



链滴

TIDB 安装手册

作者: [ituac](#)

原文链接: <https://ld246.com/article/1564984086363>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



TiDB 作为一款开源分布式 NewSQL 数据库，可以很好的部署和运行在 Intel 架构服务器环境及主流虚拟化环境，并支持绝大多数的主流硬件网络。作为一款高性能数据库系统，TiDB 支持主流的 Linux 操作系统环境

Step 1: 准备工作

组件 例数量	CPU	内存	硬盘类型	
TIDB	16核+	32GB+	SAS	2
PD	4核+	8 GB+	SSD	3
TIKV	16核+	32GB+	SSD	3
监控	8核+	16GB+	SSD	1
中控机	-	-	-	1

上述，是我们所需要准备安装的实例分布，如果仅学习尚且没有这样实例体系的时候，TiDB、PD、控、中控机可以部署在同一台比较高配的实例机器中。下面是我接下来的安装步骤演示实例机器分布。

- 安装实例分布

组件	实例IP	组件描述	
TIDB (节点1) PD (节点1) 控机 网络用于下载软件包	172.26.106.34	中控机需要	
TIDB (节点2) PD (节点2) 2.26.106.35 同，只是带宽降低	同TIDB PD 中控机器配置		1
PD (节点2)	172.26.106.36		4

+16G		
TIKV (节点1)	172.26.106.37	1
核心+32 (官方建议16, 资源有限用12代替) 多增加额外50G数据云盘		
TIKV (节点2)	172.26.106.38	
TIKV (节点1)		
TIKV (节点3)	172.26.106.39	
TIKV (节点1)		
监控 (节点1)	172.26.106.40	8
+16G		

Step 2: 安装中控机以及应用中控机

- 中控机做什么？
 - 首先中控机可以安装在目标机器的某一台，也可以安装在独立的内网机器，用于控制所有机器
- 中控机需要什么？
 - 需要开放外网访问，用于下载软件相关依赖

现在通过172.26.106.34机器上 (CentOS 7) , root用户执行以下命令：

```
yum -y install epel-release git curl sshpass
yum -y install python2-pip
```

如果是 (Ubuntu) , 执行以下命令：

```
apt-get -y install git curl sshpass python2-pip
```

在中控机上创建tidb用户 (tidb用户用于控制tidb集群) , 并生成ssh key.

创建tidb用户并设置密码, root用户执行以下命令：

```
[root@ansiabale ~]# useradd -m -d /home/tidb tidb
[root@ansiabale ~]# passwd tidb
```

配置 tidb 用户 sudo 免密码

```
[root@ansiabale ~]# visudo
```

将 tidb ALL=(ALL) NOPASSWD: ALL 添加到文件末尾即可

生成 ssh key: 执行 su 命令从 root 用户切换到 tidb 用户下。

```
[root@ansiabale ~]# su - tidb
```

创建 tidb 用户 ssh key, 提示 Enter passphrase 时直接回车即可。执行成功后, ssh 私钥文件为 /home/tidb/.ssh/id_rsa, ssh 公钥文件为 /home/tidb/.ssh/id_rsa.pub

```
[tidb@ansiabale ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tidb/.ssh/id_rsa):
Created directory '/home/tidb/.ssh'.
```

```

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tidb/.ssh/id_rsa.
Your public key has been saved in /home/tidb/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:SMgAwGv235kzsHVOigXU7RWgsO4B1IYD+z9/JZITpp8 tidb@ansible
The key's randomart image is:
+---[RSA 2048]-----+
| =.. 000..... |
| . 0.0=.0. . |
| .0+=... . |
| +. +0.. . 0 |
| 0 . oo.S o o |
| ....0.+ + . . |
| ..B Bo. + o |
| + B .oE . |
| o .. |
+-----[SHA256]-----+

```

- 在中控机上下载Tidb-Ansible

以 `tidb` 用户登录中控机并进入 `/home/tidb` 目录。以下为 `tidb-ansible` 与 TiDB 的版本对应关系，本选择可以咨询官方。

TiDB 版本	tidb-ansible tag	备注
2.0 版本 0 稳定版本，可用于生产环境。	v2.0.10、v2.0.11	最新 2
2.1 版本 .1 稳定版本，可用于生产环境（建议）。	v2.1.1 ~ v2.1.8 等	最新
3.0 版本 前是 beta 版本，不建议用于生产环境。	v3.0.0-beta、v3.0.0-beta.1 等	
latest 版本 每日更新，不建议用于生产环境。	None	包含最新特性

下载指定 tag 的 `tidb-ansible`：

```
git clone https://github.com/pingcap/tidb-ansible.git
```

ps: 请务必按文档操作，将 `tidb-ansible` 下载到 `/home/tidb` 目录下，权限为 `tidb` 用户，不要下载到 `/root` 下，否则会遇到权限问题。

- 在中控机上安装Ansible及依赖

以 `tidb` 用户登录中控机，请务必按以下方式通过 `pip` 安装 Ansible 及其相关依赖的指定版本，否则有兼容问题。安装完成后，可通过 `ansible --version` 查看 Ansible 版本。目前 `release-2.0`、`release-2.1` 及 `master` 版本兼容 Ansible 2.4 及 Ansible 2.5 版本，Ansible 及相关依赖版本记录在 `tidb-ansible/requirements.txt` 文件中。

```

[tidb@ansible tidb-ansible]$ cd /home/tidb/tidb-ansible
[tidb@ansible tidb-ansible]$ sudo pip install -r ./requirements.txt
[tidb@ansible tidb-ansible]$ ansible --version

```

- 在中控机上配置部署ssh互信及sudo规则

以 tidb 用户登录中控机，将你的部署目标机器 IP 添加到 `hosts.ini` 文件 `[servers]` 区块下。

```
[tidb@ansiacle tidb-ansible]$ cd /home/tidb/tidb-ansible
[tidb@ansiacle tidb-ansible]$ vi hosts.ini
```

#修改vim host.ini的文件如下

```
[servers]
172.26.106.34
172.26.106.35
172.26.106.36
172.26.106.37
172.26.106.38
172.26.106.39
172.26.106.40
```

```
[all:vars]
username = tidb
ntp_server = pool.ntp.org
```

执行以下命令，按提示输入部署目标机器 `root` 用户密码。该步骤将在部署目标机器上创建 `tidb` 用户并配置 `sudo` 规则，配置中控机与部署目标机器之间的 `ssh` 互信。

```
[tidb@ansiacle tidb-ansible]$ ansible-playbook -i hosts.ini create_users.yml -u root -k
```

Step 3: 安装NTP服务

如果你的部署目标机器时间、时区设置一致，已开启 `NTP` 服务且在正常同步时间，此步骤可忽略。参考[如何检测 NTP 服务是否正常](#)。

该步骤将在部署目标机器上使用系统自带软件源联网安装并启动 `NTP` 服务，服务使用安装包默认的 `NTP server` 列表，见配置文件 `/etc/ntp.conf` 中 `server` 参数，如果使用默认的 `NTP server`，你的机器要连接外网。为了让 `NTP` 尽快开始同步，启动 `NTP` 服务前，系统会 `ntpdate` `hosts.ini` 文件中的 `ntp_server` 一次，默认为 `pool.ntp.org`，也可替换为你的 `NTP server`。

- 通过中控机，执行以下命令：

```
[tidb@ansiacle tidb-ansible]$ cd /home/tidb/tidb-ansible
[tidb@ansiacle tidb-ansible]$ ansible-playbook -i hosts.ini deploy_ntp.yml -u tidb -b
```

Step 4: CPUfreq 调节器模式

ps: tidb官方在这一步有CPUFreq调节模式，为让cpu发挥好性能，此步骤略过，因为我用的虚拟机试（虚拟机调试会遇到很多问题，偷个懒^_^）。有需要的话，可以参考官网进行，没有必要的话，以忽略

Step5: Tikv数据磁盘ext4文件系统挂载

ps: 这一步所有操作，我仅演示tikv的节点一，所有tikv节点同这个步骤

如果你的数据盘已经格式化成 `ext4` 并挂载，可先执行 `umount` 命令卸载，从编辑 `/etc/fstab` 文件步

开始执行，添加挂载参数重新挂载即可。（此步骤为配置tikv数据存储的磁盘，分配机器的时候，我意给tikv三个节点都增加了50个G的磁盘，用于存储数据，当然只是测试用，生产上磁盘大小看自家数据量吧）

```
[root@iZ8vbe0foda664humkdou3Z ~]# umount /dev/nvme0n1
```

下面以 /dev/nvme0n1 数据盘为例：

1. 查看数据盘

```
[root@iZ8vbe0foda664humkdou3Z ~]# fdisk -l
Disk /dev/nvme0n1: 53.7 GB, 53687091200 bytes, 104857600 sectors
```

2. 创建分区表

```
# parted -s -a optimal /dev/nvme0n1 mklabel gpt -- mkpart primary ext4 1 -1
```

3. 格式化文件系统

```
# mkfs.ext4 /dev/nvme0n1
```

4. 查看数据盘分区 UUID，本例中 nvme0n1 的 UUID 为8b79a29f-2bdc-4f0c-b028-a845aaad7016

```
[root@iZ8vbe0foda664humkdou3Z ~]# lsblk -f
NAME FSTYPE LABEL UUID MOUNTPOINT
vda
└─vda1 ext4 105fa8ff-bacd-491f-a6d0-f99865afc3d6 /
nvme0n1 ext4 8b79a29f-2bdc-4f0c-b028-a845aaad7016
```

5. 编辑 /etc/fstab 文件，添加 nodelalloc 挂载参数

```
[root@iZ8vbe0foda664humkdou3Z ~]# vi /etc/fstab
# 修改之后的fstab文件如下
# /etc/fstab
# Created by anaconda on Wed Jun 19 06:43:05 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=105fa8ff-bacd-491f-a6d0-f99865afc3d6 / ext4 defaults 1 1
UUID=8b79a29f-2bdc-4f0c-b028-a845aaad7016 /data ext4 defaults,nodelalloc,no
time 0 2
```

6. 挂载数据盘

```
[root@iZ8vbe0foda664humkdou3Z ~]# mkdir /data
[root@iZ8vbe0foda664humkdou3Z ~]# mount -a
```

7. 执行以下命令，如果文件系统为 ext4，并且挂载参数中包含 nodelalloc 表示生效：

```
[root@iZ8vbe0foda664humkdou3Z ~]# mount -t ext4
/dev/vda1 on / type ext4 (rw,relatime,data=ordered)
/dev/vdb on /data type ext4 (rw,noatime,nodelalloc,data=ordered)
```

Step 6: 分配机器资源

通过 tidb 用户登录中控机，inventory.ini 文件路径为 /home/tidb/tidb-ansible/inventory.ini

请使用内网 IP 来部署集群，如果部署目标机器 SSH 端口非默认 22 端口，需添加 `ansible_port` 变量如 TiDB1 `ansible_host=172.16.10.1 ansible_port=5555`

tidb 用户执行以下命令：

```
[tidb@ansiabile ~]$ cd /home/tidb/tidb-ansible/  
[tidb@ansiabile tidb-ansible]$ vi inventory.ini
```

##编辑后的inventory.ini如下

```
## TiDB Cluster Part
```

```
[tidb_servers]  
172.26.106.34  
172.26.106.35
```

```
[tikv_servers]  
172.26.106.37  
172.26.106.38  
172.26.106.39
```

```
[pd_servers]  
172.26.106.34  
172.26.106.35  
172.26.106.36
```

```
## Monitoring Part  
## prometheus and pushgateway servers  
[monitoring_servers]  
172.26.106.40
```

```
[grafana_servers]  
172.26.106.40
```

```
## node_exporter and blackbox_exporter servers  
[monitored_servers]  
172.26.106.34  
172.26.106.35  
172.26.106.36  
172.26.106.37  
172.26.106.38  
172.26.106.39  
172.26.106.40
```

```
[alertmanager_servers]  
172.26.106.40
```

- inventory.ini变量调整

部署目录通过 `deploy_dir` 变量控制，默认全局变量已设置为 `/home/tidb/deploy`，对所有服务生效如数据盘挂载目录为 `/data1`，可设置为 `/data1/deploy`，样例如下：

```
## Global variables  
[all:vars]  
deploy_dir = /data/deploy
```

如为某一服务单独设置部署目录，可在配置服务主机列表时配置主机变量，以 TiKV 节点为例，其他务类推，请务必添加第一列别名，以免服务混布时混淆。

```
TiKV1-1 ansible_host=172.16.10.4 deploy_dir=/data1/deploy
```

Step 7: 部署任务

ansible-playbook 执行 Playbook 时默认并发为 5，部署目标机器较多时可添加 -f 参数指定并发，如 `ansible-playbook deploy.yml -f 10`

1. 确认 `tidb-ansible/inventory.ini` 文件中 `ansible_user = tidb`，本例使用 `tidb` 用户作为服务运用户，配置如下：

执行以下命令如果所有 server 返回 `tidb` 表示 ssh 互信配置成功。

```
[tidb@ansible tidb-ansible]$ ansible -i inventory.ini all -m shell -a 'whoami'
[tidb@ansible tidb-ansible]$ ansible -i inventory.ini all -m shell -a 'whoami' -b
```

2. 执行 `local_prepare.yml` playbook，联网下载 TiDB binary 到中控机：

```
[tidb@ansible tidb-ansible]$ ansible-playbook local_prepare.yml
```

3. 初始化系统环境，修改内核参数

```
[tidb@ansible tidb-ansible]$ ansible-playbook bootstrap.yml
```

4. 部署 TiDB 集群软件

```
[tidb@ansible tidb-ansible]$ ansible-playbook deploy.yml
```

ps: 监控服务器上，Report用来生成监控的pdf，此功能依赖了fontconfig的字体，如果使用此功能需要在监控机器上下载依赖包，安装命令如下：

```
sudo yum install fontconfig open-sans-fonts
```

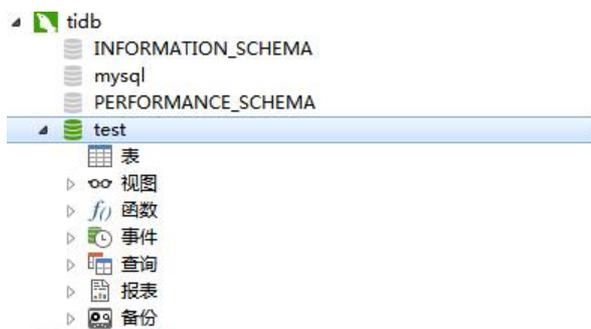
5. 启动 TiDB 集群

```
[tidb@ansible tidb-ansible]$ ansible-playbook start.yml
```

Step 8: 集群测试

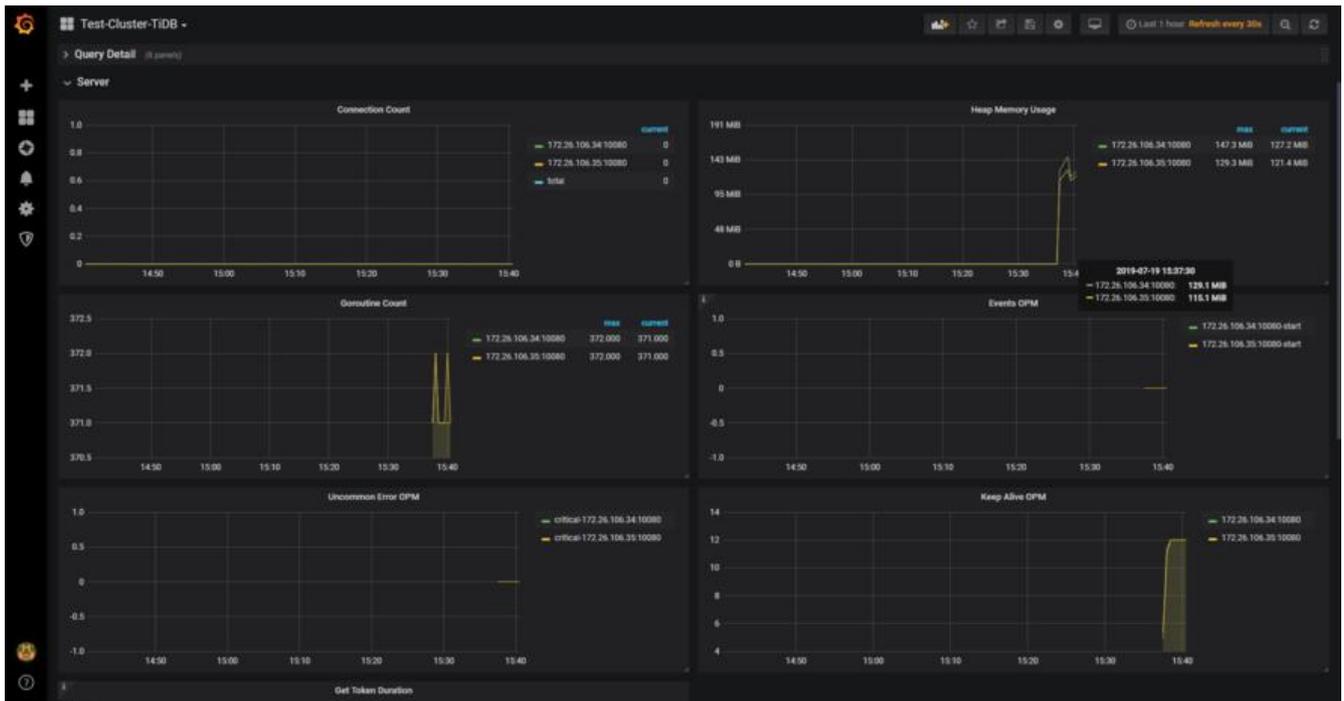
测试连接 TiDB 集群，推荐在 TiDB 前配置负载均衡来对外统一提供 SQL 接口。

- 使用mysql命令行或者mysql客户端工具，TiDB默认端口为4000（类似于mysql默认连接端口是336），通过tidb实例ip访问数据库172.26.106.34/172.26.106.35:4000 默认账号是:root 密码为空



- 通过浏览器测试监控平台

访问地址: <http://172.26.106.40> 默认帐号密码是: [admin/admin](#)



完工总结

整体搭建过程，除了对机器要求比较高，遇到的安装问题，基本都是和硬件有关或者是机器配置上的题，其次学习难度上并没有太复杂，希望你也可以顺利搭建完成，这里还有一篇文章是关于tidb安装见问题总结。