

第3天-LNMP生产环境部署

一、编译安装 MySQL

1、MySQL 官方链接

- MySQL官网：<https://nginx.mysql.com/>
- MySQL下载：<https://dev.mysql.com/downloads/mirrors/>
- MySQL文档：<https://dev.mysql.com/doc/relnotes/mysql/5.7/en/>
 - 注意：MySQL 每个版本都有对应的文档。上面的例子是MySQL5.7的文档。

2、MySQL 版本

- 版本分为四种：Alpha版、Beta版、RC版(Release Candidate)、GA版(Generally Available)。

1、Alpha 版

- Alpha版软件，这是软件工程对软件开发过程软件版本定义使用的版本说明。Alpha是内部测试版，一般不向外部发布，会有很多Bug，除非你也是测试人员，否则不建议使用。是希腊字母的第一位，表示最初级的版本，alpha 就是α。

2、Beta 版

- Beta版软件，这也是软件工程中对软件开发测试版本控制的版本说明。Beta一般是Alpha后面的版本。该版本相对于α版已有了很大的改进，消除了严重的错误，但还是存在着一缺陷，需要经过多次测试来进一步消除。这个阶段的版本会一直加入新的功能。beta 就是β。

3、RC 版

- RC版，RC即Release Candidate的简写。这是Beta后面的版本，一般RC版并没有新增功能，而是修复了一些反馈的Beta中存在的BUG。所以RC版更接近最终发行版即稳定版(GA版)

4、GA 版

- GA版，GA即Generally Available的简写。这就是软件最终的发行版。这个版本一般BUG相对较少。这个发行版也可以叫稳定版。

5、Release 版

- Release版，在有些软件存在，在MySQL中一般没有这个版本。该版本意味“最终版本”，在前面版本的一系列测试版之后，终归会有一个正式版本，是最终交付用户使用的版本。该版本有时也称为标准版。一般情况下，Release不会以单词形式出现在软件封面上，取而代之的是符号(R)。

3、MySQL 版本号

1、系统查询已安装MySQL版本方法

1、已登录MySQL方法

```
mysql> status;  
mysql> selectversion();  
mysql> select@@version;  
mysql> show variables like"%version%";
```

2、不登录直接查询方法

```
[root@qfedu.com ~]# mysql -u root -p #连接mysql  
[root@qfedu.com ~]# mysqladmin -uroot -p version;  
[root@qfedu.com ~]# mysql -V  
[root@qfedu.com ~]# mysql --version  
[root@qfedu.com ~]# mysql --help | grep Distrib$ mysql -? | grep Distrib$ mysql  
-I | grep Distrib  
[root@qfedu.com ~]# rpm -qa|grep mysql
```

3、MySQL 版本说明

以 MySQL 5.7.27 这个版本的版本号为例说明每个数字含义。

- 第一个数字(5)主版本号：文件格式改动时，将作为新的版本发布；
- 第二个数字(7)发行版本号：新增特性或者改动不兼容时，发行版本号需要更改；
- 第三个数字(27)发行序列号：主要是小的改动，如bug的修复、函数添加或更改、配置参数的更改等。

4、MySQL 类型

1、MySQL Community Server

- MySQL Community Server是社区版本，开源免费，但不提供官方技术支持。遵循GPL协议。MySQL Community Server也是我们通常用的MySQL的版本。根据不同的操作系统平台细分为多个版本。

2、MySQL Enterprise Edition

- MySQL Enterprise Edition企业版本，需付费，可以试用30天。

3、MySQL Cluster

- MySQL Cluster集群版，开源免费。可将几个MySQL Server封装成一个Server。MySQL Cluster CGE 高级集群版，需付费。

5、MySQL 安装方式

1、yum 安装

- 优点：操作简单易用。不用单独下载，服务器可以联网且yum源没有问题即可(可以选择国内的163/sohu/阿里源)

2、编译安装

- 5.1.X 及之前的版本是通过下载tar包以后解压后进入软件包解压路径。然后./configure、make、make install
- 5.4.X 到 5.7.X 通过下载tar包以后解压后进入软件包解压路径。然后 cmake、make、make install(cmake需要提前安装)
- 优点：可以定制功能特性。

3、二进制安装

- 官方下载二进制包，解压初始化即可直接使用不用安装。

4、rpm 安装

- 需要提前下载 rpm 软件包上传到服务器系统本地。
- 使用 rpm 或者 yum 命令直接安装

6、MySQL 编译安装

1、MySQL 编译环境

1、创建个 mysql 安装包目录

```
[root@qfedu.com ~]# mkdir -p /usr/local/src/mysql
```

2、安装编译环境依赖包

```
[root@qfedu.com ~]# yum -y install gcc gcc-c++ glibc automake autoconf libtool make
```

2、软件包下载

```
[root@qfedu.com ~]# cd /usr/local/src/mysql
[root@qfedu.com mysql]# wget
https://nchc.dl.sourceforge.net/project/boost/boost/1.59.0/boost_1_59_0.tar.gz
[root@qfedu.com mysql]# wget
https://github.com/Kitware/CMake/releases/download/v3.14.5/cmake-3.14.5.tar.gz
[root@qfedu.com mysql]# wget http://ftp.ntu.edu.tw/MySQL/Downloads/MySQL-
5.7/mysql-5.7.27.tar.gz
[root@qfedu.com mysql]# ls -l
total 143528
-rw-r--r--  1 root root 83709983 Feb 24 15:24 boost_1_59_0.tar.gz
-rw-r--r--  1 root root 8854929 Feb 24 15:24 cmake-3.14.5.tar.gz
-rw-r--r--  1 root root 54398696 Feb 24 15:24 mysql-5.7.27.tar.gz
```

3、安装 cmake

1、安装 cmake

```
[root@qfedu.com ~]# tar -zxvf cmake-3.14.5.tar.gz
[root@qfedu.com ~]# cd cmake-3.14.5
[root@qfedu.com ~]# ./bootstrap
[root@qfedu.com ~]# gmake
[root@qfedu.com ~]# gmake install
```

2、查看 cmake 版本

```
[root@qfedu.com ~]# cmake --version
```

4、解压 boost 库到/usr/local/目录

```
[root@qfedu.com mysql]# tar xf boost_1_59_0.tar.gz -C /usr/local/
```

```
[root@qfedu.com mysql]# cd /usr/local/
[root@qfedu.com local]# mv boost_1_59_0/ boost
[root@qfedu.com local]# cd boost
[root@qfedu.com boost]# ls -l
total 108
drwxr-xr-x 105 501 games 8192 Aug 12 2015 boost
-rw-r--r-- 1 501 games 850 Aug 11 2015 boost-build.jam
-rw-r--r-- 1 501 games 21920 Aug 11 2015 boostcpp.jam
-rw-r--r-- 1 501 games 989 Aug 11 2015 boost.css
-rw-r--r-- 1 501 games 6308 Aug 11 2015 boost.png
-rw-r--r-- 1 501 games 2477 Aug 11 2015 bootstrap.bat
-rwxr-xr-x 1 501 games 10631 Aug 11 2015 bootstrap.sh
drwx----- 7 501 games 84 Aug 12 2015 doc
-rw-r--r-- 1 501 games 769 Aug 11 2015 index.htm
-rw-r--r-- 1 501 games 5476 Aug 11 2015 index.html
-rw-r--r-- 1 501 games 291 Aug 11 2015 INSTALL
-rw-r--r-- 1 501 games 10779 Aug 11 2015 Jamroot
drwx----- 116 501 games 4096 Aug 12 2015 libs
-rw-r--r-- 1 501 games 1338 Aug 11 2015 LICENSE_1_0.txt
drwxr-xr-x 4 501 games 159 Aug 11 2015 more
-rw-r--r-- 1 501 games 2608 Aug 11 2015 rst.css
drwxr-xr-x 2 501 games 141 Aug 11 2015 status
drwxr-xr-x 10 501 games 190 Aug 11 2015 tools
```

5、编译安装 MySQL

1、安装编译 MySQL 依赖的包

```
[root@qfedu.com mysql]# yum install ncurses-devel -y
```

- MySQL 5.7依赖的软件包：gcc gcc-c++ ncurses ncurses-devel bison make等供参考，安装前可以使用 rpm -qa|grep \$(软件包名称)检查一下

2、创建安装MySQL相关目录

```
[root@qfedu.com mysql]# mkdir -p /database/mysql/{data,tmp,binlog,logs}
[root@qfedu.com mysql]# tree -L 2 /database/
/database/
├── mysql
│   ├── binlog
│   ├── data
│   ├── logs
│   └── tmp
5 directories, 0 files
```

3、创建 MySQL 组和用户

```
[root@qfedu.com mysql]# groupadd mysql
[root@qfedu.com mysql]# useradd mysql -s /sbin/nologin -M -g mysql
[root@qfedu.com mysql]# id mysql
uid=1000(mysql) gid=1000(mysql) groups=1000(mysql)
```

- s /sbin/nologin 用户不能登录系统
- M 不创建用户的HOME目录
- g 指定用户登录组的GID或组名

4、解压 MySQL 源码包

```
[root@qfedu.com mysql]# cd /usr/local/src/mysql
[root@qfedu.com mysql]# tar xf mysql-5.7.27.tar.gz
```

5、cmake 编译 MySQL

```
[root@qfedu.com mysql]# cd mysql-5.7.27
[root@qfedu.com mysql-5.7.27]# cmake . -DCMAKE_INSTALL_PREFIX=/database/mysql \
-DMySQL_DATADIR=/database/mysql/data \
-DMySQL_UNIX_ADDR=/database/mysql/tmp/mysql.sock \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci \
-DEXTRA_CHARSETS=gbk,gb2312,utf8,ascii \
-DENABLED_LOCAL_INFILE=ON \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DWITH_FEDERATED_STORAGE_ENGINE=1 \
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
-DWITHOUT_EXAMPLE_STORAGE_ENGINE=1 \
-DWITHOUT_PARTITION_STORAGE_ENGINE=1 \
-DWITH_ZLIB=bundled \
-DWITH_EMBEDDED_SERVER=1 \
-DWITH_DEBUG=0 \
-DWITH_BOOST=/usr/local/boost
```

- 看到如下信息，说明cmake编译完成

```
-- Configuring done
-- Generating done
-- Build files have been written to: /usr/local/src/mysql/mysqlmysql-5.7.27
```

6、安装 MySQL

```
[root@qfedu.com mysql-5.7.27]# make install
...省略
Linking CXX executable my_safe_process
[100%] Built target my_safe_process
```

7、MySQL 安装完成

```
...省略
-- Installing: /database/mysql/support-files/mysqld_multi.server
-- Installing: /database/mysql/support-files/mysql-log-rotate
-- Installing: /database/mysql/support-files/magic
-- Installing: /database/mysql/share/aclocal/mysql.m4
-- Installing: /database/mysql/support-files/mysql.server
```

- 如果上述操作未出现错误（每个步骤结束后，都可以使用echo \$?看返回值是否为0，为0则表示正确），查看/database/mysql/目录，若其下有内容，则表示MySQL5.7.27源代码包采用 cmake方式安装成功了

8、MySQL 编译常见参数选项说明

编译参数	说明
-DCMAKE_INSTALL_PREFIX=/database/mysql-5.7.27	设定mysql安装目录
-DMYSQL_DATADIR=/database/mysql-5.7.27/data	设定mysql数据文件目录
-DMYSQL_UNIX_ADDR=/database/mysql-5.7.27/tmp/mysql.sock	设定mysql.sock路径
-DDEFAULT_CHARSET=utf8	设定默认的字符集为utf8
-DDEFAULT_COLLATION=utf8_general_ci	设定默认排序规则
-DEXTRA_CHARSETS=gbk,gb2312,utf8,ascii	启用额外的字符集类型
-DENABLED_LOCAL_INFILE=ON	启用本地数据导入支持
-DWITH_INNOBASE_STORAGE_ENGINE=1	启用存储引擎；若想启用某个引擎的支持，-DWITHSTORAGEENGINE=1；若想禁用某个引擎的支持，-DWITHOUT_STORAGE_ENGINE=0
-DWITH_FEDERATED_STORAGE_ENGINE=1	启用存储引擎
-DWITH_BLACKHOLE_STORAGE_ENGINE=1	启用存储引擎
-DWITHOUT_EXAMPLE_STORAGE_ENGINE=1	启用存储引擎
-DWITHOUT_PARTITION_STORAGE_ENGINE=1	启用存储引擎
-DWITH_ZLIB=bundled	启用zlib库支持
-DWITH_EMBEDDED_SERVER=1	编译嵌入式服务器支持
-DWITH_DEBUG=0	禁用debug（默认为禁用）
-DWITH_BOOST=/usr/local/boost	启用boost库支持，MySQL 5.7版本编译安装需要依赖boost库

- 更多编译参数选项说明参见官方文档：https://dev.mysql.com/doc/refman/5.7/en/source-configuration-options.html#option_cmake_with_boost

6、配置 MySQL 系统环境变量

```
[root@qfedu.com ~]# echo 'export PATH=/database/mysql/bin:$PATH' >>/etc/profile
[root@qfedu.com ~]# tail -1 /etc/profile
export PATH=/database/mysql/bin:$PATH
```

- 使/etc/profile配置文件立即生效

```
[root@qfedu.com ~]# source /etc/profile
[root@qfedu.com ~]# echo $PATH
/database/mysql/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

7、初始化 MySQL 数据库

1、更改MySQL安装目录的属主和属组

```
[root@qfedu.com ~]# chown -R mysql:mysql /database/mysql/
```

2、修改 MySQL 配置文件的属主和属组

```
[root@qfedu.com ~]# chown mysql:mysql /etc/my.cnf
```

3、修改 MySQL 配置文件

- 修改/etc/my.cnf文件，和5.6之前的版本不一样，5.7版本support-files目录下没有.cnf文件，因此，需要自己手动编辑

```
[root@qfedu.com ~]# cp /etc/my.cnf /etc/my.cnf.bak
[root@qfedu.com ~]# vim /etc/my.cnf

[client]
port = 3306
socket = /database/mysql/tmp/mysql.sock
default-character-set = utf8

[mysqld]
port = 3306
user = mysql
basedir = /database/mysql
datadir = /database/mysql/data
pid-file = /database/mysql/mysqld.pid
socket = /database/mysql/tmp/mysql.sock
tmpdir = /database/mysql/tmp
character_set_server = utf8
server-id = 1
max_connections = 100
max_connect_errors = 10
sql_mode =
NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_AUTO_VALUE_ON_
ZERO,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,PIPES_AS_CONCAT,ANS
I_QUOTES
log-bin = /database/mysql/binlog/mysql-bin
log-error = /database/mysql/logs/mysql_5_7_27.err
```

4、MySQL 配置文件常用选项详解

```
[client]
# 默认连接端口
port = 3306
# 用于本地连接的socket套接字
socket = /database/mysql/tmp/mysql.sock
# 编码
default-character-set = utf8
```

```
[mysqld]
# 服务端口号, 默认3306
port = 3306
# mysql启动用户
user = mysql
# mysql安装根目录
basedir = /database/mysql
# mysql数据文件所在位置
datadir = /database/mysql/data
# pid文件所在目录
pid-file = /database/mysql/mysql.pid
# 为MySQL客户端程序和服务器之间的本地通讯指定一个套接字文件
socket = /database/mysql/tmp/mysql.sock
# MySQL存放临时文件的目录
tmpdir = /database/mysql/tmp
# 数据库默认字符集, 主流字符集支持一些特殊表情符号(特殊表情符占用4个字节)
character_set_server = utf8
# MySQL服务的唯一编号, 每个mysql服务Id需唯一
server-id = 1
# 最大连接数
max_connections = 100
# 最大错误连接数
max_connect_errors = 10
# sql_mode, 定义了mysql应该支持的sql语法, 数据校验
# NO_ENGINE_SUBSTITUTION: 如果需要的存储引擎被禁用或未编译, 可以防止自动替换存储引擎
# STRICT_TRANS_TABLES: 存储引擎启用严格模式, 非法数据值被拒绝
# NO_AUTO_CREATE_USER: 禁止GRANT创建密码为空的用户
# NO_AUTO_VALUE_ON_ZERO: 默认情况下自增长列是从1开始的, 如果你插入值为0的数据会报错, 设置这个之后就可以正常插入为0的数据了
# NO_ZERO_IN_DATE: 在严格模式, 不接受月或日部分为0的日期
# NO_ZERO_DATE: 在严格模式, 不要将 '0000-00-00' 做为合法日期
# ERROR_FOR_DIVISION_BY_ZERO: 在严格模式, 在INSERT或UPDATE过程中, 如果被零除(或MOD(X, 0)), 则产生错误(否则为警告)。如果未给出该模式, 被零除时MySQL返回NULL
# PIPES_AS_CONCAT: 将||视为字符串连接操作符(+) (同CONCAT()), 而不视为OR
# ANSI_QUOTES: 将''视为识别符引号('`'引号字符), 不要视为字符串的引号字符
sql_mode =
NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_AUTO_VALUE_ON_
ZERO,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,PIPES_AS_CONCAT,ANS
I_QUOTES
# 开启二进制日志功能, binlog数据位置
log-bin = /database/mysql/binlog/mysql-bin
# mysql生成的错误日志存放的路径
log-error = /database/mysql/logs/mysql_5_7_27.err
```

5、初始化 MySQL 数据库

```
[root@qfedu.com ~]# cd /database/mysql/
[root@qfedu.com mysql]# ./bin/mysqld --initialize-insecure --user=mysql --
basedir=/database/mysql --datadir=/database/mysql/data
```

- 生成初始化密码参数(5.7版本才有), 必须要添加, 否则会初始化失败, 会生成空密码
 - --initialize-insecure
- 设定初始化用户
 - --user=mysql
- 设定mysql工作目录

- --basedir=/database/mysql
- 设定数据文件目录
 - --datadir=/database/mysql/data

6、MySQL 实例开启SSL连接

```
[root@qfedu.com mysql]# cd /database/mysql/
[root@qfedu.com mysql]# ./bin/mysql_ssl_rsa_setup --initialize-insecure --
user=mysql --basedir=/database/mysql --datadir=/database/mysql/data
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ca-key.pem'
-----
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'server-key.pem'
-----
Generating a 2048 bit RSA private key
....+++
...+++
writing new private key to 'client-key.pem'
-----
```

7、修改 server-key.pem文件权限

```
[root@qfedu.com ~]# cd /database/mysql/data/
[root@qfedu.com data]# chmod +r server-key.pem
[root@qfedu.com data]# ls -l server-key.pem
-rw-r--r-- 1 root root 1675 Jul 31 20:01 server-key.pem
```

- 如果不给文件server-key.pem其他用户和组可读权限的话，当启动数据库的时候，会报如下错误
- 2019-08-07T06:39:24.111526Z 0 [ERROR] SSL error: Unable to get private key from 'server-key.pem'
- 2019-08-07T06:39:24.111555Z 0 [Warning] Failed to set up SSL because of the following SSL library error: Unable to get private key

8、启动MySQL数据库

- 拷贝启动脚本到/etc/init.d/目录下，并改名mysqld

```
[root@qfedu.com mysql]# cp support-files/mysql.server /etc/init.d/mysqld
[root@qfedu.com mysql]# ls -l /etc/init.d/mysqld
-rwxr-xr-x 1 root root 10588 Aug 1 18:33 /etc/init.d/mysqld
```

- 重新加载系统服务，将MySQL数据库加入开机自启动

```
[root@qfedu.com ~]# systemctl daemon-reload
[root@qfedu.com ~]# systemctl enable mysqld.service
mysqld.service is not a native service, redirecting to /sbin/chkconfig.
Executing /sbin/chkconfig mysqld on
[root@qfedu.com ~]# chkconfig mysqld on
```

- 启动MySQL数据库，并检查端口监听状态

```
[root@qfedu.com ~]# /etc/init.d/mysqld start
Starting MySQL. SUCCESS!
# 或者
[root@localhost mysql]# systemctl start mysqld

[root@localhost mysql]# systemctl status mysqld
• mysqld.service - LSB: start and stop MySQL
   Loaded: loaded (/etc/rc.d/init.d/mysqld; bad; vendor preset: disabled)
   Active: active (running) since Mon 2020-02-24 16:29:10 CST; 6s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 30683 ExecStart=/etc/rc.d/init.d/mysqld start (code=exited,
status=0/SUCCESS)
   CGroup: /system.slice/mysqld.service
           └─30696 /bin/sh /database/mysql/bin/mysqld_safe --
datadir=/database/mysql/data --pid-file=/database/mysq...
               └─30959 /database/mysql/bin/mysqld --basedir=/database/mysql --
datadir=/database/mysql/data --plugin-dir..

[root@qfedu.com ~]# ps aux | grep mysqld
root      30696  0.0  0.0 115436 1720 ?        S      16:29   0:00 /bin/sh
/database/mysql/bin/mysqld_safe --datadir=/database/mysql/data --pid-
file=/database/mysql/mysqld.pid
mysql     30959  0.3  4.1 1571428 167476 ?        sl     16:29   0:00
/database/mysql/bin/mysqld --basedir=/database/mysql --
datadir=/database/mysql/data --plugin-dir=/database/mysql/lib/plugin --
user=mysql --log-error=/database/mysql/logs/mysql_5_7_27.err --pid-
file=/database/mysql/mysqld.pid --socket=/database/mysql/tmp/mysql.sock --
port=3306
root      30990  0.0  0.0 112708  972 pts/0    S+    16:30   0:00 grep --
color=auto mysqld

[root@qfedu.com ~]# ss -anpt | grep 3306
tcp      LISTEN   0       70      :::3306      :::*
users: (("mysqld",pid=30959,fd=23))
```

- 默认CentOS 7默认没有netstat命令，而是使用ss命令代替；如果想使用netstat命令的话，需要执行yum install net-mysql进行安装net-mysql软件包，然后就可以使用了；

8、MySQL 数据库基本优化(安全)

- 账户设置密码，添加额外管理员，重新加载mysql授予表

```
[root@qfedu.com ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.27-log Source distribution

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> select user,host from mysql.user;
```

```

+-----+-----+
| user          | host          |
+-----+-----+
| mysql.session | localhost     |
| mysql.sys     | localhost     |
| root          | localhost     |
+-----+-----+
3 rows in set (0.00 sec)

```

```

mysql> grant all privileges on *.* to qfedu@'localhost' identified by
'Qfedu.123com' with grant option;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set password for 'mysql.session'@'localhost' = password('Qfedu.123com');
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set password for 'mysql.sys'@'localhost' = password('Qfedu.123com');
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set password for root@localhost = password('Qfedu.123com');
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

```

9、优雅的关闭 MySQL 方法

1、使用MySQL自带的脚本

```

[root@qfedu.com ~]# /etc/init.d/mysqld stop
Shutting down MySQL. SUCCESS!
[root@qfedu.com ~]# systemctl stop mysqld

```

2、使用 mysqladmin 的方法

```

[root@qfedu.com mysql]# mysqladmin -uqfedu -pQfedu.123com shutdown
mysqladmin: [warning] Using a password on the command line interface can be
insecure.
# mysqladmin:[警告]在命令行界面上使用密码可能不安全

```

- 5.7版本优化了数据库的安全配置，建议配置SSL加密连接，默认在命令行输入的密码都是明文传输的，极为不安全；

10、授权远程用户连接 MySQL

- 授权 qfedu 用户能够使用192.168.152网段对 MySQL 所有库和所有表具有 privileges 权限；

```

mysql> grant all privileges on *.* to qfedu1@'192.168.152.%' identified by
'Qfedu.123com' with grant option;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> select user,host from mysql.user;
+-----+-----+

```

user	host
qfedu1	192.168.152.%
mysql.session	localhost
mysql.sys	localhost
qfedu	localhost
root	localhost

5 rows in set (0.00 sec)

11、测试远程登陆

```
[root@qfedu.com ~]# mysql -h192.168.152.167 -uqfedu1 -pQfedu.123com -P3306
mysql: [warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.27-log Source distribution
```

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

二、编译安装 Nginx

Nginx简介：Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务，并在一个BSD-like 协议下发行。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、京东、新浪、网易、腾讯、淘宝等。

1、安装编译 Nginx 依赖包

```
[root@qfedu.com ~]# yum install -y gcc gcc-c++ pcre pcre-devel zlib zlib-devel
openssl openssl-devel perl-devel perl-ExtUtils-Embed gd-devel
```

- 编译依赖 gcc 环境，所以需要：gcc gcc-c++;
- PCRE(Perl Compatible Regular Expressions) 是一个Perl库，包括 perl 兼容的正则表达式库。nginx 的 http 模块使用 pcre 来解析正则表达式，所以需要在 linux 上安装 pcre 库，pcre-devel 是使用 pcre 开发的一个二次开发库，所以需要：pcre pcre-devel；

- zlib 库提供了很多种压缩和解压缩的方式，nginx 使用 zlib 对 http 包的内容进行 gzip，所以需要
在 Centos 上安装 zlib 库，所以需要：zlib zlib-devel；
- OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理
功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用。nginx 不仅支持 http 协议，还
支持 https（即在 ssl 协议上传输 http），所以需要在 Centos 安装 OpenSSL 库，所以需要：
openssl openssl-devel；

2、官网下载 Nginx 安装包

```
[root@qfedu.com ~]# mkdir -p /usr/local/src/nginx  
[root@qfedu.com ~]# cd /usr/local/src/nginx  
[root@qfedu.com nginx]# wget https://nginx.org/download/nginx-1.16.0.tar.gz
```

3、创建 Nginx 运行用户

```
[root@qfedu.com ~]# useradd -s /sbin/nologin -M nginx
```

4、创建 Nginx 临时文件目录

```
[root@qfedu.com ~]# mkdir -p /var/tmp/nginx/client/
```

5、解压配置 Nginx 编译

```
[root@qfedu.com nginx]# tar zxvf nginx-1.16.0.tar.gz  
[root@qfedu.com nginx]# cd nginx-1.16.0  
[root@qfedu.com nginx-1.16.0]# ./configure \  
--user=nginx \  
--group=nginx \  
--prefix=/usr/local/nginx \  
--conf-path=/etc/nginx/nginx.conf \  
--sbin-path=/usr/sbin/nginx \  
--error-log-path=/var/log/nginx/nginx_error.log \  
--http-log-path=/var/log/nginx/nginx_access.log \  
--pid-path=/usr/local/nginx/run/nginx.pid \  
--lock-path=/usr/local/nginx/lock/nginx \  
--with-http_image_filter_module \  
--with-http_ssl_module \  
--with-http_realip_module \  
--with-http_addition_module \  
--with-http_sub_module \  
--with-http_dav_module \  
--with-http_flv_module \  
--with-http_gzip_static_module \  
--with-http_stub_status_module \  
--with-http_perl_module \  
--with-mail \  
--with-mail_ssl_module \  
--with-pcre \  
--http-client-body-temp-path=/var/tmp/nginx/client/ \  
--http-proxy-temp-path=/var/tmp/nginx/proxy \  
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi \  
--http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \  
--http-scgi-temp-path=/var/tmp/nginx/scgi
```

6、Nginx 编译配置说明

```
[root@qfedu.com nginx-1.16.0]# ./configure
--user=nginx \ # worker进程运行用户
--group=nginx \ # worker进程运行的组
--prefix=/usr/local/nginx \ # Nginx安装的根路径，所有其他的路径
都要依赖于该选项
--conf-path=/etc/nginx/nginx.conf \ # 如果在命令行没有指定配置文件，那么
将会通过这里指定的路径，Nginx将会去那里查找它的配置文件
--sbin-path=/usr/sbin/nginx \ # 指定Nginx二进制文件的路径。如果没有
有指定，那么这个路径会依赖于--prefix选项
--error-log-path=/var/log/nginx/nginx_error.log \ # 指定错误文件的路径，Nginx将会往
其中写入错误日志文件，除非有其他配置
--http-log-path=/var/log/nginx/nginx_access.log \ # http访问日志的默认路径
--pid-path=/usr/local/nginx/run/nginx.pid \ # 指定的文件将会写入Nginx master进
程的pid，通常在/var/run下
--lock-path=/usr/local/nginx/lock/nginx \ # 共享存储器互斥锁文件的路径
--with-http_ssl_module \ # 启用 ngx_http_ssl_module 支持
(使支持 https 请求，需已安装openssl)
--with-http_realip_module \ # 启用 ngx_http_realip_module 支
持(这个模块允许从请求标头更改客户端的 IP 地址值，默认为关)
--with-http_addition_module \ # 启用 ngx_http_addition_module
支持(作为一个输出过滤器，支持不完全缓冲，分部分响应请求)
--with-http_sub_module \ # 启用 ngx_http_sub_module 支持
(允许用一些其他文本替换nginx 响应中的一些文本)
--with-http_dav_module \ #启用ngx_http_dav_module支持(增加
PUT,DELETE,MKCOL: 创建集合,COPY 和 MOVE 方法)默认情况下为关闭，需编译开启
--with-http_flv_module \ # 启用 ngx_http_flv_module 支持
(提供寻求内存使用基于时间的偏移量文件)
--with-http_gzip_static_module \ # 启用
ngx_http_gzip_static_module 支持(在线实时压缩输出数据流)
--with-http_stub_status_module \ # 启用ngx_http_stub_status_module
支持(获取nginx自上次启动以来的工作状态)
--with-http_perl_module \ # Nginx配置能够扩展使用Perl代码。这
个选项启用这个模块(然而使用这个模块会降低性能)
--with-mail \ # 该选项用于启用mail模块，该模块默认
没有被激活
--with-mail_ssl_module \ # 为了代理任何一种类型的使用SSL/TLS
的mail，激活该模块
--with-pcre \
--http-client-body-temp-path=/var/tmp/nginx/client/ \ # 从客户端收到请求后，该选项
设置的目录用于作为请求体零食存放的目录，如果webDAV模块启用，那么推荐设置该路径为同一文件系统
上的目录作为最终的目的地
--http-proxy-temp-path=/var/tmp/nginx/proxy \ # 在使用代理后，通过该选项设置存放临
时文件路径
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi \ # 设置FastCGI临时文件的目录
--http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \ # 设置uwsgi临时文件的目录
--http-scgi-temp-path=/var/tmp/nginx/scgi \ # 设置SCGI临时文件的目录
```

7、Nginx 编译安装

```
[root@qfedu.com nginx]# make -j2 && make install
```

8、Nginx 编译安装常见问题

1、源码编译会出现

```
./configure: error: perl module ExtUtils::Embed is required
```

- 解决方法

```
[root@qfedu.com ~]# yum -y install perl-ExtUtils-Embed
```

2、使用 Nginx 启动脚本出现

```
nginx: [emerg] getpwnam("nginx") failed
```

- 解决方法
 - nginx.conf 中去除 user nobody 注释即可
 - 错误原因是没有 nginx 这个用户，加上就好了
 - ```
[root@qfedu.com ~]# useradd -s /sbin/nologin -M nginx
```

## 3、启动 Nginx 出现

```
-bash: /etc/init.d/nginx: /bin/sh^M: bad interpreter: # 没有那个文件或目录
```

- 解决方法
  - 查看脚本格式：set ff
  - 显示 fileformat=doc
  - 改一下：set ff=unix 保存执行

## 9、测试 Nginx 是否安装成功

```
[root@qfedu.com ~]# nginx -V
nginx version: nginx/1.16.0
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
built with OpenSSL 1.0.2k-fips 26 Jan 2017
TLS SNI support enabled
configure arguments: --user=nginx --group=nginx --prefix=/usr/local/nginx --
conf-path=/etc/nginx/nginx.conf --sbin-path=/usr/sbin/nginx --error-log-
path=/var/log/nginx/nginx_error.log --http-log-
path=/var/log/nginx/nginx_access.log --pid-path=/usr/local/nginx/run/nginx.pid -
-lock-path=/usr/local/nginx/lock/nginx --with-http_image_filter_module --with-
http_ssl_module --with-http_realip_module --with-http_addition_module --with-
http_sub_module --with-http_dav_module --with-http_flv_module --with-
http_gzip_static_module --with-http_stub_status_module --with-http_perl_module -
-with-mail --with-mail_ssl_module --with-pcre --http-client-body-temp-
path=/var/tmp/nginx/client/ --http-proxy-temp-path=/var/tmp/nginx/proxy --http-
fastcgi-temp-path=/var/tmp/nginx/fcgi --http_uwsgi-temp-
path=/var/tmp/nginx/uwsgi --http_scgi-temp-path=/var/tmp/nginx/scgi
```

## 10、启动 Nginx 服务

```
[root@qfedu.com ~]# /usr/sbin/nginx
```

## 11、验证 Nginx 服务是否启动成功

```
[root@qfedu.com ~]# netstat -ntlp | grep nginx
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
349/nginx: master
```

## 12、系统添加 Nginx 服务

### 1、以 systemd 形式添加

#### 1、创建 nginx.service 文件

```
[root@qfedu.com ~]# vim /lib/systemd/system/nginx.service
[Unit]
Description=nginx
After=network.target

[Service]
Type=forking
ExecStart=/usr/sbin/nginx
ExecReload=/usr/sbin/nginx -s reload
ExecStop=/usr/sbin/nginx -s quit
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

#### 2、以 systemctl 方式启动 Nginx

```
[root@qfedu.com ~]# pkill nginx
[root@qfedu.com ~]# systemctl daemon-reload
[root@qfedu.com ~]# systemctl start nginx
```

#### 3、查看 Nginx 服务状态

```
[root@qfedu.com ~]# systemctl status nginx
• nginx.service - nginx
 Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
 Active: active (running) since Mon 2019-04-29 23:19:39 EDT; 18min ago
 Process: 348 ExecStart=/usr/local/nginx/sbin/nginx (code=exited, status=0/SUCCESS)
 Main PID: 349 (nginx)
 Tasks: 2
 Memory: 976.0K
 CGroup: /system.slice/nginx.service
 └─349 nginx: master process /usr/local/nginx/sbin/nginx
 └─350 nginx: worker process

Apr 29 23:19:39 localhost.localdomain systemd[1]: Starting nginx...
Apr 29 23:19:39 localhost.localdomain systemd[1]: Started nginx.
```

#### 4、验证 Nginx 服务是否成功启动



```
[root@qfedu.com ~]# netstat -ntlp | grep nginx
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
349/nginx: master p
```

## 5、配置 Nginx 服务自动启动

```
[root@qfedu.com ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service
to /usr/lib/systemd/system/nginx.service.
```

## 2、以启动脚本的形式添加 Nginx 服务

### 1、创建 Nginx 启动脚本

```
[root@qfedu.com ~]# vim /etc/rc.d/init.d/nginx
#!/bin/bash
#
nginx - this script starts and stops the nginx daemon
#
chkconfig: - 85 15
description: Nginx is an HTTP(S) server, HTTP(S) reverse \
proxy and IMAP/POP3 proxy server
processname: nginx
config: /etc/nginx/nginx.conf
pidfile: /usr/local/nginx/run/nginx.pid

Source function library.
. /etc/rc.d/init.d/functions

Source networking configuration.
. /etc/sysconfig/network-scripts

Check that networking is up.
["$NETWORKING" = "no"] && exit 0

nginx="/usr/sbin/nginx"
prog=$(basename $nginx)

NGINX_CONF_FILE="/etc/nginx/nginx.conf"

lockfile=/usr/local/nginx/lock/nginx

start() {
 [-x $nginx] || exit 5
 [-f $NGINX_CONF_FILE] || exit 6
 echo -n "Starting $prog: "
 daemon $nginx -c $NGINX_CONF_FILE
 retval=$?
 echo
 [$retval -eq 0] && touch $lockfile
 return $retval
}

stop() {
 echo -n "Stopping $prog: "
 killproc $prog -QUIT
```

```

 retval=$?
 echo
 [$retval -eq 0] && rm -f $lockfile
return $retval
}

restart() {
 configtest || return $?
 stop
 start
}

reload() {
 configtest || return $?
 echo -n "Reloading $prog: "
 killproc $nginx -HUP
 RETVAL=$?
 echo
}

force_reload() {
 restart
}

configtest() {
 $nginx -t -c $NGINX_CONF_FILE
}

rh_status() {
 status $prog
}

rh_status_q() {
 rh_status >/dev/null 2>&1
}

case "$1" in
 start)
 rh_status_q && exit 0
 $1
 ;;
 stop)
 rh_status_q || exit 0
 $1
 ;;
 restart|configtest)
 $1
 ;;
 reload)
 rh_status_q || exit 7
 $1
 ;;
 force-reload)
 force_reload
 ;;
 status)
 rh_status
 ;;

```

```

condrestart|try-restart)
 rh_status_q || exit 0
 ;;
*)
 echo $"Usage: $0 {start|stop|status|restart|condrestart|try-
restart|reload|force-reload|configtest}"
 exit 2
esac

```

## 2、Nginx 启动脚本添加执行权限

```
[root@qfedu.com ~]# chmod +x /etc/rc.d/init.d/nginx
```

## 3、Nginx 设置开机启动

```

[root@qfedu.com ~]# chkconfig --add nginx
[root@qfedu.com ~]# chkconfig nginx on

```

## 4、Nginx 服务管理命令

```

[root@qfedu.com ~]# systemctl daemon-reload
[root@qfedu.com ~]# systemctl restart nginx
[root@qfedu.com ~]# systemctl status nginx.service
[root@qfedu.com ~]# systemctl stop nginx.service

[root@qfedu.com ~]# service nginx start
[root@qfedu.com ~]# service nginx stop
[root@qfedu.com ~]# service nginx reload

```

## 5、查看有关参数

```

[root@qfedu.com ~]# nginx -v
nginx version: nginx/1.16.0
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
built with OpenSSL 1.0.2k-fips 26 Jan 2017
TLS SNI support enabled
configure arguments: --user=nginx --group=nginx --prefix=/usr/local/nginx --
conf-path=/etc/nginx/nginx.conf --sbin-path=/usr/sbin/nginx --error-log-
path=/var/log/nginx/nginx_error.log --http-log-
path=/var/log/nginx/nginx_access.log --pid-path=/usr/local/nginx/run/nginx.pid -
-lock-path=/usr/local/nginx/lock/nginx --with-http_image_filter_module --with-
http_ssl_module --with-http_realip_module --with-http_addition_module --with-
http_sub_module --with-http_dav_module --with-http_flv_module --with-
http_gzip_static_module --with-http_stub_status_module --with-http_perl_module -
-with-mail --with-mail_ssl_module --with-pcre --http-client-body-temp-
path=/var/tmp/nginx/client/ --http-proxy-temp-path=/var/tmp/nginx/proxy --http-
fastcgi-temp-path=/var/tmp/nginx/fcgi --http-uwsgi-temp-
path=/var/tmp/nginx/uwsgi --http-scgi-temp-path=/var/tmp/nginx/scgi

```

## 6、验证 Nginx 服务是否成功启动

```

[root@qfedu.com ~]# netstat -ntlp | grep nginx
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
349/nginx: master p

```

## 13、Nginx 配置文件解读

```
user nginx; # 运行用户
worker_processes 1; # 启动进程，通常设置成和cpu的数量相等
error_log /var/log/nginx/error.log; # 全局错误日志
pid /usr/local/nginx/run/nginx.pid; # pid文件
log_format access '$remote_addr - $remote_user [$time_local] "$request" '
 '$status $body_bytes_sent "$http_referer" '
 '"$http_user_agent" $http_x_forwarded_for';

日志格式,以下具体介绍变量
$remote_addr 与$http_x_forwarded_for # 用以记录客户端的 ip 地址;
$remote_user # 用来记录客户端用户名称;
$time_local # 用来记录访问时间与时区;
$request # 用来记录请求的 url 与 http 协议;
$status # 用来记录请求状态; 成功是 200;
$body_bytes_sent # 记录发送给客户端文件主体内容大小;
$http_referer # 用来记录从那个页面链接访问过来的;

工作模式及连接数上限
events {
 use epoll; # epoll 是多路复用 IO(I/O Multiplexing)中的一种方式,但是仅用于 linux2.6以上内核,可以大大提高 nginx 的性能
 worker_connections 1024; # 单个后台 worker process 进程的最大并发链接数
 # multi_accept on;
}

设定 http 服务器, 利用它的反向代理功能提供负载均衡支持
http {
 # 设定 mime 类型,类型由 mime.type 文件定义
 include /etc/nginx/mime.types;
 default_type application/octet-stream;

 # 设定日志格式
 access_log /usr/local/nginx/logs/access.log;

 # sendfile 指令指定 nginx 是否调用 sendfile 函数 (zero copy 方式) 来输出文件, 对于普通应用,
 # 必须设为 on,如果用来进行下载等应用磁盘 IO 重负载应用,可设置为 off, 以平衡磁盘与网络 I/O 处理速度,降低系统的 uptime.
 sendfile on;
 #tcp_nopush on;

 # 连接超时时间
 keepalive_timeout 0;
 keepalive_timeout 65;
 tcp_nodelay on;

 # 开启 gzip 压缩
 gzip on;
 gzip_disable "MSIE [1-6]\.(?!.*SV1)";

 # 设定请求缓冲
 client_header_buffer_size 1k;
 large_client_header_buffers 4 4k;
 include /etc/nginx/conf.d/*.conf;
 include /etc/nginx/sites-enabled/*;

 # 设定负载均衡的服务器列表
 upstream mysvr {
```

```

weighth 参数表示权值，权值越高被分配到的几率越大
server 192.168.8.1:3128 weight=5;
server 192.168.8.2:80 weight=1;
server 192.168.8.3:80 weight=6;
}
server {
 # 侦听 80 端口
 listen 80;
 # 定义使用 nginx.xx.com 访问
 server_name nginx.xxx.com;
 # 设定本虚拟主机的访问日志
 access_log /var/log/nginx/nginx.xxx.com_access.log access;
默认请求
 location / {
 root /var/nginx/html/; #定义服务器的默认网站根目录位置
 index index.php index.html index.htm; #定义首页索引文件的名称
 }
定义错误提示页面
 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root /root;
 }
静态文件，nginx 自己处理
 location ~ ^/(images|javascript|js|css|flash|media|static)/ {
 root /var/nginx/virtual/htdocs;
 #过期 30 天，静态文件不怎么更新，过期可以设大一点，如果频繁更新，则可以设
置得小一点。
 expires 30d;
 }
PHP 脚本请求全部转发到 FastCGI 处理。使用 FastCGI 默认配置。
 location ~ \.php$ {
 root /root;
 fastcgi_pass 127.0.0.1:9000;
 fastcgi_index index.php;
 fastcgi_param
SCRIPT_FILENAME/home/nginx/nginx$fastcgi_script_name;
 include fastcgi_params;
 }
设定查看 Nginx 状态的地址
 location /NginxStatus {
 stub_status on;
 access_log on;
 auth_basic "NginxStatus";
 auth_basic_user_file conf/htpasswd;
 }
禁止访问 .htxxx 文件
 location ~ /\.ht {
 allow host;
 deny all;
 }
}
}

```

## 14、Nginx 基本配置使用

### 1、Nginx 基于域名的虚拟主机

```
server {
 listen 80;
 server_name nginx.qf.edu qf.edu;
 access_log /var/log/nginx/Daniel/Daniel_access.log main;
 error_log /var/log/nginx/Daniel/Daniel_error.log crit;
 location / {
 root html/nginx;
 index index.html index.htm;
 }
 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root html;
 }
}
```

- 多域名就多个server段
- 基于端口的只需要将listen修改端口即可，server\_name也要修改成IP
- 别名：server\_name 中有两个域名，这个就是别名

## 2、Nginx 状态信息配置

```
server{
 listen 80;
 server_name status.qf.edu;
 location / {
 stub_status on;
 access_log off;
 }
}
```

- 浏览器访问 status.qf.edu
  - Active connections: 3 ：正在处理的活动连接数
  - server accepts handled requests ：server表示nginx启动到现在共处理了几个连接，accepts表示nginx启动到现在共成功创建了几次连接，请求丢失数据=（握手数-连接数），可以看出，本次状态显示没有丢失请求，handled requests表示总共处理了几次请求  
9 9 192
  - Reading: 0 Writing: 1 Waiting: 2
    - Reading：nginx 读取客户端的Header信息数
    - Writing：nginx 返回给客户端的Header信息数
    - Waiting：nginx 已经处理完正在等候下一次请求指令的驻留连接，开启keep-alive的情况下，这个值等于active-(reading + writing)

## 3、Nginx 访问控制权限

```
server {
 listen 80;
 server_name nginx.etiantian.org etiantian.org;
 access_log /var/log/nginx/etiantian/etiantian_access.log main;
 error_log /var/log/nginx/etiantian/etiantian_error.log crit;
 location / {
 root html/nginx;
 }
}
```

```

 index index.html index.htm;
 deny 172.16.50.173; # 拒绝172.16.50.173
 allow all; # 允许全部，意思就是允许除了172.16.50.173的
其他全部IP访问
 }
 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root html;
 }

server{
 listen 80;
 server_name status.qf.edu;
 location / {
 stub_status on;
 access_log off;
 allow 172.16.50.173; # 只允许172.16.50.173，其他全部拒绝，也可以用
网段表示:172.16.50.0/24
 deny all;
 }
}

```

## 15、Nginx 编译添加新的模块

- 已经安装好的nginx，需要添加一个未被编译安装的模块，以安装第三方 ngx\_http\_google\_filter\_module 模块为例

### 1、下载第三方扩展模块

```

[root@qfedu.com ~]# git clone https://github.com/cuber/nginx_http_google_filter_module # 先安装git

```

### 2、查看 nginx 编译安装已安装模块

```

[root@qfedu.com ~]# nginx -v
nginx version: nginx/1.16.0
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
built with OpenSSL 1.0.2k-fips 26 Jan 2017
TLS SNI support enabled
configure arguments: --user=nginx --group=nginx --prefix=/usr/local/nginx --
conf-path=/etc/nginx/nginx.conf --sbin-path=/usr/sbin/nginx --error-log-
path=/var/log/nginx/nginx_error.log --http-log-
path=/var/log/nginx/nginx_access.log --pid-path=/usr/local/nginx/run/nginx.pid -
-lock-path=/usr/local/nginx/lock/nginx --with-http_image_filter_module --with-
http_ssl_module --with-http_realip_module --with-http_addition_module --with-
http_sub_module --with-http_dav_module --with-http_flv_module --with-
http_gzip_static_module --with-http_stub_status_module --with-http_perl_module -
-with-mail --with-mail_ssl_module --with-pcre --http-client-body-temp-
path=/var/tmp/nginx/client/ --http-proxy-temp-path=/var/tmp/nginx/proxy --http-
fastcgi-temp-path=/var/tmp/nginx/fcgi --http-uwsgi-temp-
path=/var/tmp/nginx/uwsgi --http-scgi-temp-path=/var/tmp/nginx/scgi

```

### 3、添加模块重新编译配置

- 加入需要安装的模块，重新编译，如这里添加 `--add-module=/data/software/nginx_http_google_filter_module`

```
[root@qfedu.com nginx-1.16.0]# ./configure \
--user=nginx \
--group=nginx \
--prefix=/usr/local/nginx \
--conf-path=/etc/nginx/nginx.conf \
--sbin-path=/usr/sbin/nginx \
--error-log-path=/var/log/nginx/nginx_error.log \
--http-log-path=/var/log/nginx/nginx_access.log \
--pid-path=/usr/local/nginx/run/nginx.pid \
--lock-path=/usr/local/nginx/lock/nginx \
--with-http_image_filter_module \
--with-http_ssl_module \
--with-http_realip_module \
--with-http_addition_module \
--with-http_sub_module \
--with-http_dav_module \
--with-http_flv_module \
--with-http_gzip_static_module \
--with-http_stub_status_module \
--with-http_perl_module \
--with-mail \
--with-mail_ssl_module \
--with-pcre \
--http-client-body-temp-path=/var/tmp/nginx/client/ \
--http-proxy-temp-path=/var/tmp/nginx/proxy \
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi \
--http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \
--http-scgi-temp-path=/var/tmp/nginx/scgi \
--add-module=/data/software/nginx_http_google_filter_
module
```

## 4、重新编译 Nginx

- 只需要 `make`，不需要 `make install`

```
[root@qfedu.com nginx-1.16.0]# make
```

## 5、替换 Nginx 二进制文件

### 1、备份原来的 Nginx 二进制文件

```
[root@qfedu.com nginx-1.16.0]# cp /usr/sbin/nginx /usr/sbin/nginx.bak
```

### 2、移动编译好的新的 Nginx

```
[root@qfedu.com nginx-1.16.0]# cp ./nginx /usr/sbin/
```

## 6、重载 Nginx 服务

```
[root@qfedu.com ~]# systemctl reload nginx
```

## 7、验证 Nginx 服务是否启动成功



```
[root@qfedu.com ~]# netstat -ntlp | grep nginx
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
349/nginx: master
```

## 8、测试 Nginx 新模块是否安装成功

```
[root@qfedu.com ~]# nginx -V
```

# 三、编译安装 Php

## 1、安装编译环境依赖包

```
[root@qfedu.com ~]# yum -y install gcc gcc-c++ glibc automake autoconf libtool make
```

## 2、创建安装目录

```
[root@qfedu.com ~]# mkdir -p /usr/local/php7
```

## 3、安装编译 php 依赖库

```
[root@qfedu.com ~]# yum -y install libxslt-devel libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 bzip2-devel ncurses ncurses-devel curl curl-devel e2fsprogs e2fsprogs-devel krb5-devel libidn libidn-devel openssl openssl-devel
```

## 4、创建安装包目录

```
[root@qfedu.com ~]# mkdir -p /usr/local/src/php
```

## 5、安装 cmake

### 1、下载 cmake

```
[root@qfedu.com ~]# cd /usr/local/src/php
[root@qfedu.com ~]# wget
https://github.com/Kitware/CMake/releases/download/v3.14.5/cmake-3.14.5.tar.gz
```

### 2、安装 cmake

```
[root@qfedu.com ~]# tar -zxvf cmake-3.14.5.tar.gz
[root@qfedu.com ~]# cd cmake-3.14.5
[root@qfedu.com ~]# ./bootstrap
[root@qfedu.com ~]# gmake
[root@qfedu.com ~]# gmake install
```

### 3、查看 cmake 版本

```
[root@qfedu.com ~]# cmake --version
```

## 6、安装 libzip

### 1、下载 libzip

```
[root@qfedu.com ~]# cd /usr/local/src/php
[root@qfedu.com php]# wget https://libzip.org/download/libzip-1.5.2.tar.gz
```

### 2、安装 libzip

```
[root@qfedu.com php]# tar -zxvf libzip-1.5.2.tar.gz
[root@qfedu.com php]# cd libzip-1.5.2/
[root@qfedu.com libzip-1.5.2]# mkdir build
[root@qfedu.com libzip-1.5.2]# cd build
[root@qfedu.com build]# cmake ..
[root@qfedu.com build]# make -j2
[root@qfedu.com build]# make test
[root@qfedu.com build]# make install
```

## 7、编译安装 Php

### 1、下载 Php 源码包

```
[root@qfedu.com ~]# cd /usr/local/src/php
[root@qfedu.com php]# wget https://www.php.net/distributions/php-7.3.6.tar.gz
```

### 2、配置 Php 编译

```
[root@qfedu.com php]# tar -zxvf php-7.3.6.tar.gz
[root@qfedu.com php]# cd php-7.3.6
[root@qfedu.com php-7.3.6]# ./configure \
--prefix=/usr/local/php7 \
--with-config-file-path=/usr/local/php7 \
--with-config-file-scan-dir=/usr/local/php7/php.d \
--enable-mysqlnd \
--with-mysqli \
--with-pdo-mysql \
--enable-fpm \
--with-fpm-user=nginx \
--with-fpm-group=nginx \
--with-gd \
--with-iconv \
--with-zlib \
--enable-xml \
--enable-shmop \
--enable-sysvsem \
--enable-inline-optimization \
--enable-mbregex \
--enable-mbstring \
--enable-ftp \
--with-openssl \
--enable-pcntl \
--enable-sockets \
--with-xmldrpc \
--enable-zip \

```

```
--enable-soap \
--without-pear \
--with-gettext \
--enable-session \
--with-curl \
--with-jpeg-dir \
--with-freetype-dir \
--enable-opcache
```

### 3、编译中 off\_t 问题解决

```
[root@qfedu.com php-7.3.6]# vim /etc/ld.so.conf
添加如下几行
/usr/local/lib64
/usr/local/lib
/usr/lib
/usr/lib64
使配置生效
[root@qfedu.com php-7.3.6]# ldconfig -v
```

### 4、Php 编译参数说明

```
--prefix=/usr/local/php7 # 配置安装目录
--with-config-file-path=/usr/local/php7 # 配置文件 php.ini 的路径
--enable-sockets # 开启 socket
--enable-fpm # 启用 fpm 扩展
--enable-cli # 启用 命令行模式 (从 php 4.3.0 之后这个模块默认开启所以可以不用再加此命令)
--enable-mbstring # 启用 mbstring 库
--enable-pcntl # 启用 pcntl (仅 CLI / CGI)
--enable-soap # 启用 soap
--enable-opcache # 开启 opcache 缓存
--disable-fileinfo # 禁用 fileinfo (由于 5.3+ 之后已经不再持续维护了, 但默认是开启的, 所以还是禁止了吧)(1G以下内存服务器直接关了吧)
--disable-rpath # 禁用搜索路径中传递其他运行库。
--with-mysqli # 启用 mysqli 扩展
--with-pdo-mysql # 启用 pdo 扩展
--with-iconv-dir # 启用 XMLRPC-EPI 字符编码转换 扩展
--with-openssl # 启用 openssl 扩展 (需要 openssl openssl-devel)
--with-fpm-user=nginx # 设定 fpm 所属的用户
--with-fpm-group=nginx # 设定 fpm 所属的组别
--with-curl # 启用 curl 扩展
--with-mhash # 开启 mhash 基于离散数学原理的不可逆向的php加密方式扩展库
GD
--with-gd # 启用 GD 图片操作 扩展
--with-jpeg-dir # 开启对 jpeg 图片的支持 (需要 libjpeg)
--with-png-dir # 开启对 png 图片支持 (需要 libpng)
--with-freetype-dir # 开启 freetype
压缩
--enable-zip # 启用 zip
--with-zlib # 启用对 zlib 支持
xml
--enable-simplexml # 启用对 simplexml 支持
--with-libxml-dir # 启用对 libxml2 支持
```

```
--enable-debug
```

```
开启 debug 模式
```

## 5、编译安装 Php

```
[root@qfedu.com php-7.3.6]# make && make install
```

## 6、设置 Php 配置文件

```
[root@qfedu.com php-7.3.6]# cp /usr/local/src/php/php-7.3.6/php.ini-production /usr/local/php7/etc/php.ini
```

## 7、设置php-fpm配置文件

```
[root@qfedu.com php-7.3.6]# cd /usr/local/php7/etc
[root@qfedu.com etc]# cp php-fpm.conf.default php-fpm.conf
[root@qfedu.com etc]# vim /usr/local/php7/etc/php-fpm.conf
pid=/usr/local/php7/var/run/php-fpm.pid

[root@qfedu.com etc]# cd /usr/local/php7/etc/php-fpm.d
[root@qfedu.com php-fpm.d]# cp www.conf.default nginx.conf
[root@qfedu.com php-fpm.d]# vim /usr/local/php7/etc/php-fpm.d/nginx.conf
user = nginx
group = nginx
listen = 127.0.0.1:9000
pm.max_children = 100
pm.start_servers = 20
pm.min_spare_servers = 5
pm.max_spare_servers = 35
```

## 8、php-fpm 进程池配置

- php-fpm 进程池开启进程有两种方式，
  - 静态 static，直接开启指定数量的php-fpm进程，不再增加或者减少；
  - 动态 dynamic，开始时开启一定数量的php-fpm进程，当请求量变大时，动态的增加php-fpm进程数到上限，当空闲时自动释放空闲的进程数到一个下限。
- 两种不同的执行方式，可以根据服务器的实际需求来进行调整。
- 用到的参数，分别是pm、pm.max\_children、pm.start\_servers、pm.min\_spare\_servers和pm.max\_spare\_servers。
- pm 表示使用那种方式，有两个值可以选择，就是static（静态）或者dynamic（动态）。
- 下面4个参数的意思分别为：
  - pm.max\_children：静态方式下开启的php-fpm进程数量，在动态方式下他限定php-fpm的最大进程数（这里要注意pm.max\_spare\_servers的值只能小于等 pm.max\_children）
  - pm.start\_servers：动态方式下的起始php-fpm进程数量。
  - pm.min\_spare\_servers：动态方式空闲状态下的最小php-fpm进程数量。
  - pm.max\_spare\_servers：动态方式空闲状态下的最大php-fpm进程数量。
- 如果 pm 设置为 static，那么其实只有pm.max\_children这个参数生效。系统会开启参数设置数量的 php-fpm 进程。
- 如果 pm 设置为 dynamic，4个参数都生效。系统会在php-fpm运行开始时启动 pm.start\_servers 个 php-fpm 进程，然后根据系统的需求动态在pm.min\_spare\_servers和pm.max\_spare\_servers 之间调整php-fpm进程数。

## 9、启动 php-fpm

```
[root@qfedu.com ~]# /usr/local/php7/sbin/php-fpm
```

## 10、检查 php-fpm 是否成功启动

```
[root@qfedu.com ~]# ps aux | grep php-fpm
```

若看到相关进程，则证明启动成功。查询进程时，进程是以 nginx 用户身份执行的

## 11、配置 php-fpm 系统环境变量

```
[root@qfedu.com ~]# vim /etc/profile
export PHP_HOME=/usr/local/php7
export PATH=$PATH:$PHP_HOME/bin:$PHP_HOME/sbin
```

## 12、重载环境变量

```
[root@qfedu.com ~]# source /etc/profile
```

- 使用 echo \$PATH 命令查看环境变量中是否已经加入了相关的路径

## 13、配置 php-fpm 开机自启动

```
[root@qfedu.com ~]# vim /lib/systemd/system/php-fpm.service
[Unit]
Description=php-fpm
After=network.target
[Service]
Type=forking
ExecStart=/usr/local/php7/sbin/php-fpm
ExecStop=/bin/pkill -9 php-fpm
PrivateTmp=true
[Install]
WantedBy=multi-user.target
```

## 14、php-fpm.service 文件说明

[Unit]:服务的说明  
Description:描述服务  
After:描述服务类别  
[Service]服务运行参数的设置  
Type=forking是后台运行的形式  
ExecStart为服务的具体运行命令  
ExecReload为重启命令  
ExecStop为停止命令  
PrivateTmp=True表示给服务分配独立的临时空间  
注意: [Service]的启动、重启、停止命令全部要求使用绝对路径  
[Install]运行级别下服务安装的相关设置，可设置为多用户，即系统运行级别为3

## 15、重载 systemctl 配置

```
[root@qfedu.com ~]# systemctl daemon-reload
```

## 16、停止 php-fpm

```
[root@qfedu.com ~]# pkill php-fpm
```

## 17、用 systemctl 启动 php-fpm

```
[root@qfedu.com ~]# systemctl start php-fpm.service
```

## 18、设置 php-fpm 开机启动

```
[root@qfedu.com ~]# systemctl enable php-fpm.service
```

## 19、php-fpm 管理命令

|                                                                    |              |
|--------------------------------------------------------------------|--------------|
| <pre>[root@qfedu.com ~]# systemctl stop php-fpm.service</pre>      | # 停止服务       |
| <pre>[root@qfedu.com ~]# systemctl restart php-fpm.service</pre>   | # 重新启动服务     |
| <pre>[root@qfedu.com ~]# systemctl status php-fpm.service</pre>    | # 查看服务当前状态   |
| <pre>[root@qfedu.com ~]# systemctl disable php-fpm.service</pre>   | # 停止开机自启动    |
| <pre>[root@qfedu.com ~]# systemctl list-units --type=service</pre> | # 查看所有已启动的服务 |

# 四、Nginx 配置支持PHP

## 1、添加 Nginx 配置

```
[root@qfedu.com php-fpm.d]# cd /etc/nginx
[root@qfedu.com conf]# vim nginx.conf
#配置如下
server
{
 listen 80;
 server_name localhost;
 index index.html index.htm index.php;
 root /usr/local/nginx/html;
 location ~ \.php$
 {
 include fastcgi_params;
 fastcgi_pass 127.0.0.1:9000;
 fastcgi_index index.php;
 fastcgi_param SCRIPT_FILENAME /usr/local/nginx/html$fastcgi_script_name;
 }
}
```

## 2、添加 php 探测文件

```
[root@qfedu.com ~]# cd /usr/local/nginx/html/
[root@qfedu.com html]# vim index.php
<?php
phpinfo();
?>
```

### 3、验证 Nginx 关联 php-fpm

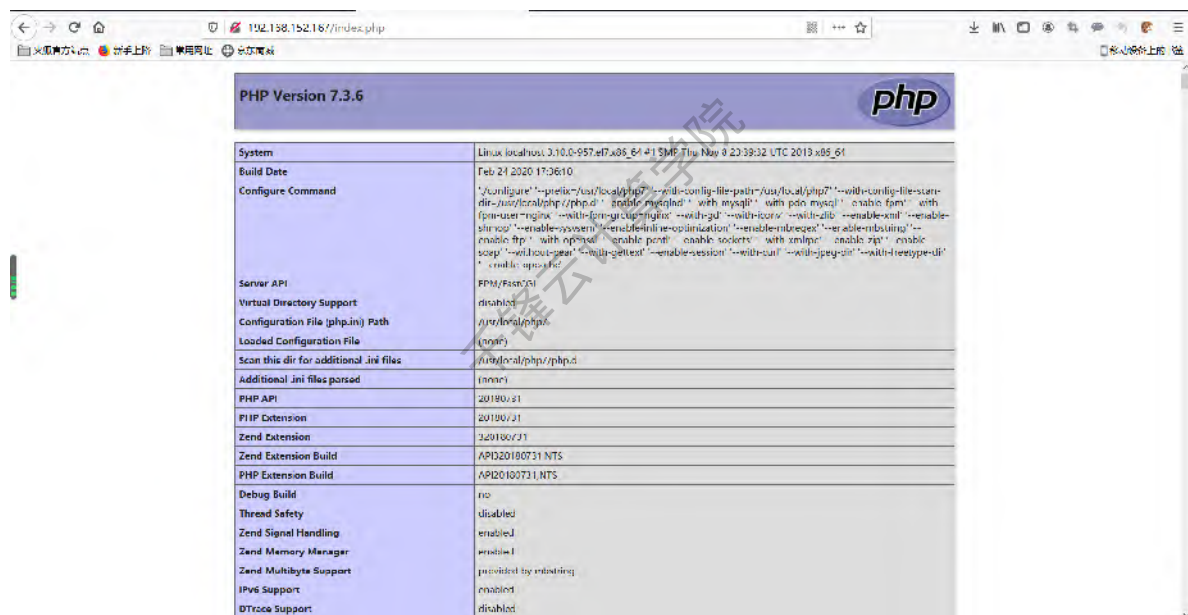
## 1、重启 php-fpm

```
[root@qfedu.com html]# systemctl restart php-fpm.service
[root@localhost ~]# ps -ef|grep php-fpm
root 69652 1 0 14:02 ? 00:00:00 php-fpm: master process
(/usr/local/php7/etc/php-fpm.conf)
nginx 69653 69652 0 14:02 ? 00:00:00 php-fpm: pool www
nginx 69654 69652 0 14:02 ? 00:00:00 php-fpm: pool www
nginx 69655 69652 0 14:02 ? 00:00:00 php-fpm: pool www
nginx 69656 69652 0 14:02 ? 00:00:00 php-fpm: pool www
nginx 69657 69652 0 14:02 ? 00:00:00 php-fpm: pool www
```

## 2、重载 Nginx 配置

```
[root@qfedu.com html]# systemctl reload nginx
[root@qfedu.com html]# systemctl status nginx
```

### 3、访问验证



## 五、部署企业级项目管理系统（蝉道）

## 1、禅道下载

- 进入禅道官网<https://www.zentao.net>，选择适用的版本进行安装，我这里下载的是“开源版11.6”

# 禅道是有保障的项目管理软件

- 禅道开发团队从04年从事开源，坚持到现在
- 已为数以万计的公司提供即时有力的支持
- 版本迭代快速，为您提供更好的功能

专业版8.5

企业版3.5.alpha

集团版1.1

开源版11.6

客户端

千锋云计算学院



## 二、下载地址

### 1、禅道项目管理软件源码下载

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

### 2、集成运行环境下载：切勿下载下面的软件进行升级，仅适用于新安装

#### Windows—键安装包（适用于windows 64位）

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

#### Windows—键安装包（适用于windows 32位）

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

#### Windows—键安装包（未加安全设置）

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

#### Linux—键安装包64位（适用于Ubuntu17+， centos7.4+）

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

#### Linux—键安装包32位（适用于Ubuntu17+， centos7.4+）

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

#### 低版本Linux—键安装包64位（适用于ubuntu16及以下版本、centos7.3及以下版本）

官方下载源、 sourceforge 下载

#### 低版本Linux—键安装包32位（适用于ubuntu16及以下版本、centos7.3及以下版本）

官方下载源、 sourceforge 下载

注：Linux—键安装包必须直接解压到/opt目录下。

### 3、DEB包下载：可以通过dpkg包管理器在Ubuntu和Debian系统下安装

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

### 4、RPM包下载：可以通过rpm包管理器在Centos系统下安装

中文版： 官方下载源、 sourceforge 下载

国际版： 官方下载源、 sourceforge 下载

### 5、最新版禅道客户端下载链接

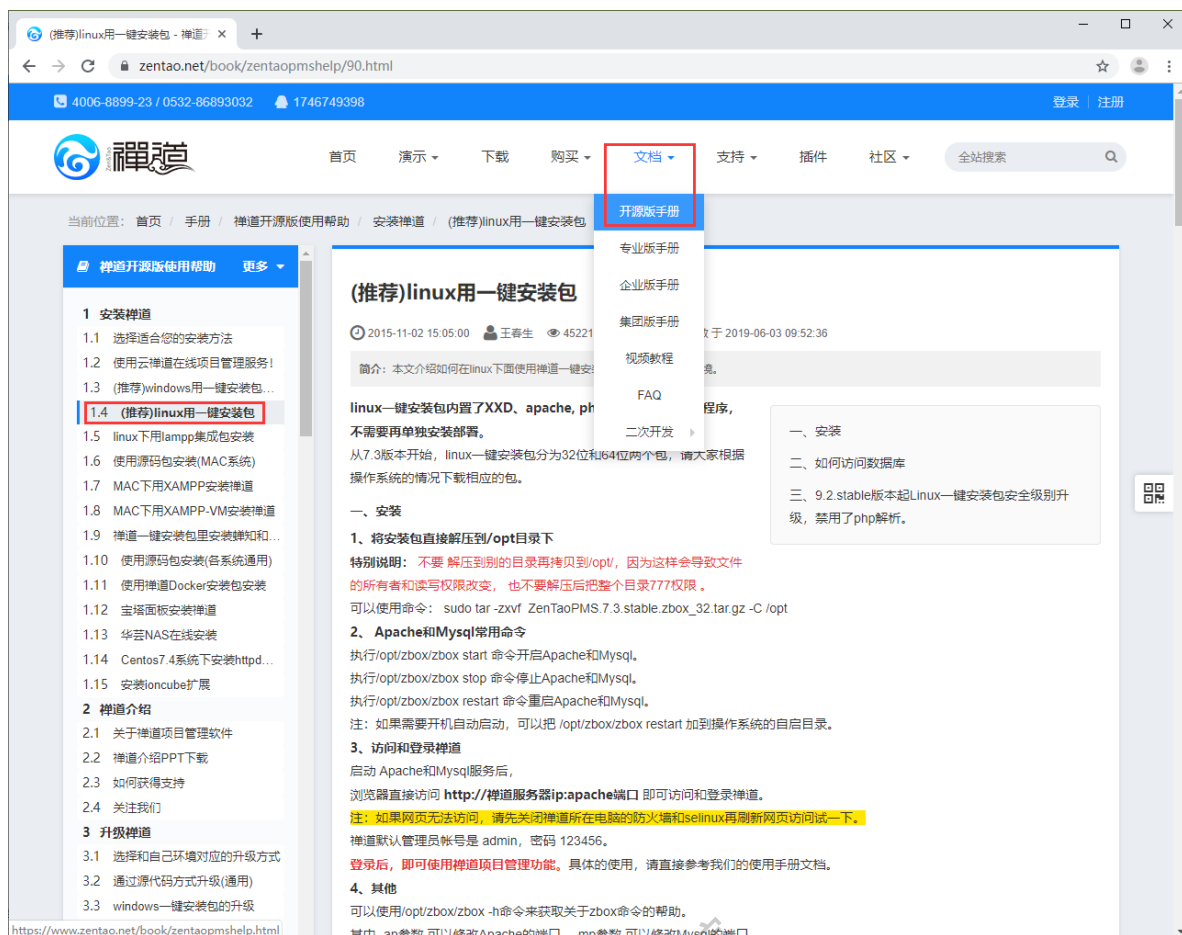
Windows64位、 Windows32位、 Linux64位、 Linux32位、 Mac

### 6、最新版禅道客户端服务器下载链接

Windows64位、 Windows32位、 Linux64位、 Linux32位、 Mac

- 官方开源版使用手册：<https://www.zentao.net/book/zentaopmshelp/40.html>

我下载的是这个



- 注：Linux一键安装包必须直接解压到/opt目录下。

## 2、禅道安装

- 将安装包直接解压到/opt目录下，不要解压到别的目录再拷贝到/opt/，因为这样会导致文件的所有者和读写权限改变，也不要解压后把整个目录777权限。
- 可以使用命令：

```
[root@qfedu.com ~]# tar -zxvf ZenTaoPMS.11.6.stable.zbox_64.tar.gz -C /opt/
```

## 3、管理命令

```
[root@qfedu.com ~]# /opt/zbox/zbox start # 命令开启Apache和MySQL
[root@qfedu.com ~]# /opt/zbox/zbox stop # 命令停止Apache和MySQL
[root@qfedu.com ~]# /opt/zbox/zbox restart # 命令重启Apache和MySQL
```

- 我这里因为服务器上已经安装了Apache和MySQL，避免端口冲突，所以启动时用命令 /opt/zbox/zbox start -ap 8888 -mp 3307 指定了Apache和MySQL的端口，如下图：

```
[root@localhost opt]# /opt/zbox/zbox start -ap 8888 -mp 3307
Start Apache success
Start Mysql success
```

- 可以使用/opt/zbox/zbox -h命令来获取关于zbox命令的帮助

```
[root@localhost opt]# /opt/zbox/zbox -h
Usage: zbox.php {start|stop|restart|status}

Options:
 -h --help Show help.
 -ap --aport Apache port, default 80.
 -mp --mport Mysql port, default 3306.
[root@localhost opt]#
```

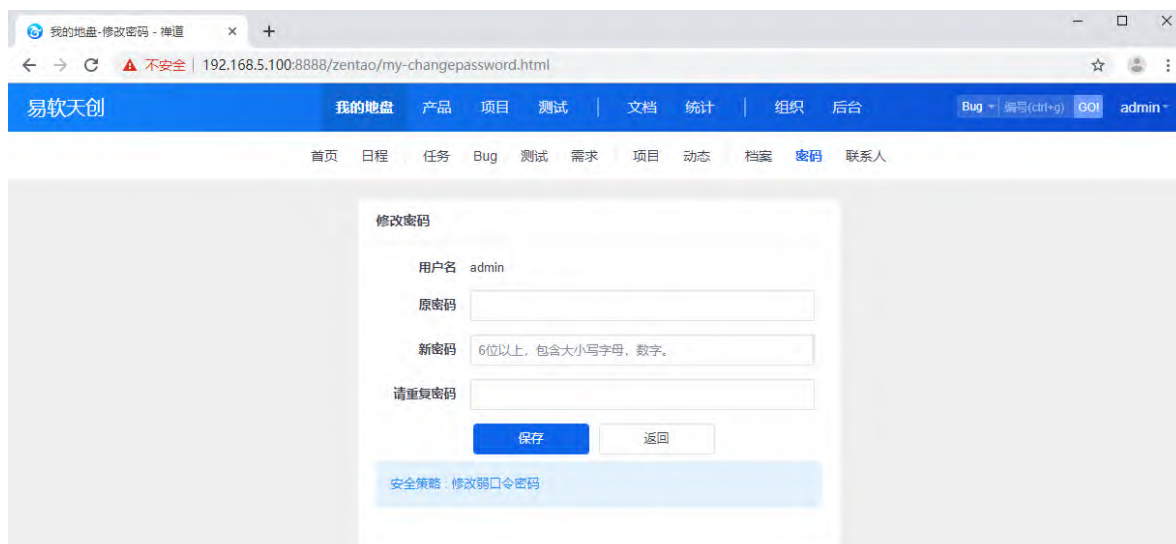
- 其中-ap参数可以修改Apache的端口，-mp参数可以修改MySQL的端口
- 顺便将Apache端口加入防火墙

```
[root@qfedu.com ~]# firewall-cmd --zone=public --add-port=8888/tcp --permanent
[root@qfedu.com ~]# firewall-cmd --reload
```

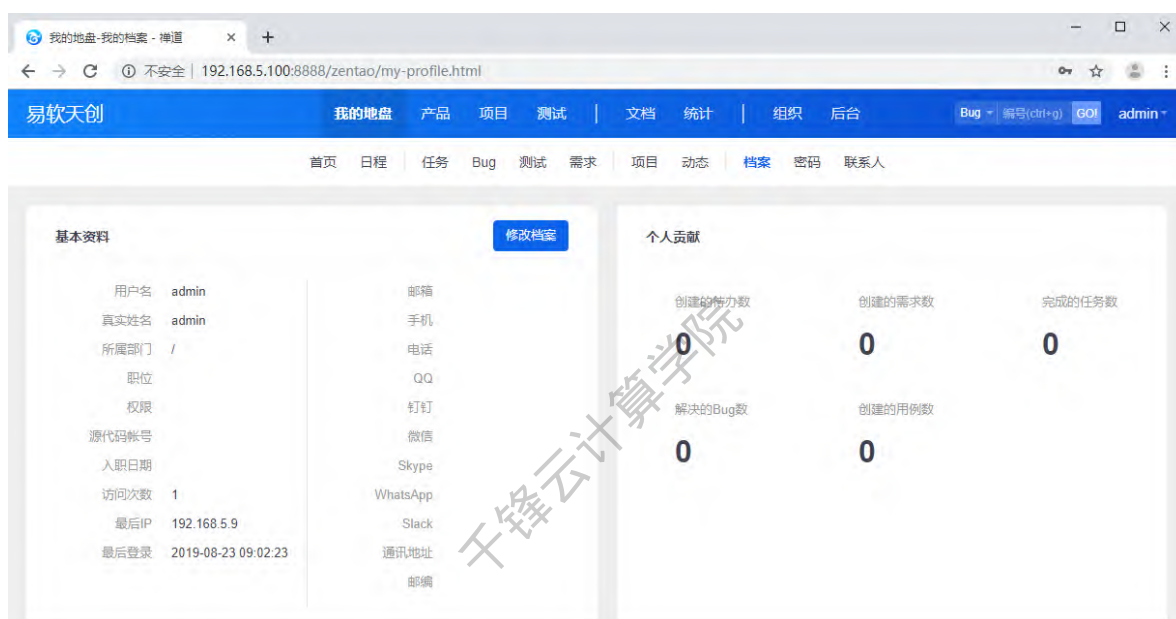
### 3、浏览器访问

- 浏览器访问 <http://ip:apache>端口，点击“开源版”，进入到禅道登录界面，输入默认用户名admin，密码123456，进入后修改密码。





- 进入后的首页如下图：



- 到此处，说明禅道已经搭建完成，可以进行一系列新建产品和项目，包括分配任务。

## 4、访问禅道数据库

### 1、网页登录数据库

- 禅道数据库管理用的是adminer，但是为了安全，访问 adminer 的时候需要身份验证，需要运行 `/opt/zbox/auth/adduser.sh` 来添加用户(先 `cd /opt/zbox/auth/` 然后执行 `./adduser.sh`)。

```
[root@localhost opt]# cd /opt/zbox/auth/
[root@localhost auth]# ./adduser.sh
This tool is used to add user to access adminer
Account: root
Password: Adding password for user root
[root@localhost auth]#
```

### 2、设置登录账号密码

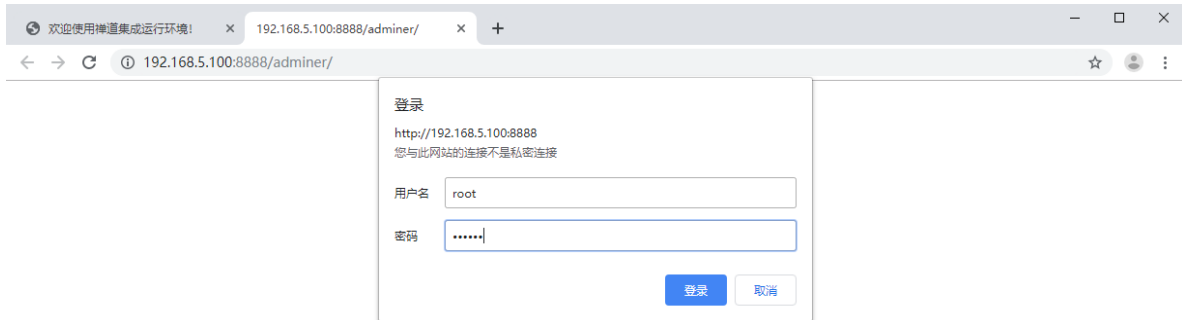
- 设置的 Account 是 root，Password 是 123456，这里添加的用户名和密码是自定义的。

### 3、登录数据库

- 网页访问 `http://禅道服务的 ip:apache端口`，点击“数据库管理”

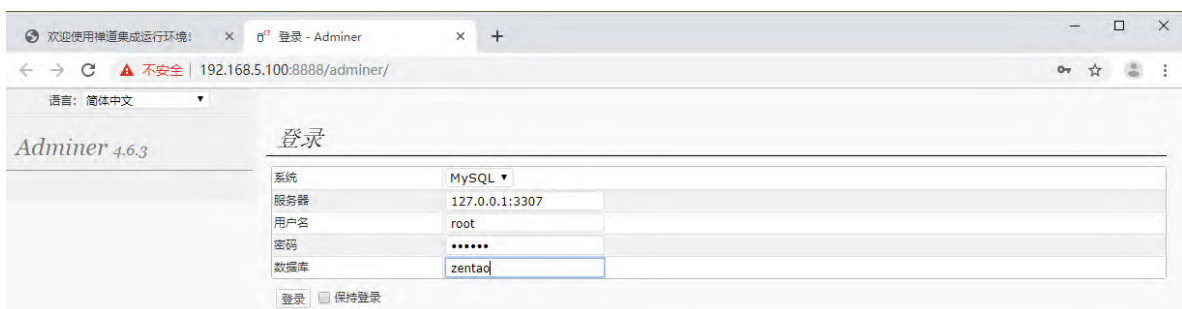


- 输入刚运行 addusers.sh 设置的用户名和密码。

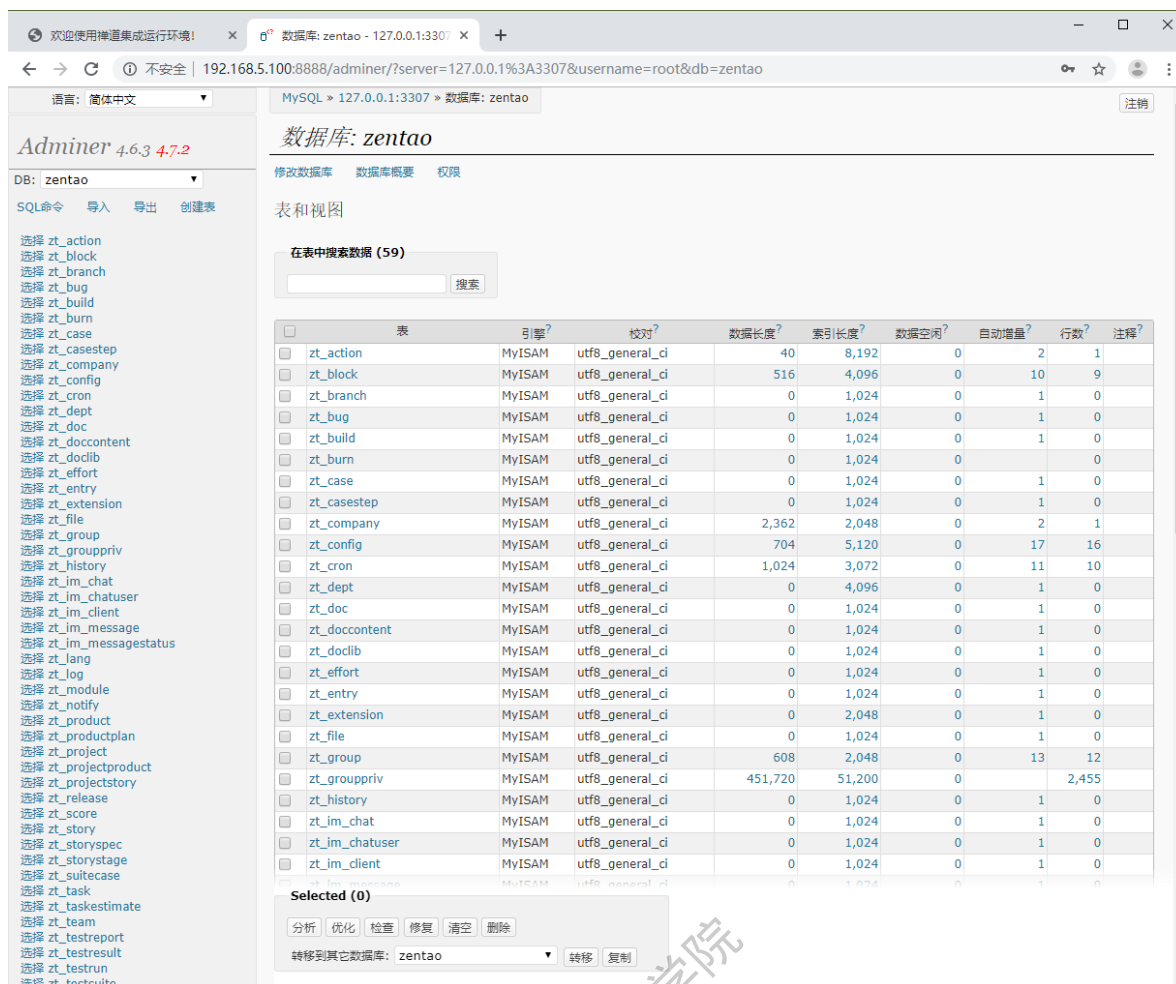


#### 4、连接数据库

- 系 统：默认选择MySQL
- 服务器：127.0.0.1:mysql端口，前面开启服务时，mysql端口启用的是3307。那么就填写127.0.0.1:3307（服务器以实际部署的ip为准）
- 用户名：root
- 密 码：123456
- 数据库：zentao



- 点击“登录”后，进入数据库管理界面，如下图：



## 5、命令行连接数据库

- 登录数据库：

```
[root@qfedu.com ~]# /opt/zbox/bin/mysql -u root -P mysql端口 -p （比如：/opt/zbox/bin/mysql -u root -P 3307 -p）
```

- 导入数据库：

```
[root@qfedu.com ~]# /opt/zbox/bin/mysql -u root -P mysql端口 -p 要导入的库名 < xxxx.sql （比如：/opt/zbox/bin/mysql -u root -P 3307 -p zentao < zentao.sql）
```

## 3、设置禅道开机自动启动

- CentOS7 系统服务脚本目录 /usr/lib/systemd/
- 有系统（system）和用户（user）之分，如需要开机没有登陆情况下就能运行的程序，存在系统服务（system）里，即 lib/systemd/system/，反之，用户登录后才能运行的程序，存在用户（user）里，服务以.service结尾。
- 下面以zentao开机运行为例：

### 1、建立服务文件

```
[root@qfedu.com ~]# vim /lib/systemd/system/zentao.service
[Unit]
Description=zentao
After=network.target
```



```
[Service]
Type=forking
ExecStart=/opt/zbox/zbox start
ExecReload=/opt/zbox/zbox restart
ExecStop=/opt/zbox/zbox stop
PrivateTmp=true
```

```
[Install]
WantedBy=multi-user.target
```

[Unit]: 服务的说明  
Description:描述服务  
After:描述服务类别

[Service]服务运行参数的设置  
Type=forking是后台运行的形式  
ExecStart为服务的具体运行命令  
ExecReload为重启命令  
ExecStop为停止命令  
PrivateTmp=True表示给服务分配独立的临时空间

注意: [Service]的启动、重启、停止命令全部要求使用绝对路径

[Install]服务安装的相关设置, 可设置为多用户

## 2、设置开机自启动

```
[root@qfedu.com ~]# systemctl daemon-reload
[root@qfedu.com ~]# systemctl enable zentao
```

## 3、其他命令

```
启动zentao服务
[root@qfedu.com ~]# systemctl start zentao

设置开机自启动
[root@qfedu.com ~]# systemctl enable zentao

停止开机自启动
[root@qfedu.com ~]# systemctl disable zentao

查看服务当前状态
[root@qfedu.com ~]# systemctl status zentao

重新启动服务
[root@qfedu.com ~]# systemctl restart zentao

查看所有已启动的服务
[root@qfedu.com ~]# systemctl list-units --type=service | grep running
```

## 4、重启服务器测试

```
[root@qfedu.com ~]# ps aux | grep zbox
```

## 六、使用禅道进行项目管理

### 1、建立部门结构

简介：禅道安装成功之后，管理员的第一件要做的事情就是设置部门结构。

禅道安装成功之后，管理员的第一件要做的事情就是设置部门结构。

#### 1、维护部门结构

1. 以管理员身份登录；
2. 进入组织--用户视图；
3. 点击维护部门；(或者直接点击二级导航里的“部门”。)
4. 在部门维护页面，维护公司的组织结构即可。



#### 2、维护子部门

点击部门名称，即可添加该部门的下级部门（子部门）。





下级部门添加成功后，即可在部门机构里查看到。

## 2、添加一个用户帐号

**简介：**部门创建之后，下一步的操作就是往系统中添加用户。

部门创建之后，下一步的操作就是往系统中添加用户。步骤如下：

1. 进组织视图
2. 选择用户列表
3. 然后选择“添加用户”，即可进入添加用户页面。
4. 用户添加完之后，即可将其关联到某一个分组中。



注意点：

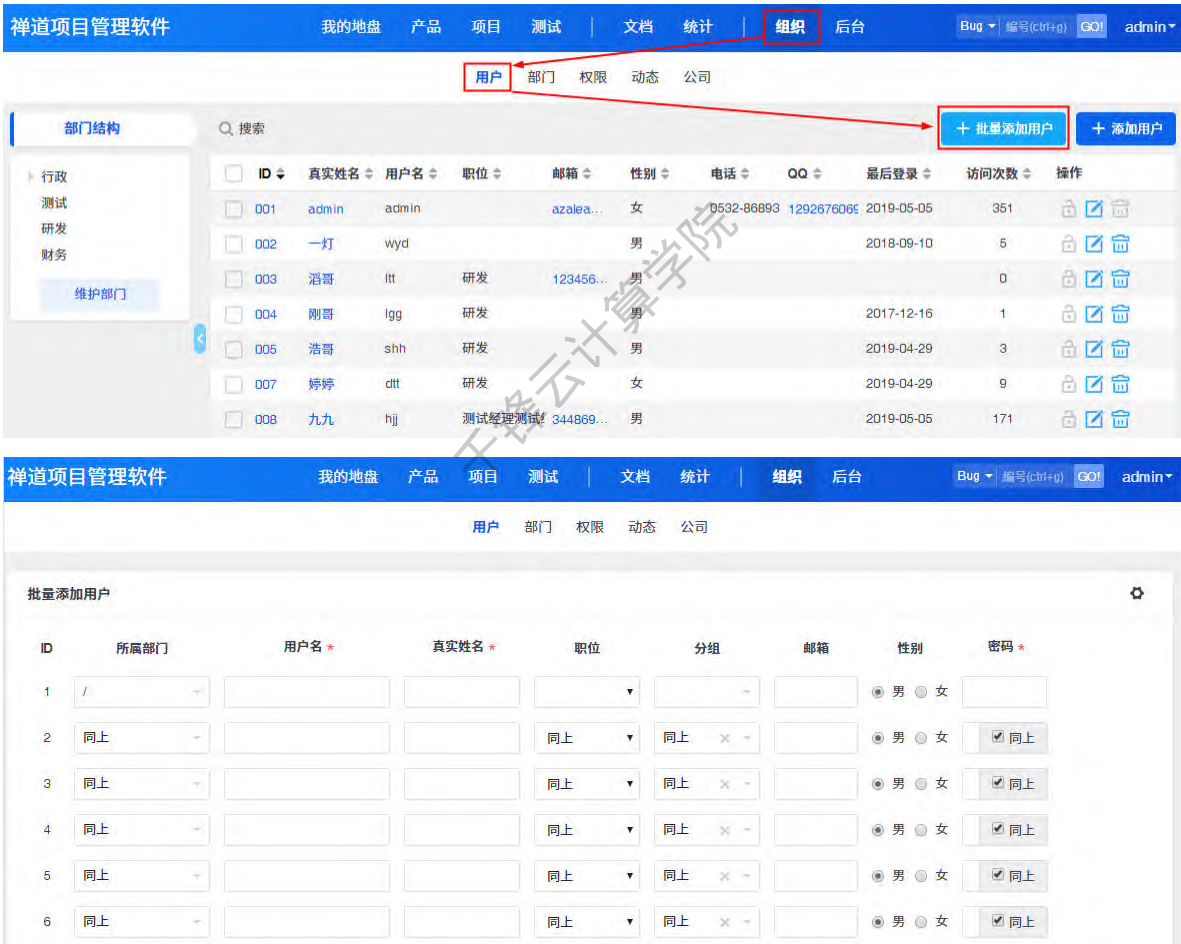
- 1. 从4.0版本开始增加了职位字段，在添加帐号的时候可以选择对应的职位。职位会影响到指派列表的顺序，比如创建bug的时候，默认会把研发职位的同学放在前面。职位还会影响到我的地盘里面内容的排列顺序。比如产品经理角色的人登录之后，我的地盘首先会显示我的需求，而研发的同学登录之后，会看到我的任务。
- 2. 用户的权限都是通过分组来获得的，因此为用户指定了一个职位之后，还需要将其关联到一个分组中。
- 3. 其中源代码提交帐号是subversion或者其他源代码管理系统中对应的用户，如果没有启用subversion集成功能，可以留空。

### 3、批量维护帐号

简介：禅道可以批量添加和编辑用户。

从4.0版本开始，我们提供了批量添加帐号的功能，可以很方便的批量创建帐号。

- 1. 使用管理员登录禅道系统，进入“组织”页面。
- 2. 选择“用户”，然后选择右侧的“批量添加用户”



除了批量添加用户，还可以在用户列表页面选择用户，进行批量编辑。



## 4、设置分组，建立权限体系

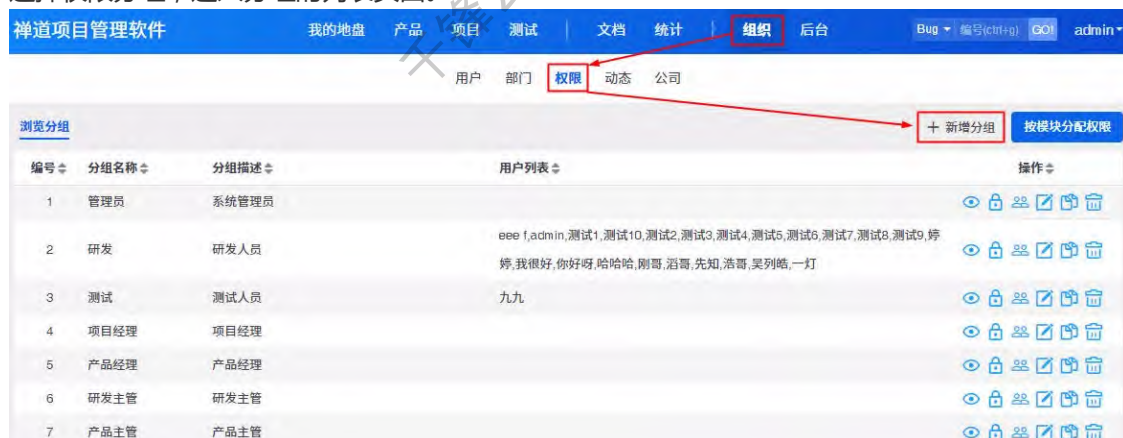
简介：用户创建之后，下一步的操作就是创建分组，设置权限。

在禅道中，用户权限都是通过分组来获得的。所以在完成部门结构划分之后，就应该建立用户分组，并为其分配权限。有的朋友可能会问，用户分组和部门结构有什么区别？我们来解释下这个问题。

部门结构是公司从组织角度来讲的一个划分，它决定了公司内部人员的上下级汇报关系。而禅道里面的用户分组则主要用来区分用户权限。二者之间并没有必然的关系。比如用户A属于产品部，用户B属于研发部，但他们都有提交bug的权限。

### 1、创建分组

1. 使用管理员登录禅道，进入组织视图。
2. 选择权限分组，进入分组的列表页面。



3. 点击新增分组，即可创建分组。
4. 在这个分组列表页面，还可以对某一个分组进行权限的维护、成员维护、或者复制。

### 2、维护权限

1. 管理员身份登录
2. 进入组织视图
3. 点击“权限分组”，进入权限分组列表页面。
4. 选择某一个分组，点击“权限维护”，即可维护该分组的权限。

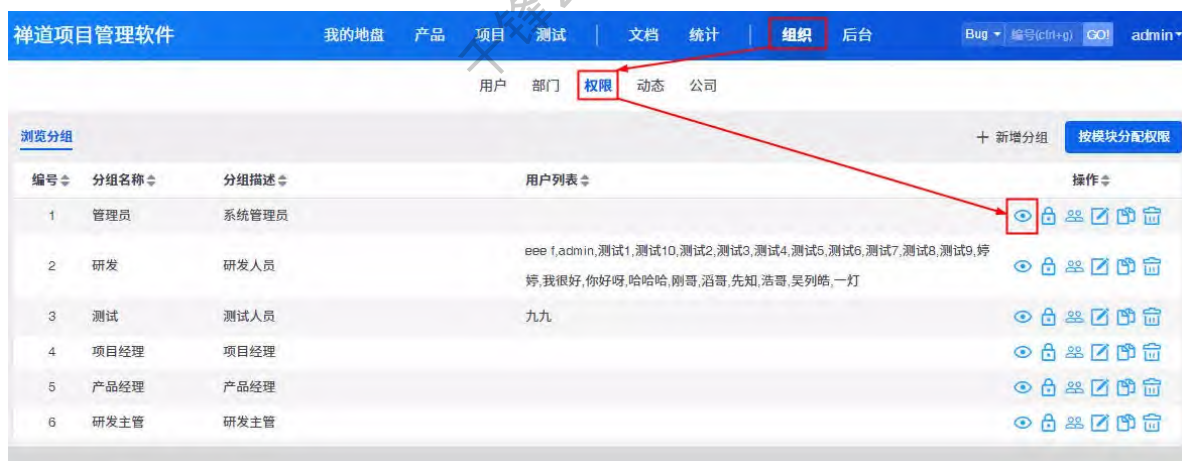




5.进入权限列表页面，点击某一个模块名后面的复选框，可以全选该模块下面的所有权限，或者全部取消选择。还可以查看某一个版本新增的权限列表，即下图的红框位置。



6.禅道7.2.stable版本开始，还提供视图维护权限。可以设置某个分组仅能查看某个产品或者项目等。



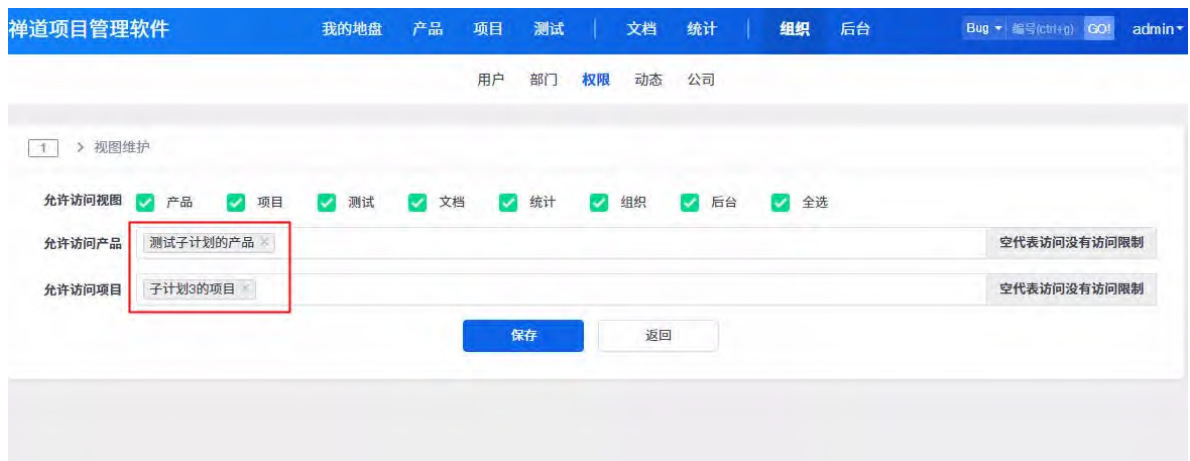
7.限制访问框里填写了相关内容，就是表示该分组仅有权访问限制访问框里的内容。

以下图为例：

**允许访问视图：**全选的话，那么禅道导航栏里的主菜单（产品、项目、测试、文档、统计、组织、后台）都显示。不勾选，导航栏就不显示，也无权限访问。

**允许访问产品：**下图允许访问产品为 **禅道项目管理软件**，那么该权限分组的用户只能访问 **禅道项目管理软件** 这个产品，别的产品都没权限访问。

**允许访问项目：**与允许访问产品的设置是一样的。




\8. 禅道9.6.2.版本，新增了 **受限操作权限分组**。

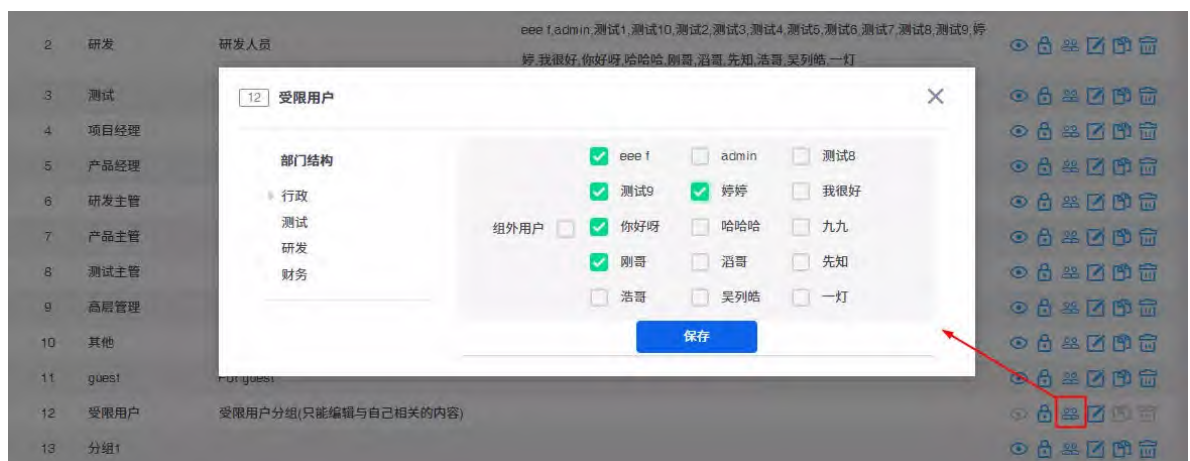
**组织--权限--受限用户分组**，维护分组成员时，属于该分组的用户在禅道里 **\*只能编辑与自己相关的\*** **需求、任务、bug等\***，不能新增 需求、任务、bug等。

注：与自己相关的内容，包含指派给、已完成、已取消、已关闭、最后编辑，不包含抄送给的内容。

如果该用户之前不是受限用户，现在是受限用户。那么之前由他创建需求，任务，bug等，他都还有相关的操作权限。

比较适用于公司新来员工的权限设置，怕其不熟悉公司业务而导致误操作。

| 编号 | 分组名称  | 分组描述                 | 用户列表                             | 操作                                                                                                                                                                                                                                                                                                                                                      |
|----|-------|----------------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | 管理员   | 系统管理员                |                                  |             |
| 2  | 研发    | 研发人员                 | 婷,我很好,你好好,哈哈哈,刚哥,滔哥,先知,浩哥,吴列皓,一灯 |         |
| 3  | 测试    | 测试人员                 | 九九                               |     |
| 4  | 项目经理  | 项目经理                 |                                  |     |
| 5  | 产品经理  | 产品经理                 |                                  |     |
| 6  | 研发主管  | 研发主管                 |                                  |     |
| 7  | 产品主管  | 产品主管                 |                                  |     |
| 8  | 测试主管  | 测试主管                 |                                  |     |
| 9  | 高层管理  | 高层管理                 |                                  |     |
| 10 | 其他    | 其他                   |                                  |     |
| 11 | guest | For guest            |                                  |     |
| 12 | 受限用户  | 受限用户分组(只能编辑与自己相关的内容) |                                  |     |



此外在 **项目--团队--团队管理**里，也增加了针对某个项目设置受限用户的功能。

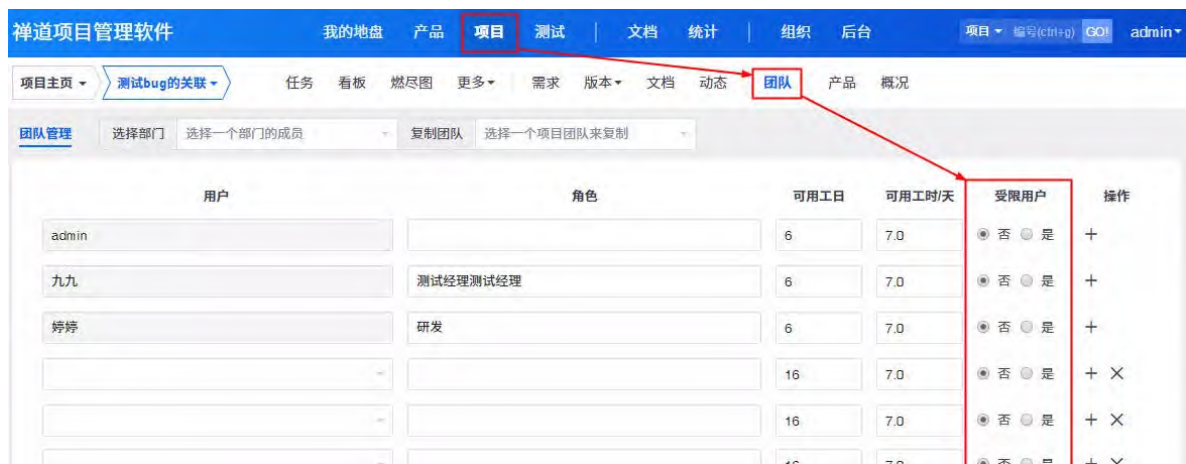
如果某个团队成员设置为该项目的受限用户，那么该团队成员只能编辑该项目里与自己相关的任务、需求、bug等。

注意：

1、组织--权限-- **\*受限用户分组\***，设置的受限操作，是针对禅道整个系统的使用受限。

2、项目--团队--团队管理里，设置的受限用户，是只针对某一个项目的受限操作，其他项目不受影响。

3、\*如果该用户之前不是受限用户，现在是受限用户。那么之前由他创建的与该项目相关的需求，任务，bug等，他都还有相关的操作权限。\*



权限维护的注意事项：

- 1、一个用户在多个权限分组里，其在禅道里的权限取的是各个权限分组里权限的合集。
- 2、要访问到一个Bug，必须同时拥有Bug所在产品，项目的访问权限和Bug详情的权限。
- 3、产品/项目还可以通过访问控制来设置查看权限。产品/项目--概况里，编辑访问控制：默认设置，私有产品/项目，自定义白名单来调整产品/项目的查看权限。
- 4、禅道里权限分配比较灵活，可以根据实际需要调整。
- 5、没必要过于的纠结权限的分配，禅道里每个操作都会记录，在详情页的历史记录可以查看到。

### 3、维护成员

1. 管理员身份登录
2. 进入组织视图中的权限分组。
3. 点击“成员维护”，进入用户维护页面。



## 5、使用禅道来进行项目任务管理

简介：禅道简使用方式就是单纯的项目任务管理，创建项目，分解任务。



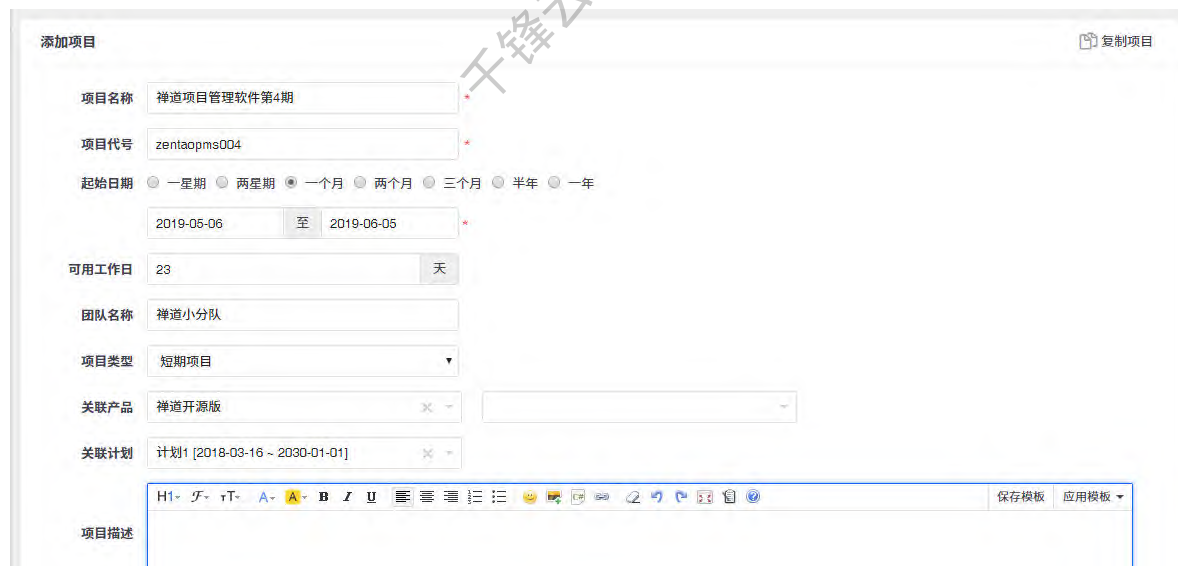
也许您刚刚看到禅道的界面，心里说，哇，这么多功能。是不是有些复杂了？我们想说的是，禅道的定位不是那种简单的任务管理软件，而是专业的协同管理软件。研发类的项目管理本身具有其复杂性，所以禅道提供的都是必备的功能。但这并不意味着您必须按照禅道的流程来使用，完全可以按照自己的实际情况来使用禅道。下面让我们来看下使用禅道的简单方式。

## 1、创建项目

添加新项目的入口有多个，进入项目视图，可以在左侧1.5级导航的下拉菜单中直接点击“添加项目”。或者在项目主页的项目总览区块中点击“添加项目”。



在项目添加的页面中，设置项目名称、代号、起止时间、可用工作日、团队名称、和项目描述等字段。其中关联产品是可以为空，也可以多选。



## 2、设置团队

点击保存按钮，会提示项目创建成功，然后可以选择设置团队。或者从项目视图中的团队菜单，也可以进行项目的团队管理。



在维护项目团队的时候，需要选择都是哪些用户可以参与到这个项目中，同时需要设置这个用户在本项目中的角色（角色可以随便设置，比如风清扬，冬瓜一号等）。可用工作日和可用工时每天需要仔细设置。通常来讲，一个人不可能每天8小时投入，也不可能一星期七天连续投入。设置完毕之后，系统会自动计算这个项目总得可用工时。

### 3、分解任务

设置了团队之后，下一步操作就是创建任务。





- 在创建任务的时候，指派给是从项目团队成员中读取。
- 姓名列表中的首字母可以用来快速筛选用户。
- 任务的优先级、预计工时（单位小时）都需要进行设置。
- 如果需要设置任务必须在某一个时间点截止，可以设置截止日期。
- 可以上传附件。

## 4、管理任务

任务分解完毕之后，每个人就非常清楚自己做什么事情。所以项目启动之后，对于项目团队的成员来讲，他要做的事情就是更新任务的进度和状态。

**任务的列表：**在任务的列表页面，可以看到系统中所有的任务列表，可以通过各种标签方便的进行筛选。点击某一个任务的名称进入详情页面。

**任务的详情页面：**在任务的详情页面可以看到任务的详细信息，包括历次的修改记录等信息。同时也给出了各种操作的按钮。

**开始任务：**开始某一个任务的时候，可以设置已经消耗的时间和预计剩余的时间。单位都是工时。

**更新任务工时：**点击操作栏里的“工时”按钮，通过更新工时消耗，来管理任务执行进度。

**完成任务：**完成任务的时候，需要设置下已经消耗的时间。

**关闭任务：**任务完成之后，点击操作栏里的“关闭”按钮，将该任务关闭，这个任务就结束了。

上述就是使用禅道来进行简单的项目和任务管理，怎么样？还是很简单的吧。下面让我们来看下只使用禅道来进行bug的跟踪管理。

## 6、只使用禅道来做bug管理

**简介：**本文讲述如何使用禅道来进行bug的跟踪和管理。

上篇文章讲述了禅道的简使用方法之一，只使用禅道来进行项目任务管理。禅道的测试功能也可以独立出来单独使用。这种方式很适合于测试团队使用。禅道里面的bug基本流程是：**测试人员提出bug -> 开发人员解决bug -> 测试人员验证关闭**。

下面我们来演示下具体的使用方法。

## 1、创建产品

使用 [bug管理](#) 功能之前，需要先创建产品。禅道里面设计的理念是bug主要附属在产品概念下面的，后面我们会详细讲述产品和项目之间的关系。

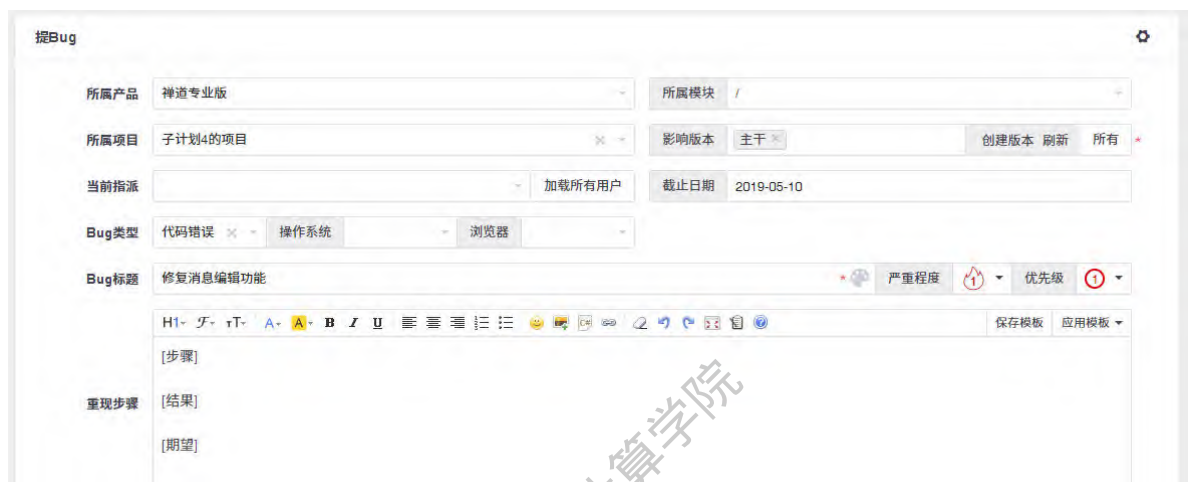
添加产品的入口有多个，可以在产品视图中的1.5级导航下拉菜单中直接点击“添加产品”。也可以在所有产品页面点击右侧的“添加产品”按钮。



新增产品的时候，需要设置产品的名称、代码，几个负责人信息。

## 2、提出bug

有了产品之后，我们就可以来创建bug了。



- 在创建bug的时候，必填的字段是：**影响版本，bug标题，所属模块。**
- 所属项目，相关产品，需求可以忽略。
- 创建bug的时候，可以直接指派给某一个人员去处理。如果不清楚的话，可以保留为空。

### 3、处理bug

当一个bug指派给某一位研发人员之后，他可以来确认、解决这个bug。在对bug进行处理之前，需要先找到需要自己处理的bug。禅道提供了各种各样的检索方式，比如指派给我，可以列出所有需要我处理的bug。



**确认bug：**确认该bug确实存在后，可以将其指派给某人，并指定bug类型、优先级、备注、抄送等。

**解决bug**：当bug修复解决后，点击解决，指定解决方案、日期、版本，并可将其再指派给测试人员。

**关闭bug**：当研发人员解决了bug之后，bug会重新指派到bug的创建者头上。这时候测试人员可以来验证这个bug是否已经修复。如果验证通过，则可以关闭该bug。（bug列表页和详情页中都有“关闭”按钮。）

**编辑bug**：对bug进行编辑操作。

**复制bug**：复制创建当前bug，在此基础上再做改动，避免重新创建的麻烦。

## 7、只使用禅道来进行产品管理

**简介**：对于产品经理来讲，禅道可以作为一个非常好用的需求管理工具。

前面两篇文章讲述的禅道的简流程，是分别站在研发团队、测试团队角度展开的。研发团队可以使用禅道来做基本的任务跟踪管理，测试团队可以使用禅道来进行缺陷的跟踪管理。那么如果您是一位产品经理，您肯定要问了，禅道能不能用来当作 [需求管理工具](#)呢？答案当然是Yes!

禅道系统中提供了完整的需求管理流程，这其中包括需求的创建，需求的变更，需求的评审等。下面让我们来看下。

### 1、创建产品。

创建需求的前提当然是要有产品，这和bug的处理是一样的。新增产品的时候，需要设置产品的名称、代码，几个负责人信息。



### 2、创建需求。

有了产品之后，就可以来创建需求了。



禅道项目管理软件

我的地盘 **产品** 项目 测试 文档 统计 组织 后台

需求 计划 发布 项目 文档 动态 路线图 模块 概况

产品主页 禅道专业版

所有模块

所有 未关闭 120 指给我 我创建 我评审 草稿 更多 自定义 搜索 报表 导出

+ 批量创建 + 提需求

| ID  | P | 需求名称 | 计划         | 创建    | 指派 | 预计 | 状态  | 阶段  | T | 操作 |
|-----|---|------|------------|-------|----|----|-----|-----|---|----|
| 223 | 0 | 9    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 222 | 0 | 8    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 221 | 0 | 7    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 220 | 0 | 6    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 219 | 0 | 5    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 218 | 0 | 4    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 217 | 0 | 3    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 216 | 0 | 2    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 215 | 0 | 1    |            | admin | 0  | 草稿 | 未开始 | 0   |   |    |
| 211 | 3 | 付费的  | pro1.0子... | 九九    | 婷婷 | 3  | 激活  | 已计划 | 0 |    |

### 需求的创建页面

提需求

所属产品 禅道专业版 所属模块 /

所属计划 需求来源 来源备注

由谁评审 W:一灯 ☐ 不需要评审 勾选不需要评审，需求状态直接为激活

需求名称 一句话简要表达需求的内容 优先级 3 预计 小时

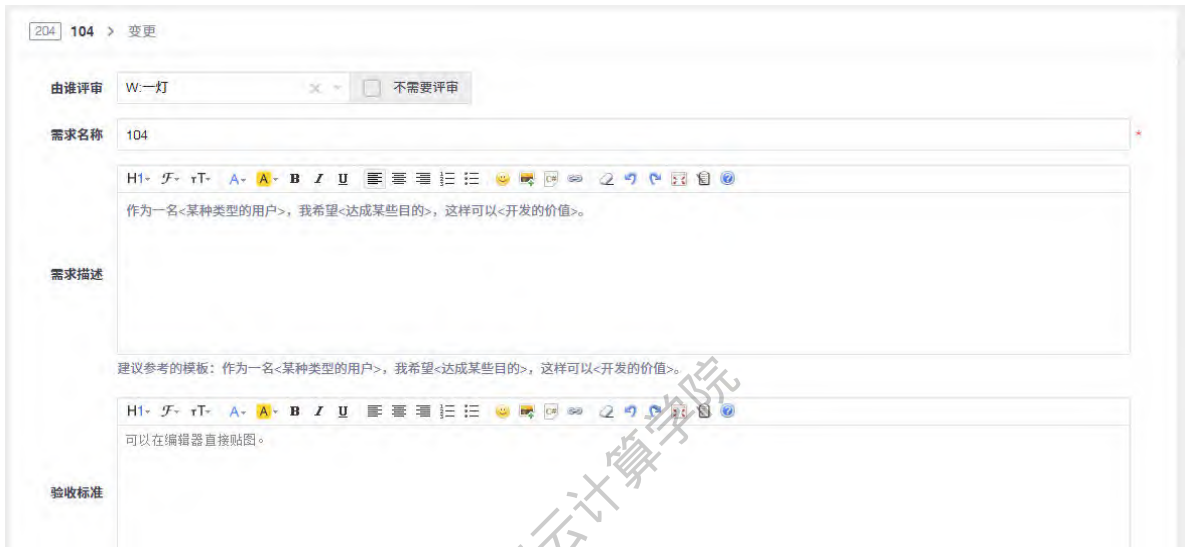
需求描述

验收标准

- 创建需求的时候，可以选择需求的来源。
- 需求的标题、描述和验证标准是需求的主要的三个元素，应该认真清晰的进行填写。
- 在创建需求的时候，可以指定需求的优先级、预计工时等字段。
- 创建需求的时候可以选择由谁来进行评审，这样创建的需求状态是 **草稿状态**。如果勾选了“不需要评审”，则是 **激活状态**。

### 3、变更需求。

禅道专门提供了需求的变更流程。凡是对需求 **标题、描述、验证标准和附件**的修改，都应该走变更流程。变更之后的需求状态为 **变更中**。



## 4、评审需求。

### 4.1 通过需求的详情页面查看变更前后的变化



### 4.2 评审需求，给出评审结果

44 文档添加上传下载功能 > 评审

评审时间 2019-05-06

评审结果 
 确认通过  
 撤销变更  
 有待明确

指派给

由谁评审

备注

可以在编辑器直接贴图。

影响的项目 影响的Bug 影响的用例

影响范围 然之第一期 A:admin H:九九 L:刚哥 S:浩哥

评审结果可以选择确认通过，撤销变更，有待明确或者拒绝。

## 8、个人使用禅道来做事务跟踪管理

简介：对于普通的个人用户来讲，禅道也可以用作事务管理工具。

禅道设计的目标是团队协作工具，但其实个人使用禅道也可以派上很大的用场。笔者在开发禅道过程中，从2009年10月开始，用禅道来管理禅道本身的项目管理，那时候禅道的开发团队只有笔者光杆司令一人，后来和朋友们聊起，戏称一个人的项目管理。)

下面让我们来展示下如何使用禅道来进行个人的 事务管理。

### 1、创建待办

禅道项目管理软件

我的地盘 产品 项目 测试 文档 统计 组织 后台

Bug 编号(ctrl+g) GO! admin

首页 日程 任务 Bug 测试 需求 项目 动态 档案 密码 积分 联系人

所有待办 238 本年 待定 未完 周期 2019-05-06 导出 + 批量添加 + 添加待办

| ID  | 日期         | 类型  | P      | 待办名称    | 开始    | 结束    | 状态  | 操作      |
|-----|------------|-----|--------|---------|-------|-------|-----|---------|
| 123 | 待定         | 自定义 |        | 天天天     | 06:00 | 06:00 | 未开始 | 🔖 ☑️ 🗑️ |
| 122 | 待定         | 自定义 |        | 项目1     | 14:00 | 14:30 | 未开始 | 🔖 ☑️ 🗑️ |
| 121 | 待定         | 自定义 |        | 点点滴滴    | 14:00 | 14:30 | 未开始 | 🔖 ☑️ 🗑️ |
| 184 | 待定         | 自定义 |        | ffffwww | 14:20 | 14:50 | 未开始 | 🔖 ☑️ 🗑️ |
| 183 | 待定         | 自定义 |        | ffff    | 14:20 | 14:50 | 未开始 | 🔖 ☑️ 🗑️ |
| 182 | 待定         | 自定义 |        | ffff    | 14:20 | 14:50 | 未开始 | 🔖 ☑️ 🗑️ |
| 181 | 待定         | 自定义 |        | ee      | 14:20 | 14:50 | 未开始 | 🔖 ☑️ 🗑️ |
| 129 | 待定         | 自定义 |        | 222     | 14:30 | 15:00 | 未开始 | 🔖 ☑️ 🗑️ |
| 151 | 待定         | 自定义 |        | 主机1     | 16:20 | 16:50 | 未开始 | 🔖 ☑️ 🗑️ |
| 103 | 待定         | 自定义 | 最严重... | 点点滴滴!   |       |       | 未开始 | 🔖 ☑️ 🗑️ |
| 267 | 2019-05-07 | 周期  | 最高     | 预约煮稀饭   | 19:20 | 19:50 | 未开始 | 🔖 ☑️ 🗑️ |

★★

★★

### 1、自定义待办类型和优先级

最新版本禅道可以自定义待办的优先级和类型。

到后台---自定义---待办里即可根据使用需求来自定义。

### 2、新增待办

我的地盘---日程页面的右上角有新增待办按钮，点击即可进入新增待办的页面。

- 新增待办的时候，可以设定起止时间。也可以选择暂时不设定。
- 如果是私人事务，可以勾选上。

### 3、最新版本禅道可以新增待定日期和周期性待办

新增待办页面，勾选待定或者周期即可。

以下以创建周期性待办为例，周期的设置可以天、周、月为周期，间隔多少天以及提前几天创建同一待办。

还可以设置该待办过期时间。

周期性待办的创建，可以免去手动创建重复待办的麻烦。周期性待办创建成功后，系统会自动生成相关的计划任务，按照周期设置自动创建同一待办。

## 2、待办的相关操作

| 禅道项目管理软件                                           |    |     |    |         |            |       |        |        |  |
|----------------------------------------------------|----|-----|----|---------|------------|-------|--------|--------|--|
| 我的地盘 产品 项目 测试 文档 统计 组织 后台 Bug 编号(ctrl+g) GO! admin |    |     |    |         |            |       |        |        |  |
| 首页 日程 任务 Bug 测试 需求 项目 动态 档案 密码 积分 联系人              |    |     |    |         |            |       |        |        |  |
| 所有待办 239                                           | 本年 | 待定  | 未完 | 周期      | 2019-05-06 | 导出    | + 批量添加 | + 添加待办 |  |
| ID                                                 | 日期 | 类型  | P  | 待办名称    | 开始         | 结束    | 状态     | 操作     |  |
| 123                                                | 待定 | 自定义 |    | 天天天     | 06:00      | 06:00 | 未开始    |        |  |
| 122                                                | 待定 | 自定义 |    | 项目1     | 14:00      | 14:30 | 未开始    |        |  |
| 121                                                | 待定 | 自定义 |    | 点点滴滴    | 14:00      | 14:30 | 未开始    |        |  |
| 184                                                | 待定 | 自定义 |    | ffffwww | 14:20      | 14:50 | 未开始    |        |  |
| 183                                                | 待定 | 自定义 |    | ffff    | 14:20      | 14:50 | 未开始    |        |  |
| 182                                                | 待定 | 自定义 |    | ffff    | 14:20      | 14:50 | 未开始    |        |  |
| 181                                                | 待定 | 自定义 |    | ee      | 14:20      | 14:50 | 未开始    |        |  |
| 129                                                | 待定 | 自定义 |    | 222     | 14:30      | 15:00 | 未开始    |        |  |

### 1、浏览待办



禅道提供了各种标签来检索待办信息。

## 2、待办的指派和备注

待办可以指派给其他用户，也可以为待办添加备注。

点击待办列表页右侧操作按钮的指派，即可指派待办。

点击待办的名称，进入待办的详情页面，点击备注按钮，就可以给该待办添加备注信息了。

## 3、待办的转需求、转任务、转Bug操作

待办支持转需求、转任务、转Bug的功能。

点击待办名称，进入待办详情页面，就可以进行相关操作了。

待办转需求、转Bug，会先选择需要转入的产品，确定转入的产品后，进入新增需求和新增Bug的页面。

待办转为任务时，需要先选择所属的项目，然后进入新增任务的页面。

其实个人使用禅道，还可以借鉴项目管理的概念，把自己要处理的事情放在项目里面进行跟踪管理，也是非常方便的。比如买房，考研，复习考试等等。

# 9、禅道使用的基本流程和产品、研发、测试之间的三权分立

**简介：**禅道的基本使用流程：创建产品，维护需求，创建项目，关联需求，分解任务，测试，提交bug。

上一个章节我们讲述了《禅道使用的简流程》，您完全可以按照自己的实际情况选择简流程来使用。但从我们设计初衷来讲，还是希望您可以将禅道完整的功能使用起来。本章后续系列文章我们将向您展示禅道使用的基本流程。

禅道管理软件中，核心的三种角色：产品经理、研发团队和测试团队，这三者之间通过需求进行协作，实现了研发管理中的三权分立。其中产品经理整理需求，研发团队实现任务，测试团队则保障质量，其三者的关系如下图：

## 禅道项目管理三权分立图

www.zentao.net



基本流程如下：

1. 产品经理创建产品
2. 产品经理创建需求
3. 项目经理创建项目
4. 项目经理确定项目要做的需求
5. 项目经理分解任务，指派到人。
6. 测试人员测试，提交bug。

在开始讲解禅道的基本使用流程之前，先让我们了解下敏捷开发和scrum。

## 10、敏捷开发及scrum简介

**简介：**在正式讲述禅道的基本使用流程之前，先让我们了解下敏捷开发和scrum。因为禅道的设计理念基础是基于scrum的，因此了解scrum的基本概念有助于您了解和使用禅道。

在正式讲述禅道的基本使用流程之前，先让我们了解下敏捷开发和scrum。因为禅道的设计理念基础是基于 [scrum](#) 的，因此了解scrum的基本概念有助于您了解和使用禅道。

### 1、敏捷开发

我们比较熟知的软件项目管理方法是瀑布。其基本流程是需求->设计->开发->测试。基本假设只要把每一个环节都做正确，那么终得到的结果也是正确的。瀑布开发有非常成功的案例，比如微软。但从总体来讲，瀑布项目失败率比较高。国外的软件先行者们针对瀑布开发中暴露出来的问题进行了一系列的探索、思考和总结，终提出了Agile的概念，中文翻译为敏捷开发。敏捷开发有很多种方式，其中scrum是比较流行的一种。

### 2、scrum

## 2.1 scrum里面的角色

scrum由product owner(产品经理)、scrum master(项目经理)和team(研发团队)组成。

- 其中product owner负责整理user story(用户故事)，定义其商业价值，对其进行排序，制定发布计划，对产品负责。
- scrum master 负责召开各种会议，协调项目，为研发团队服务。
- 研发团队则由不同技能的成员组成，通过紧密协同，完成每一次迭代的目标，交付产品。

## 2.2 迭代开发

与瀑布不同，scrum将产品的开发分解为若干个小sprint(迭代)，其周期从1周到4周不等，但不会超过4周。参与的团队成员一般是5到9人。每期迭代要完成的user story是固定的。每次迭代会产生一定的交付。

## 2.3 scrum的基本流程



scrum的基本流程如上图所示：

- 产品负责人负责整理user story，形成左侧的product backlog。
- 发布计划会议：product owner负责讲解user story，对其进行估算和排序，发布计划会议的产出就是制定出这一期迭代要完成的story列表，sprint backlog。
- 迭代计划会议：项目团队对每一个story进行任务分解，分解的标准是完成该story的所有任务，终每个任务都有明确的负责人，并完成工时的初估计。
- 每日例会：每天scrum master召集站立会议，团队成员回答昨天做了什么今天计划做什么，有什么问题。
- 演示会议：迭代结束之后，召开演示会议，相关人员都受邀参加，团队负责向大家展示本次迭代取得的成果。期间大家的反馈记录下来，由po整理，形成新的story。
- 回顾会议：项目团队对本期迭代进行总结，发现不足，制定改进计划，下一次迭代继续改进，已达到持续改进的效果。

上面所讲的就是scrum的基本流程，后面我们会陆续在相关地方再做进一步的阐述。禅道的基本框架是基于scrum的，因此您对scrum的流程有了基本了解之后，再来看禅道就会比较容易了解。

# 11、禅道和scrum的对应关系

**简介：**禅道在设计的时候，基础框架是基于scrum的，同时又补充完善了测试，文档，发布，计划等功能。肯定有同学会问，是不是我使用禅道，就必须是用敏捷。答案当然是否定的，您完全可以使用禅道来进行瀑布式管理，也可以使用得很好。正是基于这一点考虑，我们在设计禅道的时候，都是采用了比较中性的概念，而不是scrum里面特有的概念。

禅道在设计的时候，基础框架是基于scrum的，同时又补充完善了测试，文档，发布，计划等功能。肯定有同学会问，是不是我使用禅道，就必须是用敏捷。答案当然是否定的，您完全可以使用禅道来进行瀑布式管理，也可以使用的很好。正是基于这一点考虑，我们在设计禅道的时候，都是采用了比较中性的概念，而不是scrum里面特有的概念。下面让我们来看下 [禅道和scrum](#) 的对应关系。

## 1、概念的对应

| scrum          | 禅道  | 备注                                    |
|----------------|-----|---------------------------------------|
| product        | 产品  | 禅道里面使用了产品的概念。                         |
| user story     | 需求  | 其实标准的翻译应该为用户故事，不过禅道还是采用了比较中性的概念。      |
| sprint         | 项目  | 很多朋友对产品和项目的概念不是很清楚，其实有了这个对应关系就很容易理解了。 |
| task           | 任务  | 这两个概念是一致的。                            |
| team           | 团队  |                                       |
| burndown chart | 燃尽图 |                                       |
| 潜在交付物          | 发布  |                                       |

## 2、角色的对应

| scrum         | 禅道   | 备注                   |
|---------------|------|----------------------|
| product owner | 产品经理 | 禅道使用的还是比较中性的产品经理的概念。 |
| scrum master  | 项目经理 |                      |
| team          | 团队成员 |                      |

## 3、禅道新增的概念

除了scrum基本的概念外，禅道还补充完善了下面概念，以达到对软件生命周期的完整覆盖。

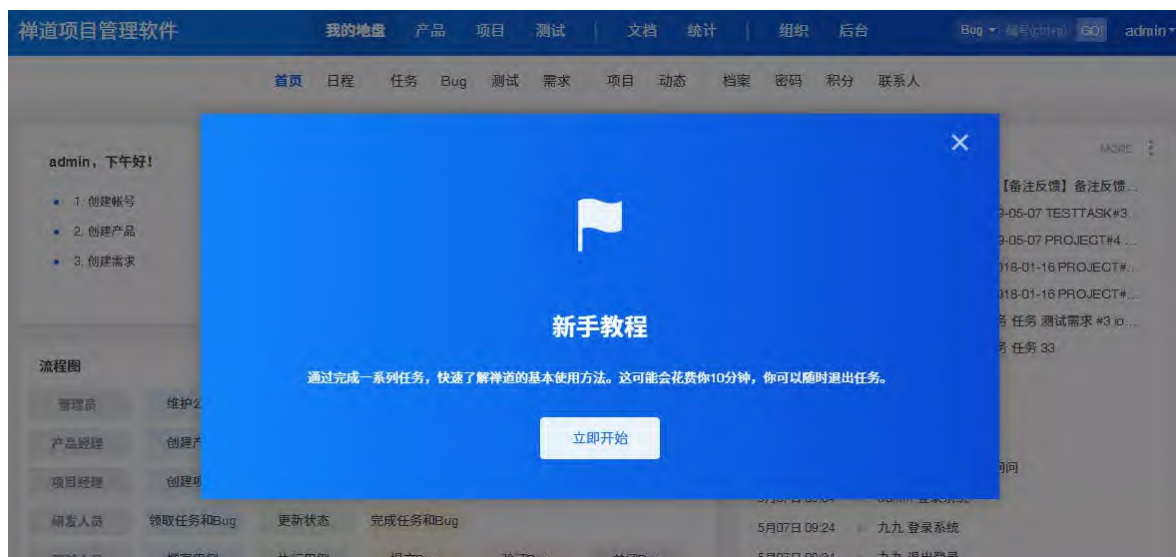
- 产品发布计划
- 产品路线图
- 项目版本
- bug
- 测试用例
- 测试单
- 测试结果
- 文档
- 待办

现在我们了解了禅道和敏捷，和scrum之间的关系了，让我们来看下禅道的基本使用流程吧！

## 12、禅道的新手教程

禅道8.2beta版本开始，添加了禅道新手教程，以任务的形式教大家禅道的最简的使用操作。

首次登录使用禅道时，会提示是否进入禅道新手教程页面。或者点击右上角的 帮助--新手教程。



点击 立即开始 后，页面会跳转到新手教程页面。

新手教程包含8个任务：创建帐号、创建产品、创建需求、创建项目、管理团队、关联需求、分解任务、提Bug。

禅道页面右侧会显示具体任务，左侧的禅道页面有当前任务的操作提示。

按照教程流程具体操作即可，需要注意的是 新手教程里8个任务填写的数据系统是不会保存的。

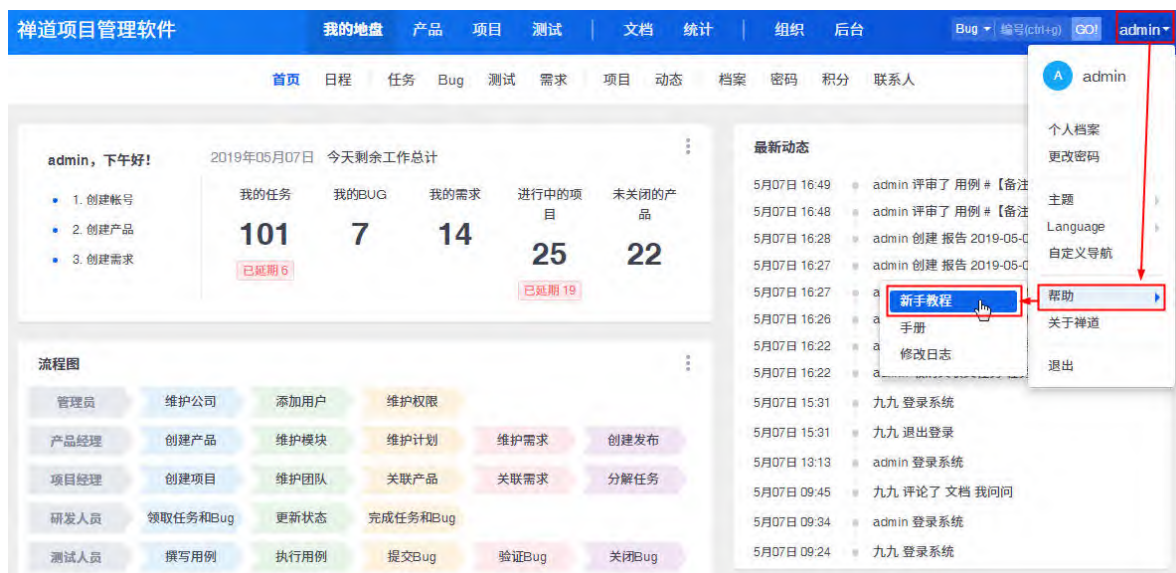
需要中途退出新手教程，点击右上角的 退出教程 按钮。



新手教程和禅道使用手册都已内置到禅道页面右上角的 帮助 栏里。

想重新打开新手教程或者使用期间有不清楚的地方，可以点击 帮助 获取。





## 13、创建第一个产品

简介：本文讲述如何在禅道中创建一个产品。

禅道的设计理念是围绕产品展开的，因此我们首先要做的就是创建一个产品。进入产品主页，在1.5级导航的下拉菜单中，选择“添加产品”。或者先进入“所有产品”页面，点击右侧的“添加产品”。之后便可以添加产品的具体信息。



新增产品的时候，需要设置产品的名称、代码，几个负责人信息。

添加产品

产品名称

禅道项目管理软件专业版

产品代号

zentao-pms-pro

产品线

维护产品线

产品负责人

A.admin

测试负责人

L.李文瑞

发布负责人

D.婷婷

产品类型

正常

H1

可以在编辑器直接贴图。

产品描述

保存模板

应用模板

- 产品名称和产品代号是必填项。其中产品代号可以理解为团队内部约定俗称的一个称呼，比如我们禅道的代号是zentao，需要是英文字母和数字的组合。

- 产品线：该产品属于那一个产品线。比如禅道这个产品线，下面包含禅道专业版，禅道开源版，禅道企业版。
- 产品负责人：负责整理需求，对需求进行解释负责，制定发布计划，验收需求。
- 测试负责人：可以为某一个产品指定测试负责人，这样当创建bug，而不知道由谁进行处理的时候，该产品的测试负责人会成为默认的负责人。
- 发布负责人：由这个角色负责创建发布。
- 产品类型：默认是正常的类型，还可以选择多分支（适用于客户定制场景）和多平台（适用于跨平台应用开发，比如iOS，安卓，pc端等）的产品。
- 访问控制：可以设置产品的访问权限，其中默认设置只要有产品视图的访问权限就可以访问。如果这个产品是私有产品，可以将其设置为私有项目，那么就只有项目团队成员才可以访问。或者还可以设置白名单，指定某些分组里面的用户可以访问该产品。

## 14、添加第一个需求

**简介：**本文讲述如何在禅道中创建第一个需求。

上篇文章我们讲述了如何添加产品。有了产品之后，我们就可以来创建我们的第一个需求了。

我们产品经理可能都习惯了写需求设计文档，或者规格说明书，通过一个非常完整的word文档将某一个产品的需求都定义出来。但在禅道里面，我们提倡按照功能点的方式来写需求。简单来讲，就是将原来需求设计文档中的每一个功能点摘出来，录在禅道里面，作为一个个独立的功能点。如果按照scrum标准走的话，我们可以称之为 用户故事(user story)。所谓用户故事，就是来描述一件事情，作为什么用户，希望如何，这样做的目的或者价值何在，这样有用户角色，有行为，也有目的和价值所在，非常方便与团队成员进行沟通。

创建需求的步骤如下：

1. 使用产品经理角色登录系统。
2. 进入产品视图。
3. 点击二级导航的“需求”，在页面右侧，有提需求的按钮，并支持批量创建，点击新增需求的页面。



- 需求的标题是必填项。
- 所属计划和模块，可以暂时保留为空。
- 需求评审，我们选上不需要评审，这样新创建的需求状态就是激活的，否则只有指定人员进行评审通过后，才能转为激活状态。只有激活状态的需求才能关联到项目中，进行开发。
- 需求可以设置抄送给字段，这样需求的变化都可以通过email的形式抄送给相关人员。
- 可以设置关键词，这样可以比较方便的通过关键词进行检索。

批量添加需求时，支持多图上传。

点击多图上传，选择要上传的图片。

多图上传，目前支持jpg, jpeg, gif, png格式的图片。

图片上传成功后，程序会把图片的名作为需求的名称，图片作为需求的内容。



## 15、开始第一个项目

简介：本篇文章讲述如何通过禅道项目管理软件中开始一个项目。

产品经理按照我们前面的操作创建需求之后，下面该 [项目经理](#) 上场了。下面我们一起来看下如何在禅道中开始一个 [项目](#)。

### 1、创建项目

创建项目的入口有多个：

a. 进入项目视图，点击左侧1.5级导航下拉菜单，可以直接点击“添加项目”。



- b.或者先进入所有项目页面，点击右侧的“添加项目”按钮。
- c.再或者在项目主页中的项目总览区块中也有“添加项目”按钮。



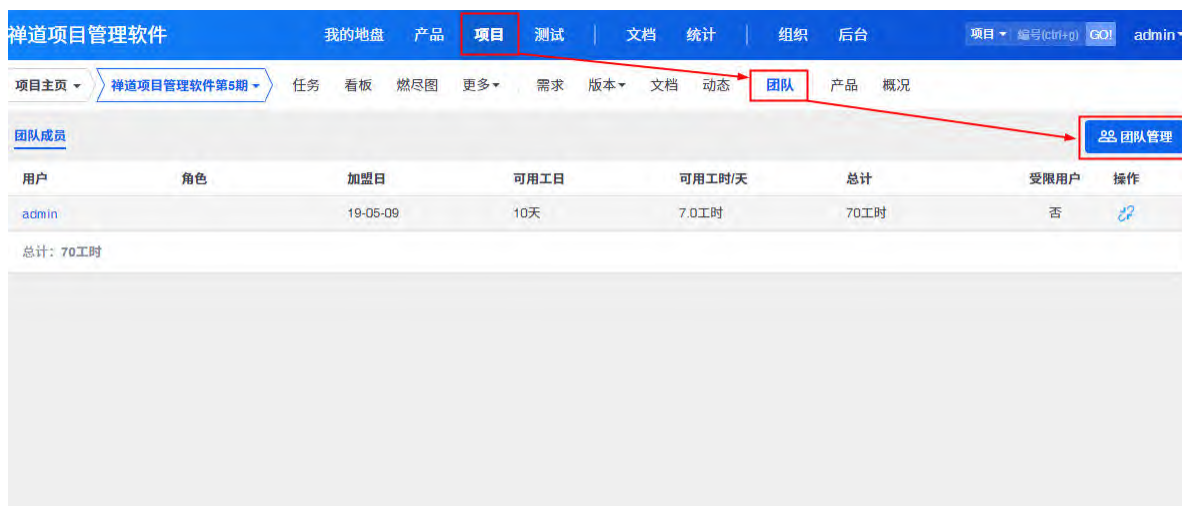
进入项目添加的页面后，在这个页面设置项目名称、代号、起止时间、可用工作日、团队名称、项目目标和项目描述等字段。其中关联产品是可以为空的。

- 和我们前面讲述的简使用流程有一点区别就是在添加项目的时候，需要关联下产品，可以多选。
- 项目可以控制它的访问权限，分为默认、私有和自定义白名单三种。

## 2、设置团队

创建项目点击保存按钮，会提示项目创建成功，然后可以选择设置团队。或者从项目视图中的团队菜单，也可以进行项目的团队管理。

在维护项目团队的时候，需要选择都是哪些用户可以参与到这个项目中，同时需要设置这个用户在本项目中的角色（角色可以随便设置，比如风清扬，冬瓜一号等）。可用工作日和可用工时每天需要仔细设置。通常来讲，一个人不可能每天8小时投入，也不可能一星期七天连续投入。设置完毕之后，系统会自动计算这个项目总得可用工时。



## 16、确定项目要完成的需求列表

**简介：**迭代开发区别于瀑布式开发就是它将众多的需求分成若干个迭代来完成，每个迭代只完成当下优先级高的那部分需求。禅道软件中项目关联需求的过程，就是对需求进行排序筛选的过程。

迭代开发区别于瀑布式开发就是它将众多的需求分成若干个迭代来完成，每个迭代只完成当下优先级高的那部分需求。禅道软件中项目关联需求的过程，就是对需求进行排序筛选的过程。下面我们来看下如何关联需求。

### 1、关联产品

如果在创建项目的时候，已经关联过产品，可以忽略这个步骤。

1. 以项目经理身份登录。
2. 进入项目视图。
3. 点击“关联产品”按钮。然后点选该项目相关的产品即可。



### 2、关联需求





1. 在关联需求的时候，可以按照优先级进行排序。
2. 关联的需求状态必须是激活的（评审通过，不能是草稿）

## 17、为需求分解任务

**简介：**确定了当前项目要完成的需求之后，下一步的操作就是为每一个需求做任务分解。

需求确定之后，项目中几个关键的因素都有了：周期确定、资源确定、需求确定。下面我们要做的事情就是为每一个需求做wbs任务分解，生成完成这个需求的所有的任务。note：是完成需求的所有任务，这里面包括但不限于设计，开发，测试等。

### 1、分解任务

在项目的需求列表页面，可以很方便地对某一个需求进行任务分解。同时还可以查看这个需求已经分解的任务数。列表字段中的 T/B/C分别代表 **任务数/bug数/用例数**。



建任务

所属项目 测试bug的关联

任务类型

所属模块 /广告歌

指派给 ☐ 多人

相关需求 73广告歌 (优先级:0,预计工时:0)

任务名称

同需求 优先级 ③ 预计

颜色标签

可以在编辑器直接贴图。

任务描述

- 这时候创建任务，就可以选择需求了。
- 我们同时提供了需求查看的链接。
- 如果需求和任务的标题是一样的，可以通过“同需求”按钮快捷的复制需求的标题。
- 支持自定义任务标题的颜色。

## 2、多人任务和子任务

禅道9.6版本开始，新增多人任务和子任务功能。

### 1、多人任务

建任务的时候，指派给选择框里，勾选多人。

会出现团队按钮，点击团队，就可以选择将该任务指派给多人。

在团队里，选择团队成员，填写上对应的预计工时。

最右侧的向上和向下的箭头，可以排序。

禅道项目管理软件

我的地盘 产品 项目 测试 文档 统计 组织 后台

任务 编号(0114g) GO! admin

项目主页 测试bug的关联 任务 看板 燃尽图 更多 需求 版本 文档 动态 团队 产品 概况

建任务

所属项目 测试bug的关联

任务类型

所属模块 /

指派给 ☐ 团队 ☒ 多人

相关需求

任务名称

同需求 优先级 ③ 预计

禅道项目管理软件

我的地盘 产品 项目 测试 文档 统计 组织 后台

任务 编号(0114g) GO! admin

项目主页 测试bug的关联 任务 看板 燃尽图 更多 需求 版本 文档 动态 团队 产品 概况

建任务

所属项目 测试bug的关联

任务类型

所属模块 /

指派给

相关需求

任务名称

团队

|         |    |    |   |   |   |
|---------|----|----|---|---|---|
| A.admin | 6  | 小时 | + | + | × |
| H:九九    | 7  | 小时 | + | + | × |
| D:婷婷    | 8  | 小时 | + | + | × |
|         | 预计 | 小时 | + | + | × |
|         | 预计 | 小时 | + | + | × |

预计



选择指派给多人后，"指派给"菜单里会显示指派的人员名单。

该任务的预计工时是每个指派给人员的预计工时之和。

#### 特别说明：

- 1、多人任务，指派给团队成员的顺序，影响到任务的开始和转交顺序。
- 2、多人任务是创建一个任务，指派给多人。事务类型的任务是同时创建多条相同的任务分别指派给多人。
- 3、多人任务，只能由指派给的第一个人来开始，完成后转交给第二个人，以此类推。
- 4、若点击 **同需求**，则名称、优先级、描述、工时都会同步。

## 2、子任务

正常创建任务后，返回项目任务列表页。

在任务列表页的右侧操作按钮那，点击最右侧的子任务按钮，即可创建该任务的子任务。

创建子任务的页面，其实就是批量添加任务的页面。填写好内容，保存就好。

禅道项目管理软件

我的地盘 产品 项目 测试 文档 统计 组织 后台

任务 编号(必填) GO! admin

项目主页 测试bug的关联

任务 看板 燃尽图 更多 需求 版本 文档 动态 团队 产品 概况

所有模块 所有 未关闭 7 指派给我 由我参与 已延期 需求变更 更多 自定义 搜索

广告歌

维护模块

列表页是否显示模块名

| ID  | P | 任务名称           | 状态  | 预计 | 消耗 | 剩余 | 进度  | 截 | 操作          |
|-----|---|----------------|-----|----|----|----|-----|---|-------------|
| 244 | 3 | 任务6            | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 211 | 3 | 转任务 > [BUG#69] | 未开始 | 15 | 5  | 5  | 50% |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 243 | 3 | 子任务5           | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 242 | 3 | 子任务4           | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |

禅道项目管理软件

我的地盘 产品 项目 测试 文档 统计 组织 后台

任务 编号(必填) GO! admin

项目主页 测试bug的关联

任务 看板 燃尽图 更多 需求 版本 文档 动态 团队 产品 概况

所有模块 所有 未关闭 7 指派给我 由我参与 已延期 需求变更 更多 自定义 搜索

广告歌

维护模块

列表页是否显示模块名

| ID  | P | 任务名称           | 状态  | 预计 | 消耗 | 剩余 | 进度  | 截 | 操作          |
|-----|---|----------------|-----|----|----|----|-----|---|-------------|
| 244 | 3 | 任务6            | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 211 | 3 | 转任务 > [BUG#69] | 未开始 | 15 | 5  | 5  | 50% |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 243 | 3 | 子任务5           | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 242 | 3 | 子任务4           | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 241 | 3 | 子任务3           | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 213 | 3 | 子任务2           | 未开始 | 5  | 0  | 5  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |
| 212 | 3 | 子任务1           | 未开始 | 0  | 0  | 0  | 0%  |   | ▶ ◀ ☑ ☒ ☑ ☒ |

创建成功的子任务，可以在任务列表页中查看。

可以点击父任务标题前面的三角符号，来切换子任务的查看方式。

默认是展开显示的。

#### 特别说明：

- 1、创建子任务后，父任务的预计、消耗、剩余工时，是所有子任务的预计、消耗、剩余工时之和。
- 2、创建父任务时，填写了相关的工时信息，再添加子任务后，子任务的相关工时之和，会覆盖掉父任务的相关工时。
- 3、多人任务不可以创建子任务。

### 3、任务分解的几个注意事项

1. 需要将所有的任务都分解出来。这里面包括设计，开发，测试，美工，甚至包括购买机器，部署测试环境等等。
2. 任务分解的粒度越小越好，比如几个小时就可以完成。
3. 如果一个任务需要多个人负责，继续考虑将其拆分。
4. 事务型的事务可以批量指派，比如需要让团队里面的每一个人都写个项目总结，可以选择类型是事务，然后批量指派给团队里面的所有人员。
5. 任务的类型请仔细设置，这个会涉及到需求研发阶段的自动计算。后面我们会有讲解。
6. 任务的分配最好是自由领取，这样可以大程度上调动大家的积极性。

## 18、提交bug

**简介：**项目进展到后期主要的工作就是测试。测试人员和开发通过bug进行互动，保证产品的质量。

项目进展到后期主要的工作就是测试。测试人员和开发通过bug进行互动，保证产品的质量。

### 1、禅道里面bug的基本处理流程

禅道里面缺陷处理的基本流程是：测试提交bug => 开发确认bug => 开发解决bug => 测试验证bug => 测试关闭bug。

如果bug验证没有通过，可以激活：测试提交bug => 开发确认bug => 开发解决bug => 测试验证bug => 测试激活bug => 开发解决bug => 测试验证 => 测试关闭。

还有一个流程就是bug关闭之后，又发生了。测试提交bug => 开发确认bug => 开发解决bug => 测试验证bug => 测试关闭bug => 测试激活bug => 开发解决bug => 测试验证 => 测试关闭。

### 2、提出bug

易软天创 我的地盘 产品 项目 **测试** 文档 统计 组织 后台 Bug 编号(dh+g) GO! admin

测试主页 公司企业网站建设 - Bug 用例 测试单 报告 套件 用例库

所有模块 您现在还没有模块信息 请维护测试模块 维护模块 列表页面是否显示模块名

所有 未关闭 7 由我创建 指派给我 由我解决 待关闭 未解决 更多 搜索

| ID  | 级别 | P | 确认 | Bug标题       | 状态 | 创建者     | 创建日期        | 指派给   | 方案 | 操作      |
|-----|----|---|----|-------------|----|---------|-------------|-------|----|---------|
| 010 | 👍  | ③ | 是  | 123         | 激活 | admin   | 07-09 15:25 | 未指派   |    | 🔍 📄 📧 🔄 |
| 007 | 👍  | ③ | 是  | 售后服务的测试用例测试 | 激活 | admin   | 04-23 11:08 | admin |    | 🔍 📄 📧 🔄 |
| 005 | 👍  | ③ | 否  | 售后服务的测试用例测试 | 激活 | admin   | 04-22 14:54 | 未指派   |    | 🔍 📄 📧 🔄 |
| 004 | 👍  | ③ | 否  | 售后服务页面问题    | 激活 | tester3 | 06-05 11:00 | dev1  |    | 🔍 📄 📧 🔄 |
| 003 | 👍  | ③ | 否  | 成果展示页面问题    | 激活 | tester2 | 06-05 10:58 | dev3  |    | 🔍 📄 📧 🔄 |
| 002 | 👍  | ③ | 否  | 新闻中心页面问题    | 激活 | tester1 | 06-05 10:57 | dev2  |    | 🔍 📄 📧 🔄 |
| 001 | 👍  | ③ | 否  | 首页页面问题      | 激活 | tester1 | 06-05 10:56 | dev1  |    | 🔍 📄 📧 🔄 |

共 7 项 每页 20 项 < 1/1 >

- 在创建bug的时候，必填的字段是影响版本，bug标题，重现步骤这些基本的信息。
- 所属项目，相关产品，需求可以忽略。
- 创建bug的时候，可以直接指派给某一个人去处理。如果不清楚的话，可以保留为空。
- 批量添加bug时，支持多图上传。支持jpg，jpeg，gif，png格式的图片，图片上传成功后，图片名称将作为bug的名称，图片作为bug的内容。

### 3、处理bug

当一个bug指派给某一位研发人员之后，他可以来确认解决这个bug。在对bug进行处理之前，需要先找到需要自己处理的bug。禅道提供了各种各样的检索方式，比如指派给我，可以列出所有需要我处理的bug。

| ID  | 级别 | P | 确认 | Bug标题       | 状态 | 创建者     | 创建日期        | 指派给   | 方案 | 操作      |
|-----|----|---|----|-------------|----|---------|-------------|-------|----|---------|
| 010 | 3  | 3 | 是  | 123         | 激活 | admin   | 07-09 15:25 | 未指派   |    | [Icons] |
| 007 | 3  | 3 | 是  | 售后服务测试用例测试  | 激活 | admin   | 04-23 11:08 | admin |    | [Icons] |
| 005 | 3  | 3 | 否  | 售后服务的测试用例测试 | 激活 | admin   | 04-22 14:54 | 未指派   |    | [Icons] |
| 004 | 3  | 3 | 否  | 售后服务页面问题    | 激活 | tester3 | 06-05 11:00 | dev1  |    | [Icons] |
| 003 | 3  | 3 | 否  | 成果展示页面问题    | 激活 | tester2 | 06-05 10:58 | dev3  |    | [Icons] |
| 002 | 3  | 3 | 否  | 新闻中心页面问题    | 激活 | tester1 | 06-05 10:57 | dev2  |    | [Icons] |
| 001 | 3  | 3 | 否  | 首页页面问题      | 激活 | tester1 | 06-05 10:56 | dev1  |    | [Icons] |

**确认bug**：确认该bug确实存在后，可以将其指派给某人，并指定bug类型、优先级、备注、抄送等。

**解决bug**：当bug修复解决后，点击解决，指定解决方案、日期、版本，并可将其再指派给测试人员。

**关闭bug**：当研发人员解决了bug之后，bug会重新指派到bug的创建者头上。这时候测试人员可以来验证这个bug是否已经修复。如果验证通过，则可以关闭该bug。（bug列表页和详情页中都有“关闭”按钮。）

**编辑bug**：对bug进行编辑操作。

**复制bug**：复制创建当前bug，在此基础上再做改动，避免重新创建的麻烦。

## 19、视频教程：第一个演示项目

如果您看了以上手册，对禅道的基本使用还有疑问或者是流程不清楚的，建议您看看我们的这个视频教程。

该视频教程由禅道创始人王春生（春哥）主讲。

视频教程按照禅道基本的流程，详细的实际演示介绍一个项目具体的组建、开发、发布等操作。

把以上的基本使用文档串联起来，整体完整的介绍禅道的基本使用。

视频地址：<http://www.zentao.net/video/use-of-zentao-the-first-demo-79961.html>

禅道项目管理系列视频教程可移步：<http://www.zentao.net/video/c1455.html>

## 20、更多配置查看官方文档

<https://www.zentao.net/book/zentaopmshelp/40.html>

千锋云计算学院