显示最后一个被回退的升级补丁的信息。

vpatch -I | --information

* -U|--upgradeInformation：显示升级历史信息，若补丁已被回退，则相关信息不在此处显示。此信息在用户需要指定回退时尤为重要。需要回退时可以参考升级历史信息中提供的升级操作号(patch\_record\_id)，及升级补丁的相关信息。

vpatch -U [-s] | --upgradeInformation [--simple]

完整显示升级历史信息。

vpatch -U [-s] | --upgradeInformation --complete

* 为vpatch工具设定参数(root用户执行)。

1、命令行直接指定参数。

vpatch --setparam -u 用户名 -g 用户组 -a 产品目录 -d 集簇目录 -p 端口号 -t 临时目录

2、给定ini配置文件。

vpatch --setparam -f ini配置文件绝对路径

* 对升级或回退补丁做基本检查(root用户执行)。

vpatch -C | --check [ --rollbackId=patch\_record\_id | --rollbacktoId=patch\_record\_id]

* 为升级或回退做准备工作(root用户执行)。

vpatch -P | --prepare [ --rollbackId=patch\_record\_id]|--rollbackToId=patch\_record\_id]

* 连续回退。

vpatch --rollbackTo=patch\_record\_id

**注意事项**

* 对于集群升级，需还原为单机模式进行升级，升级后重新部署集群。
* 升级时指定的端口号应避开日常使用的服务端口号，防止升级过程被外部干扰。
* 在升级或回退前应关闭数据库服务。
* 不支持修改用户数据(vb\_initdb初始化后并未包含的由用户操作产生的表) 升级的场景。
* 不支持修改过数据字典的场景进行rollback 和rollback to回退操作。
* 仅支持未修改系统表且允许回退的补丁进行rollback 和rollback to回退操作。
* 不支持多实例场景，多实例即为单一产品对应多个数据库集簇的场景。
* 升级过程中，check和prepare阶段会向数据库初始化用户的环境配置文件中写入环境变量，所以切换用户时若使用了不同的终端，需要执行以下命令避免出现环境变量缺失的问题。

source ~/.bashrc

* 仅支持Vastbase G100 V2.2(build 10)及以上版本的升级。

**参数说明**

**表1**补丁信息

| **参数** | **说明** |
| --- | --- |
| vbVpatchInfo | 供vpatch使用的信息列表。 |
| patchNum | 补丁编号。 |
| patchFixedBugList | 修补bug列表。 |
| patchType | 补丁类型。 |
| patchReleaseDate | 补丁发布日期。 |
| compatibleProductVersionNum | 保留参数。 |
| patchConflictList | 保留参数。 |
| patchIncludeList | 保留参数。 |
| trunkKernelVersion | 保留参数。 |
| isUpdateCatalog | 是否修改系统表。 |
| isUpdateOnlyFunc | 修改系统表，但仅限于函数。 |
| allowRollback | 补丁是否允许回退，true可以回退，false不可回退。 |
| base\_product\_version | 适配的基础版本。 |
| baseSvn | 原始版本的SVN提交号。 |
| newSvn | 新版本的SVN提交号。 |
| isFallbackPatch | 用于升级还是回退的补丁，true升级，false回退。 |
| osInfo | 适用的操作系统信息。 |
| cpuInfo | 适用的cpu信息。 |
| deleteFile | 此次升级删除的文件，包括数据库和om工具。 |

**表2**历史信息参数说明

|  |  |
| --- | --- |
| **参数** | **说明** |
| auditId | 审计号。 |
| patch\_record\_id | 操作序列号。 |
| patchNum | 补丁编号。 |
| date | 升级操作的日期和时间。 |
| opType | 操作类型（One off patch，升级PSU,升级PSR）。 |
| commitId | svn提交号。 |
| fixedBugs | 修复的bug列表。 |
| isUpdateCatalog | 是否修改系统表。 |
| isUpdateOnlyFunc | 是否修改了函数。 |
| isRollback | 是否已被回退。 |
| upgradeBackupPath | 升级备份的目录。 |
| patchConflictList | 保留参数。 |
| patchInlcudeList | 保留参数。 |
| vpatchVersion | vpatch版本。 |
| vpatchTmpDir | vpatch临时文件目录。 |
| vpatchLogDir | vpatch日志目录 |
| newSVN | 新版本的SVN提交号。 |
| baseSVN | 原始版本的SVN提交号。 |
| deleteFile | 此次升级删除的文件，包括数据库和om工具。 |
| isFallbackPatch | 用于升级还是回退的补丁，true升级，false回退。 |
| allowRollback | 补丁是否允许回退，true可以回退，false不可回退。 |
| base\_product\_version | 适配的基础版本。 |
| osInfo | 适用的操作系统信息。 |
| cpuInfo | 适用的cpu信息。 |
| ConsumedOpSeqNum | 数组的大小。 |

**示例**

**示例1：**升级操作。

1、解压升级包并进入目录。

cd /home/数据库用户/解压的升级包/VPatch/script

2、指定配置文件。

vpatch --setparam -f /home/jq\_vpatch\_two/xxx.ini

配置文件参考如下：

[vpatch]

userName=jq\_vpatch\_two

userGroup=jq\_vpatch\_two

appPath=/home/jq\_vpatch\_two/local/vastbase

dataPath=/home/jq\_vpatch\_two/data/vastbase

Port=8560

tmpPath=/home/jq\_vpatch\_two/local/omTmp/tmp

3、root执行命令进行检查。

./vpatch -C

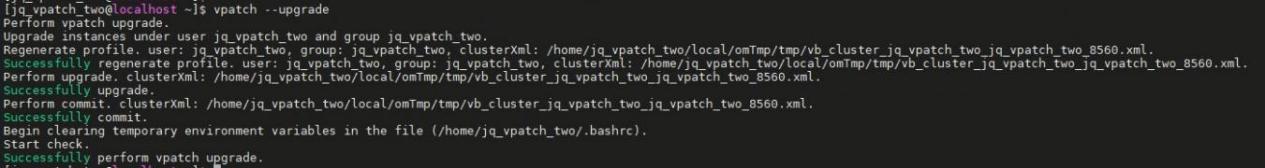
4、root执行命令进行用户权限检查。

./vpatch -P

5、切换数据库初始化用户进行升级。

vpatch --upgrade

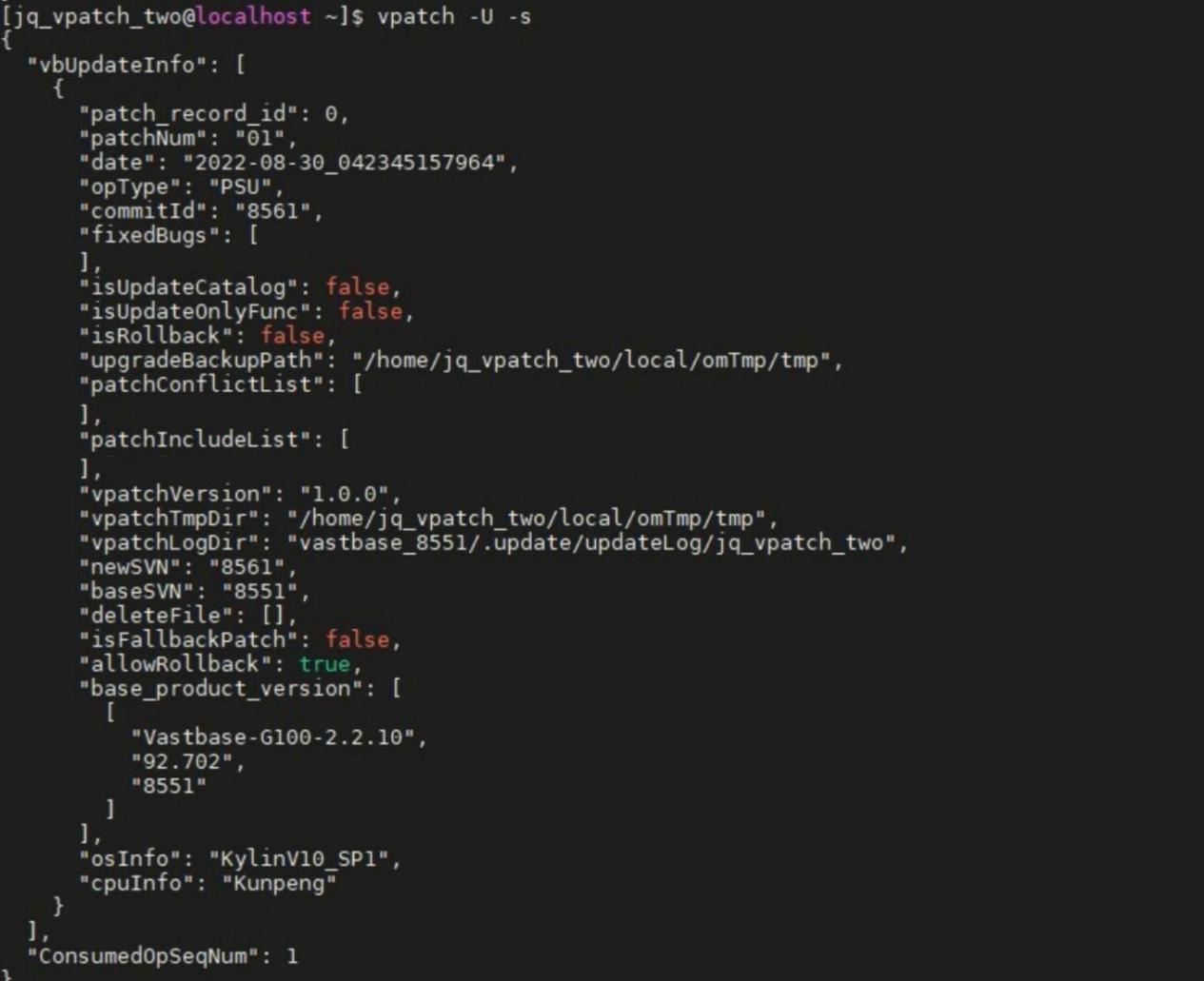
结果返回如下：

6、升级完成后信息检查。

* 简略显示升级历史信息，信息包括操作序列号、补丁编号、升级操作的日期和时间、操作类型、修复的bug列表、升级是否修改数据字典、是否已被回退、升级备份的目录。参数说明参考表2历史信息。

vpatch -U 或 vpatch -U -s

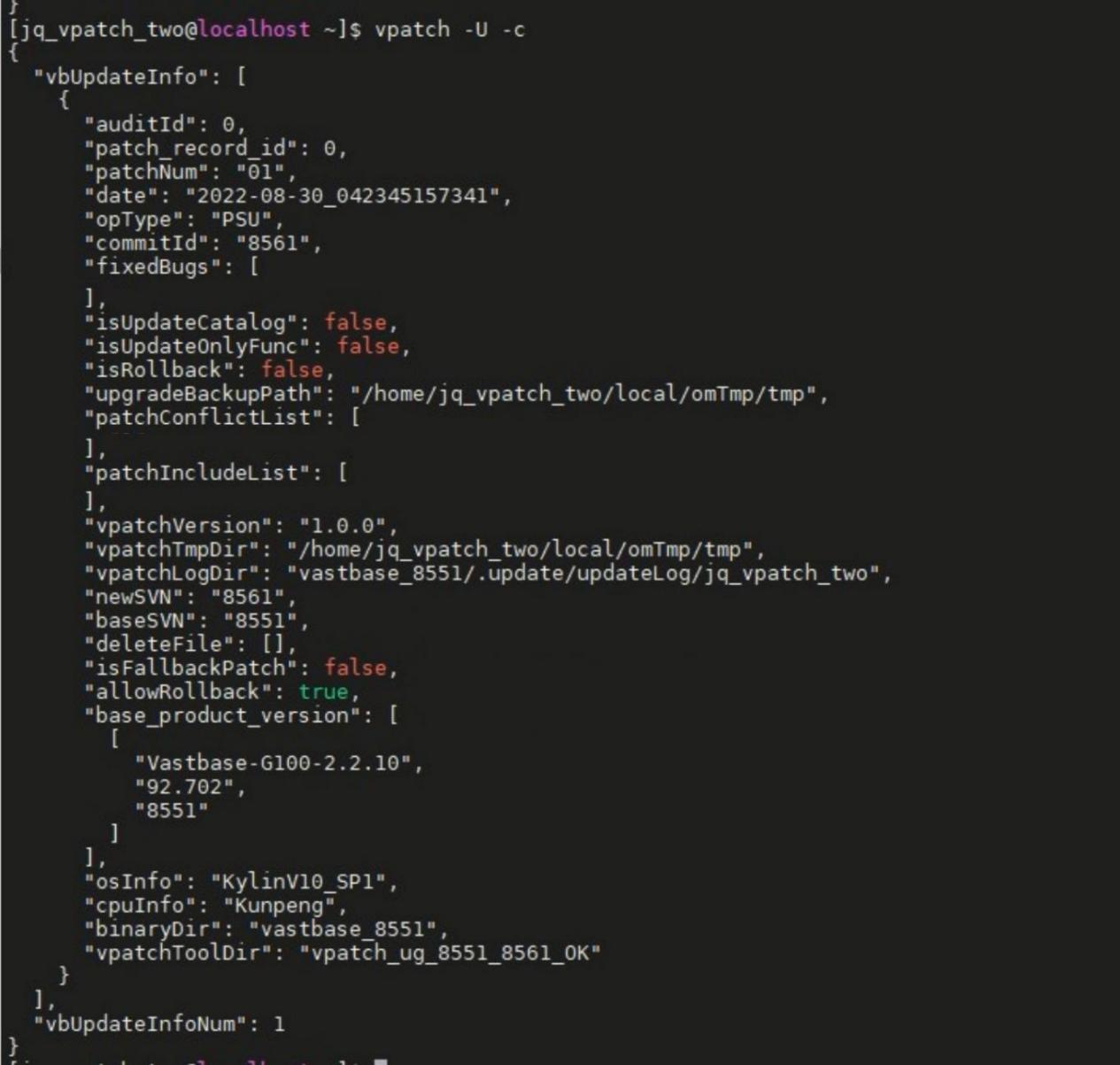
结果返回如下：



* 完整信息升级记录信息，信息包括操作序列号、补丁编号、升级操作的日期和时间、操作类型、修复的bug列表、升级是否修改数据字典、是否已被回退、升级备份的目录。参数说明参考表2历史信息。

vpatch -U -c

结果返回如下：



* 显示升级过程，信息包含停止服务，启动服务操作，执行的每个脚本文件，操作数据库的相关命令，变更的设置参数，查看数据库升级日志。

cat $GAUSSHOME/.update/updatelog/jq\_vpatch\_two/gs\_upgradectlxxx.log

**示例2：**回退操作。

1. 停止数据库服务。
2. root用户指定配置文件。

vpatch --setparam -f /home/jq\_vpatch\_two/xxx.ini

3、root用户进行检查。

./vpatch -C --rollbackId=0

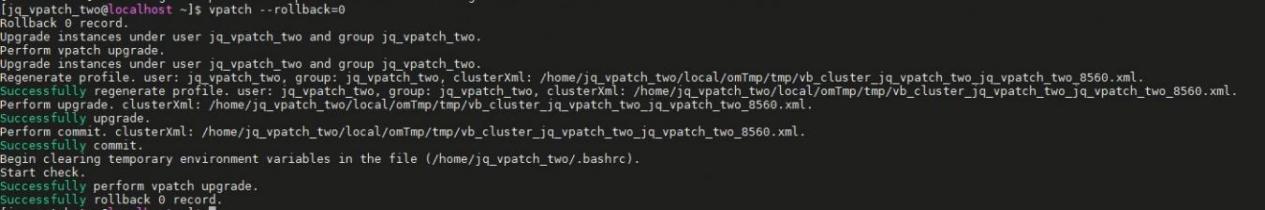
4、root执行命令进行用户权限检查。

./vpatch -P --rollbackId=0

5、切换数据库初始化用户执行回退。

vpatch --rollback=0

结果返回如下：



**示例3：**连续回退操作。

1、停止数据库服务。

2、root用户指定配置文件。

vpatch --setparam -f /home/jq\_vpatch\_two/xxx.ini

3、root用户进行检查。

./vpatch -C --rollbackToId=0

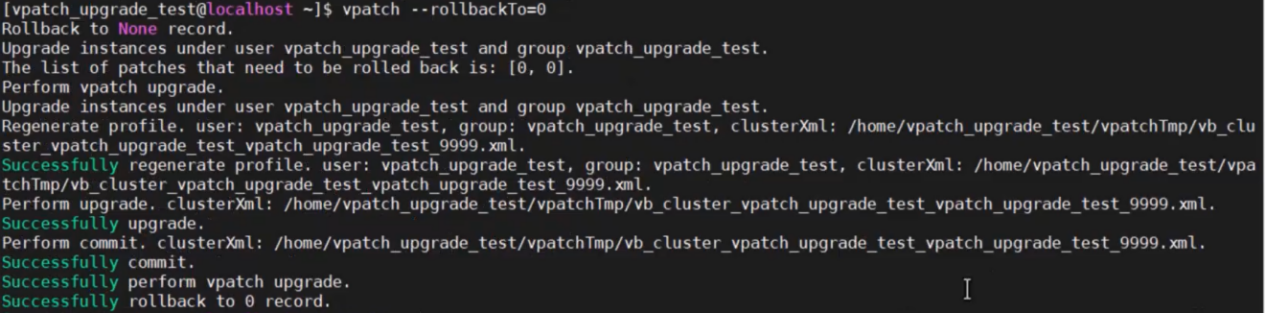
4、root执行命令进行用户权限检查。

./vpatch -P --rollbackToId=0

5、切换数据库初始化用户执行连续回退。

vpatch --rollbackTo=0

结果返回如下：



**示例4：**显示vpatch同目录下升级补丁的信息，参数说明参考表1补丁信息。

./vpatch - I

结果返回如下：



**示例5：**集群升级操作。

1. 使用root用户登录主机，执行以下命令查看主备状态，/home/vastbase/has/bin/hasctl为hasctl工具目录。

/home/vastbase/has/bin/hasctl -c /home/vastbase/has/etc/patroni\_conf.yml list

1. root用户登录备节点执行以下命令停止has

systemctl stop has

1. root用户登录主节点，执行步骤2命令停止主节点has服务。
2. 复制postgresql.conf做配置文件备份

cp postgresql.conf postgresql.conf.bak

5、注释掉主备节点配置文件postgresql.conf中的replconn参数，并修改port指定升级专用端口号，重启数据库。确认数据库可以正常启动后，停止所有节点的数据库服务进行升级操作。

6、升级参考示例1分别对主备节点进行操作。

7、升级成功后，关闭数据库，恢复postgresql.conf文件主备及端口配置。可将步骤4中备份的postgresql.conf.bak文件恢复为postgresql.conf。

8、以root用户登录主节点，并执行以下命令先启动主节点has：

systemctl start has

9、使用以下命令查看主节点状态，确认主节点状态正常。

/home/vastbase/has/bin/hasctl  -c /home/vastbase/has/etc/patroni\_conf.yml list

10、以root用户登录备节点，并执行以下命令启动备节点has：

systemctl start has

11、所有节点启动后执行以下命令查看集群状态是否正常。集群正常则升级成功。

/home/vastbase/has/bin/hasctl  -c /home/vastbase/has/etc/patroni\_conf.yml list

# 启停数据库

vb\_ctl是Vastbase提供的数据库服务控制工具，可以用来启停数据库服务和查询数据库状态，主要供Vastbase管理模块调用。

**命令格式**

* 启动数据库

vb\_ctl start [-w] [-t seconds] [-s] [-D datadir] [-l filename][-o options] [-p path] [-c]

* 停止数据库

vb\_ctl stop [-W] [-t seconds] [-s] [-D datadir] [-m s[mart]| f[ast] | i[mmediate] ]

* 重启数据库

vb\_ctl restart [-w] [-t seconds] [-s] [-D datadir] [-c] [-ms[mart] | f[ast] | i[mmediate] ] [-o options]

* 显示数据库运行状态

vb\_ctl status [-D datadir]

**参数说明**

* 公共参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| -D, --pgdata=DATADIR | 指定数据目录的位置。 | DATADIR的取值必须为有效的数据目录。 |
| -s, --silent | 只打印错误，不打印提示信息。 | - |
| -t, --timeout=SECS | 等待数据库启动、关闭或者主备切换完成的最大秒数。如果等待超时，命令会结束退出，并通知不再等待。 | 取值范围：整型  取值范围：整型（秒）。默认值：等待数据库启动、停止或者主备切换完成：60秒。 |
| -w | 启动或者关闭数据库时，会等待事务操作完成后再执行。 | 默认值：省略此参数时，默认会等待事务操作完成。 |
| -W | 启动或者关闭数据库时，不会等待事务操作完成，强制执行启动或者关闭操作。 | - |

* start或restart共有参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| -c, --core-file | 允许Vastbase进程产生内核文件。 | - |
| -l, --log=FILENAME | 把服务器日志输出附加在FILENAME文件上。 | FILENAME的取值为启动数据库服务的用户有权限的文件。  例如：data目录下的某文件。 |
| -o OPTIONS | 声明要直接传递给由Vastbase执行的Vastbase的命令行选项。  参数通常都用单或者双引号包围以保证它们作为一个整体传递。 | - |
| -p path | 声明Vastbase可执行文件的位置。 | 缺省位于vb\_ctl自身所在目录，一般不需要此选项。 |

* stop和restart共有参数

| **参数** | **参数说明** | **取值范围** |
| --- | --- | --- |
| -m, --mode=MODE | 声明关闭模式。 | * mode的取值： * smart：单机模式下主机立即断开，同fast。 * fast：不等待客户端中断连接，所有活跃事务都被回滚并且客户端都被强制断开，然后服务器将被关闭。 * immediate：强行关闭，在下次重新启动的时候将导致故障恢复。   默认值：fast |

**示例：**

切换至安装过程中创建的操作系统用户，通过以下命令启动、停止、重启和显示数据库状态。

* 启动数据库

vb\_ctl start

* 停止数据库

vb\_ctl stop

* 重启数据库

vb\_ctl restart

* 显示数据库状态

vb\_ctl status

# 使用数据库

成功安装Vastbase后，就可以开始使用Vastbase进行数据库构建和管理了，Vastbase自带客户端连接工具，用户可以通过vsql连接数据库，并对数据库进行管理。

## 使用vsql连接数据库

vsql是Vastbase提供的在命令行下运行的数据库连接工具。此工具除了具备操作数据库的基本功能，还提供了若干高级特性，便于用户使用。本节主要介绍如何使用vsql连接数据库。

**前提条件**

* 已确认连接信息，包括连接的数据库名称与数据库端口。
* 已确认连接信息，包括连接的IP、数据库名称与数据库端口。

**本地连接数据库**

**步骤1** 执行如下命令以操作系统用户vastbase登录数据库主节点，连接成功后即可执行数据库命令对数据库进行管理。

vsql -d vastbase -p 5432

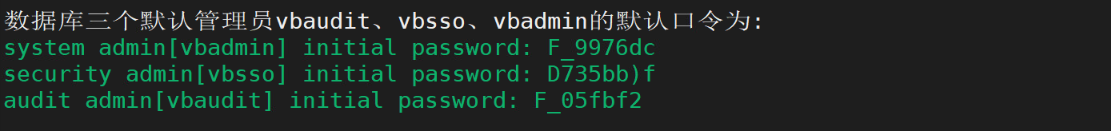
* vastbase：需要连接的数据库名称，安装完成后，默认生成名称为vastbase的数据库名称，实际登录时请根据具体情况替换。
* 5432：数据库主节点的端口号，实际登录时请根据具体情况替换。
* 使用vsql链接数据库时需采用-d和-p指定要连接的数据库及端口号。

**步骤2** （管理员首次登录时）如果您的数据库版本是2.2（Build 5.75）及以上且角色是系统管理员（vbadmin）、安全管理员（vbsso）或者审计管理员（vbaudit），首次登录时系统会强制要求修改口令，修改语法如下：

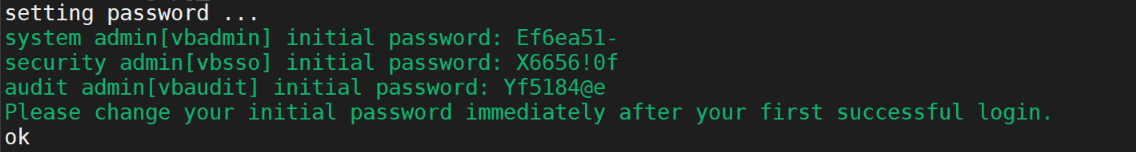
ALTER ROLE 用户名 IDENTIFIED BY '新口令' REPLACE '原随机口令';

原随机口令：

* 实例化数据库安装：安装过程中会生成随机口令作为管理员初始口令，显示信息如下：



* 非实例化数据库安装：vb\_initdb初始化数据库时生成的管理员随机口令，显示信息如下：



**步骤3** （普通用户首次登录时）如果您的数据库版本是2.2（Build 5.75）及以上且属于管理员新创建用户或者修改过口令后的普通用户，首次登录数据库，需要自行修改口令后才可以正常操作数据库，修改语法如下：

ALTER ROLE 用户名 IDENTIFIED BY '新口令' REPLACE '原口令';

**步骤4** 退出数据库。

\q

**使用vsql远程连接**

**步骤1** 以操作系统用户vastbase登录数据库主节点。

vsql -d vastbase -p 5432

**步骤2** 配置客户端认证方式，在数据目录中pg\_hba.conf文件添加相应的认证策略。

local DATABASE USER METHOD [OPTIONS]

host DATABASE USER ADDRESS METHOD [OPTIONS]

hostssl DATABASE USER ADDRESS METHOD [OPTIONS]

hostnossl DATABASE USER ADDRESS METHOD [OPTIONS]

注："pg\_hba.conf"文件中的每条记录可以是以上四种格式之一，因为认证时系统是为每个连接请求顺序检查"pg\_hba.conf"里的记录的，所以这些记录的顺序是非常关键。

**步骤3** 执行如下命令连接数据库。

vsql -h <IP地址> -p <端口号> -d <数据库名> -U <用户名>

示例：

vsql -h 192.168.1.7 -p 5432 -d postgres -U user1

## 数据库管理

### 创建数据库

数据库安装完成后，默认生成名称为vastbase的数据库。您需要自己创建一个新的数据库。默认情况下新数据库将通过复制标准系统数据库template0来创建，且仅支持使用template0来创建。

**注意事项**

* 只有拥有CREATEDB权限的用户才可以创建新数据库，系统管理员默认拥有此权限。
* 不能在事务块中执行创建数据库语句。
* 在创建数据库过程中，若出现类似"could not initializedataba se directory"的错误提示，可能是由于文件系统上数据目录的权限不足或磁盘满等原因引起。

**语法格式**

* 创建数据库

CREATE DATABASE database\_name

[ [ WITH ] { [ OWNER [=] user\_name ] |

[ TEMPLATE [=] template ] |

[ ENCODING [=] encoding ] |

[ LC\_COLLATE [=] lc\_collate ] |

[ LC\_CTYPE [=] lc\_ctype ] |

[ DBCOMPATIBILITY [=] compatibilty\_type ] |

[ TABLESPACE [=] tablespace\_name ] |

[ CONNECTION LIMIT [=] connlimit ]}[...] ];

* 查看数据库（vsql客户端可用）

\l

* 修改数据库

ALTER DATABASE database\_name RENAME TO new\_name;

* 删除数据库

DROP DATABASE database\_name;

**参数说明**

* database\_name：要创建、修改或者删除的数据库名称。
* new\_name：数据库的新名称。

**示例**

* 创建一个新的数据库testdb1。

CREATE DATABASE testdb1;

* 用模板template0创建数据库testdb2，并指定所有者为dbuser。

CREATE DATABASE testdb2 OWNER dbuser TEMPLATE template0;

* 查看数据库（vsql客户端可用）。

\l

* 修改将testdb1数据库名称修改为testdb3。

ALTER DATABASE testdb1 RENAME TO testdb3;

* 删除数据库testdb3。

DROP DATABASE testdb3;

### 创建表

表是建立在数据库中的，在不同的数据库中可以存放相同的表。甚至可以通过使用模式在同一个数据库中创建相同名称的表。

**语法格式**

* 创建表。

CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ]

TABLE [ IF NOT EXISTS ]table\_name

({ column\_name data\_type [ compress\_mode ] [ COLLATE collation ]

[column\_constraint[ ... ] ]

| table\_constraint

| LIKE source\_table [ like\_option [...] ] }

[, ... ])

[ WITH ( {storage\_parameter = value} [, ... ] ) ]

[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]

[ COMPRESS | NOCOMPRESS ]

[ TABLESPACE tablespace\_name ];

* 其中列约束column\_constraint为：

[ CONSTRAINT constraint\_name ]

{ NOT NULL |

NULL |

CHECK ( expression ) |

DEFAULT default\_expr |

UNIQUE index\_parameters |

PRIMARY KEY index\_parameters }

[ DEFERRABLE | NOT DEFERRABLE | INITIALLY DEFERRED | INITIALLY

IMMEDIATE ]

* 其中列的压缩可选项compress\_mode为：

{ DELTA | PREFIX | DICTIONARY | NUMSTR | NOCOMPRESS }

* 其中表约束table\_constraint为：

[ CONSTRAINT constraint\_name ]

{ CHECK ( expression ) |

UNIQUE ( column\_name [, ... ] ) index\_parameters |

PRIMARY KEY ( column\_name [, ... ] ) index\_parameters |

PARTIAL CLUSTER KEY ( column\_name [, ... ] ) }

[ DEFERRABLE | NOT DEFERRABLE | INITIALLY DEFERRED | INITIALLY

IMMEDIATE ]

* 其中like选项like\_option为：

{ INCLUDING | EXCLUDING } { DEFAULTS | CONSTRAINTS | INDEXES |

STORAGE | COMMENTS |PARTITION | RELOPTIONS | ALL }

* 其中索引参数index\_parameters为：

[ WITH ( {storage\_parameter = value} [, ... ] ) ]

[ USING INDEX TABLESPACE tablespace\_name ]

**示例**

* 创建简单的表。

CREATE TABLE public.warehouse\_t1

(

W\_WAREHOUSE\_SK INTEGER NOT NULL,

W\_WAREHOUSE\_ID CHAR(16) NOT NULL,

W\_WAREHOUSE\_NAME VARCHAR(20) ,

W\_WAREHOUSE\_SQ\_FT INTEGER ,

W\_COUNTRY VARCHAR(20) ,

W\_GMT\_OFFSET DECIMAL(5,2)

);

* 创建表，并指定W\_STATE字段的缺省值为GA。

CREATE TABLE public.warehouse\_t2

(

W\_WAREHOUSE\_SK INTEGER NOT NULL,

W\_WAREHOUSE\_ID CHAR(16) NOT NULL,

W\_COUNTY VARCHAR(30) ,

W\_STATE CHAR(2) DEFAULT 'GA',

W\_ZIP CHAR(10) ,

W\_COUNTRY VARCHAR(20) ,

W\_GMT\_OFFSET DECIMAL(5,2)

);

* 创建一个带有70%填充因子的表。

CREATE TABLE public.warehouse\_t3

(

W\_WAREHOUSE\_SK INTEGER NOT NULL,

W\_COUNTRY VARCHAR(20) ,

W\_GMT\_OFFSET DECIMAL(5,2)

) WITH(fillfactor=70);

* 创建表，并指定该表数据不写入预写日志。

CREATE UNLOGGED TABLE public.warehouse\_t4

(

W\_WAREHOUSE\_SK INTEGER NOT NULL,

W\_COUNTRY VARCHAR(20) ,

W\_GMT\_OFFSET DECIMAL(5,2)

);

* 创建表临时表。

CREATE TEMPORARY TABLE warehouse\_t5

(

W\_WAREHOUSE\_SK INTEGER NOT NULL,

W\_COUNTRY VARCHAR(20) ,

W\_GMT\_OFFSET DECIMAL(5,2)

);

* 创建表时，指定表空间PG\_DEFAULT。

CREATE TABLE public.warehouse\_t6

(

W\_ID INTEGER NOT NULL,

W\_NAME VARCHAR(20) ,

W\_MAIL VARCHAR(40)

) TABLESPACE PG\_DEFAULT;

* 创建一个有主键约束的表。

CREATE TABLE public.warehouse\_t7

(

W\_WAREHOUSE\_SK INTEGER PRIMARY KEY,

W\_WAREHOUSE\_ID CHAR(16) NOT NULL,

W\_COUNTRY VARCHAR(20) ,

W\_GMT\_OFFSET DECIMAL(5,2)

);

* 创建一个有复合主键约束的表。

CREATE TABLE public.warehouse\_t8

(

W\_WAREHOUSE\_SK INTEGER NOT NULL,

W\_WAREHOUSE\_ID CHAR(16) NOT NULL,

W\_WAREHOUSE\_NAME VARCHAR(20),

W\_COUNTRY VARCHAR(20),

W\_GMT\_OFFSET DECIMAL(5,2),

CONSTRAINT W\_CSTR\_KEY2 PRIMARY KEY(W\_WAREHOUSE\_SK, W\_WAREHOUSE\_ID)

);

* 定义一个检查列约束。

CREATE TABLE public.warehouse\_t9

(

W\_WAREHOUSE\_SK INTEGER CHECK (W\_WAREHOUSE\_SK > 0),

W\_WAREHOUSE\_ID CHAR(16) NOT NULL,

W\_WAREHOUSE\_NAME VARCHAR(20) CHECK (W\_WAREHOUSE\_NAME IS NOT NULL),

W\_WAREHOUSE\_SQ\_FT INTEGER ,

W\_GMT\_OFFSET DECIMAL(5,2)

);

* 向表中增加一个varchar列。

ALTER TABLE public.warehouse\_t1 ADD W\_GOODS\_CATEGORY varchar(30);

* 给表增加一个检查约束。

ALTER TABLE public.warehouse\_t2 ADD CONSTRAINT W\_CONSTR\_KEY4 CHECK (W\_WAREHOUSE\_SK >10);

* 给一个已存在字段添加非空约束。

ALTER TABLE public.warehouse\_t3 ALTER COLUMN W\_COUNTRY SET NOT NULL;

* 重命名已存在的表。

ALTER TABLE public.warehouse\_t4 RENAME TO newtab;

* 删除表。

DROP TABLE public.warehouse\_t1;

DROP TABLE public.warehouse\_t2;

DROP TABLE public.warehouse\_t3;

### 管理表

**向表中插入数据**

在创建一个表后，表中并没有数据，在使用这个表之前，需要向表中插入数据。本小节介绍如何使用Insert命令插入一行或多行数据，及从指定表插入数据。

**示例**

执行如下命令建立示例中需要使用的表customer\_t1。

CREATE TABLE customer\_t1

(

c\_customer\_sk integer,

c\_customer\_id char(5),

c\_first\_name char(6));

向表中插入数据前，意味着表已创建成功。创建表的步骤请参见创建表。

* 向表customer\_t1中插入一行：
* 数据值是按照这些字段在表中出现的顺序列出的，并且用逗号分隔。通常数据值是文本（常量），但也允许使用标量表达式。

INSERT INTO customer\_t1(c\_customer\_sk, c\_customer\_id, c\_first\_name)

VALUES (3769, 'hello', 'Grace');

* 如果用户已经知道表中字段的顺序，也可无需列出表中的字段。例如以下命令与上面的命令效果相同。

INSERT INTO customer\_t1 VALUES (3769, 'hello', 'Grace');

* 如果用户不知道所有字段的数值，可以忽略其中的一些。没有数值的字段将被填充为字段的缺省值。例如：

INSERT INTO customer\_t1 (c\_customer\_sk, c\_first\_name) VALUES (3769, 'Grace');

INSERT INTO customer\_t1 VALUES (3769, 'hello');

* 如果需要在表中插入多行，请使用以下命令：

INSERT INTO customer\_t1 (c\_customer\_sk, c\_customer\_id, c\_first\_name) VALUES(6885, 'maps', 'Joes'),

(4321, 'tpcds', 'Lily'),

(9527, 'world', 'James');

* 如果需要向表中插入多条数据，除此命令外，也可以多次执行插入一行数据命令实现。但是建议使用此命令可以提升效率。
* 如果从指定表插入数据到当前表，例如在数据库中创建了一个表customer\_t1的备份表customer\_t2，现在需要将表customer\_t1中的数据插入到表customer\_t2中，则可以执行如下命令。

CREATE TABLE customer\_t2

(

c\_customer\_sk integer,

c\_customer\_id char(5),

c\_first\_name char(6),

c\_last\_name char(8)

);

INSERT INTO customer\_t2 SELECT \* FROM customer\_t1;

**更新表中数据**

修改已经存储在数据库中数据的行为叫做更新。用户可以更新单独一行，所有行或者指定的部分行。还可以独立更新每个字段，而其他字段则不受影响。

使用UPDATE命令更新现有行，需要提供以下三种信息：

* 表的名称和要更新的字段名。
* 字段的新值。
* 要更新哪些行。

SQL通常不会为数据行提供唯一标识，因此无法直接声明需要更新哪一行。但是可以通过声明一个被更新的行必须满足的条件。只有在表里存在主键的时候，才可以通过主键指定一个独立的行。

建立表和插入数据的步骤请参考创建表与管理表。

**示例**

* 需要将表customer\_t1中c\_customer\_sk为9527的地域重新定义为9876：

UPDATE customer\_t1 SET c\_customer\_sk = 9876 WHERE c\_customer\_sk = 9527;

这里的表名称也可以使用模式名修饰，否则会从默认的模式路径找到这个表。SET后面紧跟字段和新的字段值。新的字段值不仅可以是常量，也可以是变量表达式。

* 把所有c\_customer\_sk的值增加100：

UPDATE customer\_t1 SET c\_customer\_sk = c\_customer\_sk + 100;

在这里省略了WHERE子句，表示表中的所有行都要被更新。如果出现了WHERE子句，那么只有匹配其条件的行才会被更新。在SET子句中的等号是一个赋值，而在WHERE子句中的等号是比较。WHERE条件不一定是相等测试，许多其他的操作符也可以使用。

* 用户可以在一个UPDATE命令中更新更多的字段，方法是在SET子句中列出更多赋值，比如：

UPDATE customer\_t1 SET c\_customer\_id = 'Admin', c\_first\_name = 'Local' WHERE c\_customer\_sk = 4421;

**查看数据**

在以下示例执行前，需按照管理表将示例对应数据表与数据进行创建与插入。

* 执行如下命令查询表customer\_t1的数据行数。

SELECT count(\*) FROM customer\_t1;

* 执行如下命令查询表customer\_t1的所有数据。

SELECT \* FROM customer\_t1;

* 执行如下命令只查询字段c\_customer\_sk的数据。

SELECT c\_customer\_sk FROM customer\_t1;

* 执行如下命令过滤字段c\_customer\_sk的重复数据。

SELECT DISTINCT( c\_customer\_sk ) FROM customer\_t1;

* 执行如下命令查询字段c\_customer\_sk为3869的所有数据。

SELECT \* FROM customer\_t1 WHERE c\_customer\_sk = 3869;

* 执行如下命令按照字段c\_customer\_sk进行排序。

SELECT \* FROM customer\_t1 ORDER BY c\_customer\_sk;

**删除表中数据**

在使用表的过程中，可能会需要删除已过期的数据，删除数据必须从表中整行的删除。

SQL不能直接访问独立的行，只能通过声明被删除行匹配的条件进行。如果表中有一个主键，用户可以指定准确的行。用户可以删除匹配条件的一组行或者一次删除表中的所有行。

**示例**

在以下示例执行前，需按照管理表将示例对应数据表与数据进行创建与插入。

* 使用DELETE命令删除行，如果删除表customer\_t1中所有c\_customer\_sk为3869的记录：

DELETE FROM customer\_t1 WHERE c\_customer\_sk = 3869;

* 如果执行如下命令之一，会删除表中所有的行。

DELETE FROM customer\_t2;

或者

TRUNCATE TABLE customer\_t2;

全表删除的场景下，建议使用truncate，不建议使用delete。

* 删除创建的表：

DROP TABLE customer\_t1;

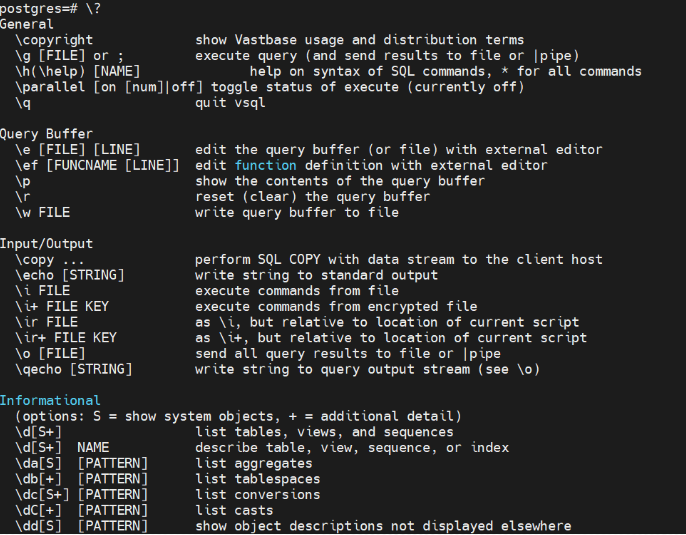
### 查看对象

vsql工具提供了元命令可帮助管理员查看数据库对象信息，常用查看对象元命令如下：

* 查看帮助信息，使用？查看所有可执行的命令，使用此命令输出结果较多，会分屏显示，可以按空格键分页，按q退出当前内容查看。

\?

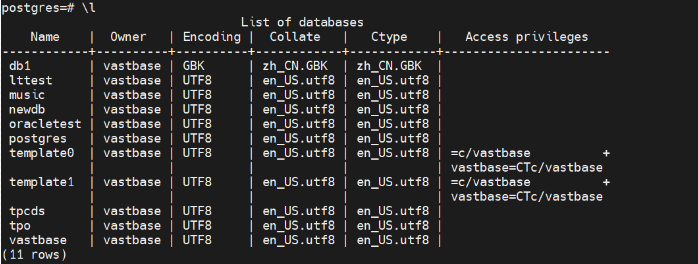
示例：



* 查看数据库（vsql客户端可用）。

\l

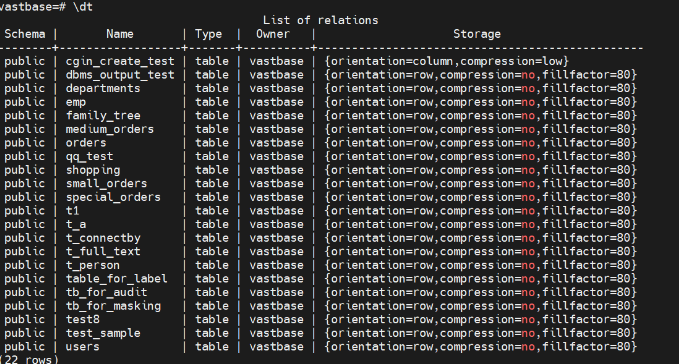
示例：



* 列举表。

\dt

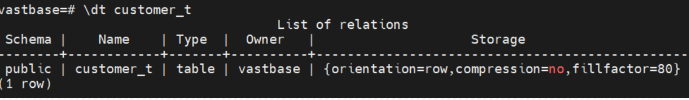
示例：



* 查看表结构。

\d tablename

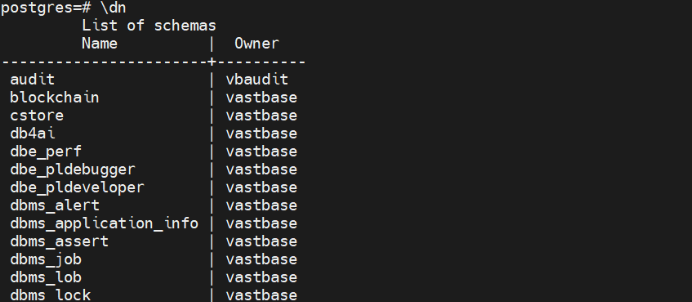
示例：



* 列举schema。

\dn

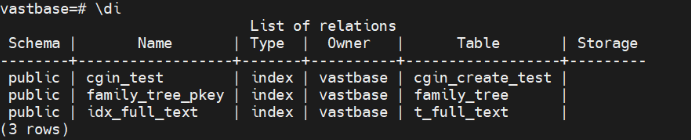
示例：



* 查看索引。

\di

示例：



* 切换数据库。

\c dbname

示例：

create database testnew;

\c testnew

### 配置运行参数

**查看参数当前取值**

Vastbase安装后，有一套默认的运行参数，为了使Vastbase与业务的配合度更高，用户需要根据业务场景和数据量的大小进行参数调整。

**操作步骤**

**步骤1** 使用如下命令，以操作系统用户vastbase登录数据库主节点。

vsql -d vastbase -p 5432

连接成功后，系统显示类似如下信息：

vsql ((Vastbase G100 V2.2 (Build 10) Alpha) compiled at 2022-08-15 21:50:21 commit 8730 last mr )

Non-SSL connection (SSL connection is recommended when requiring high-security)

Type "help" for help.

vastbase=#

**步骤2** 查看数据库运行参数当前取值。

* 方法一：使用SHOW命令。
* 使用如下命令查看单个参数：

SHOW max\_connections;

max\_connections显示数据库允许的最大连接数。

* 使用如下命令查看所有参数，使用此命令输出结果较多，会分屏显示，可以按空格键查看下一页，按q退出当前内容查看：

SHOW ALL;

* 方法二：使用pg\_settings视图。
* 使用如下命令查看单个参数：

SELECT \* FROM pg\_settings WHERE NAME='max\_connections';

* 使用如下命令查看所有参数，使用此命令输出结果较多，会分屏显示，可以按空格键分页，按q退出当前内容查看：

SELECT \* FROM pg\_settings;

**配置参数**

Vastbase提供了多种修改参数的方法，用户可以方便的针对数据库、用户、会话进行设置。

* 参数名称不区分大小写。
* 参数取值有整型、浮点型、字符串、布尔型和枚举型五类。
* 布尔值可以是（on，off）、（true，false）、（yes，no）或者（1，0），且不区分大小写。
* 枚举类型的取值是在系统表pg\_settings的enumvals字段取值定义的。
* 对于有单位的参数，在设置时请指定单位，否则将使用默认的单位。
* 参数的默认单位在系统表pg\_settings的unit字段定义的。
* 内存单位有：KB（千字节）、MB（兆字节）和GB（吉字节）。
* 时间单位：ms（毫秒）、s（秒）、min（分钟）、h（小时）和d（天）。

**参数分类**

Vastbase提供了六类参数，具体分类和设置方式请参考下表：

| **参数类型** | **说明** | **设置方式** |
| --- | --- | --- |
| INTERNAL | 固定参数，在创建数据库的时候确定，用户无法修改，只能通过show语法或者pg\_settings视图进行查看。 | 无 |
| POSTMASTER | 数据库服务端参数，在数据库启动时确定，可以通过配置文件指定。 | 支持参数设置中的方式一。 |
| SIGHUP | 数据库全局参数，可在数据库启动时设置或者在数据库启动后，发送指令重新加载。 | 支持参数设置中的方式一、方式二。 |
| BACKEND | 会话连接参数。在创建会话连接时指定，连接建立后无法修改。连接断掉后参数失效。内部使用参数，不推荐用户设置。 | 支持参数设置中的方式一、方式二。  说明：设置该参数后，下一次建立会话连接时生效。 |
| SUSET | 数据库管理员参数。可在数据库启动时、数据库启动后或者数据库管理员通过SQL进行设置。 | 支持参数设置中的方式一、方式二设置。 |
| USERSET | 普通用户参数。可被任何用户在任何时刻设置。 | 支持参数设置中的方式一、方式二设置。 |

**参数设置**

Vastbase提供了两种方式来修改参数，具体操作请参考下表：

| **序号** | **设置方法** |
| --- | --- |
| 方式一 | **编辑参数文件**  手工编辑参数文件postgresql.conf修改参数值，如果没有该参数则手动添加，重启数据库使参数生效。  说明：  重启vastbase操作会导致用户执行操作中断，请在执行之前规划好合适的执行窗口。  vb\_ctl stop && vb\_ctl srart |
| 方式二 | 修改指定数据库，用户，会话级别的参数。  设置数据库级别的参数，在下次会话中生效。  vastbase=# ALTER DATABASE dbname SET paraname TO value;  设置用户级别的参数，在下次会话中生效。  vastbase=# ALTER USER username SET paraname TO value;  设置会话级别的参数，修改本次会话中的取值。 退出会话后，设置  将失效。  vastbase=# SET paraname TO value; |

**操作步骤**

**方式一：**设置数据库参数，以在数据库主节点设置archive\_mode 参数为例。

**步骤1** 以操作系统用户vastbase登录数据库主节点。

vsql -d vastbase -p 5432

**步骤2** 查看archive\_mode参数。

cat /vastbase/data/dbnode/postgresql.conf | grep archive\_mode archive\_mode = on

注：on表示日志要进行归档操作。

**步骤3** 设置archive\_mode参数为off，直接编辑参数文件postgresql.conf，将参数设为off关闭日志的归档操作。

archive\_mode=off

**步骤4** 重启数据库使参数生效。

vb\_ctl stop && vb\_ctl start

**步骤5** 检查参数设置的正确性。

vastbase=# SHOW archive\_mode;

archive\_mode

--------------

off

(1 row)

**方式二：**设置参数，以设置explain\_perf\_mode参数为例。

**步骤1** 以操作系统用户vastbase登录数据库主节点。

vsql -d vastbase -p 5432

连接成功后，系统显示类似如下信息：

vsql ((Vastbase G100 V2.2 (Build 10) Alpha) compiled at 2022-08-15 21:50:21 commit 8730 last mr )

Non-SSL connection (SSL connection is recommended when requiring high-security)

Type "help" for help.

vastbase=#

**步骤2** 查看explain\_perf\_mode参数。

vastbase=# SHOW explain\_perf\_mode;

explain\_perf\_mode

-------------------

normal

(1 row)

**步骤3** 设置explain\_perf\_mode参数，使用以下任意方式进行置。

* 设置数据库级别的参数。

vastbase=# ALTER DATABASE vastbase SET explain\_perf\_mode TO pretty;

当结果显示为如下信息，则表示设置成功，在下次会话中生效。

ALTER DATABASE

* 设置用户级别的参数。

vastbase=# ALTER USER vuser SET explain\_perf\_mode TO pretty;

当结果显示为如下信息，则表示设置成功，在下次会话中生效。

ALTER ROLE

* 设置会话级别的参数。

vastbase=# SET explain\_perf\_mode TO pretty;

当结果显示为如下信息，则表示设置成功。

SET

**步骤4** 检查参数设置的正确性。

vastbase=# SHOW explain\_perf\_mode;

explain\_perf\_mode

--------------

pretty

(1 row)



电话：010-82838118

地址：北京市海淀区学院路30号科大天工大厦B座6层

官网：www.vastdata.com.cn