PYTHON 运行环境（含 REDIS CENTOS 6.8 64 位 安全优化）

镜像使用手册

北京君云时代科技有限公司
北京市朝阳区四惠东华腾世纪总部公园 A 座 6 层

2017-4-25

前 言

【版权与独立性说明】

（1）本文声明所介绍技术产品是基于北京君云时代科技有限公司进行的研究工作和取得的研究成果，“君云时代”（简称，下同）对本文及相应技术产品内容单独完全享有版权，任何形式的侵权盗用行为将会被依法追究责任。

（2）文中介绍技术流程与操作要点不一定完全体现镜像功能，具体细节以实际操作为准，解释权归“君云时代”所有，欢迎广大用户及技术爱好者参与使用并提出宝贵建议。

（3）如有各类建议及投诉意见，请及时拨打技术支持电话: 4008005185 转 10449，我们将真诚为您反馈处理结果。

【公司简介】

北京君云时代科技有限公司成立之初以企业级用户的信息化建设需求为导向，分析大中小型企业、政府、教育、电商、金融、信息传播等行业的信息化及信息安全现状，采用先进规范的信息技术和管理标准，致力于为广大企业级用户提供综合的信息安全服务、集成服务、云端一站式解决方案。随着云计算技术和市场的快速发展，公司以“中国最具竞争力的云服务商”为企业发展远景，于 2016 年 3-6 月先后与阿里云市场、腾讯云市场、华为云市场达成合作。近一年用户量快速成长，截止 2016 年底用户量累计过万。主要业务分布：集成项目，线上和线下解决方案。主要客户为电商、金融、信息传播等行业用户。君云时代将始终秉承“精、诚、一、新”的企业文化，助力企业用户轻松步入 DT 时代。

【联系我们】

1. 公司地址：

北京市朝阳区四惠东华腾世纪总部公园 A 座 6 层

2. 公司网站：

http://www.cldera.com

3. 通讯联络：

电话技术支持: 4008005185 转 10449

旺旺技术支持：cldera

邮箱技术支持：support@cldera.com

目 录

1 产品基本介绍

1.1 镜像配置环境

（1）操作系统：CentOS 6.8 64 位。

（2）python 运行环境：python 多版本自由切换，django V1.10.6 ，pyenv V1.0.10，nginx V1.10.2，mysql V5.7.73，ipython V5.3.0，git V1.7.1，redis V3.2.5。

1.2 镜像安装说明

本镜像环境中集成的 python 环境基于 linux 最新最安全的 yum 仓库进行安装，用户如需升级各软件版本，可使用 yum 进行升级，方便可靠。对于可能出现的相关故障问题，可依托于公司技术支持解决。

2 Python 命令介绍

2.1 关于 pyenv 的使用说明

pyenv 是一个 python 多版本管理工具。我们经常会遇到这种情况：系统自带的 python 是 2.x，我们需要 python2.x 或 3.x 中的某些特性。此时需要在系统中安装多个 python，但又不能影响系统自带的 python，即需要实现多个 python 版本共存。这时应用 pyenv 就很方便地满足了我们这种需求。

常用命令：

查看可安装的版本：

```
#pyenv install –list
```

安装指定版本：

```
#pyenv install 3.4.1 –v
```

安装完成后需要对数据库进行更新：

#pyenv rehash

卸载指定版本：

#pyenv uninstall 2.7.1

查看当前已安装的 python 版本：

#pyenv versions

设置全局的 python 版本：

#pyenv global 3.4.1

临时改变 python 版本（临时改变到系统 python 版本）：

#pyenv local system

可通过在命令行输入 python，查看输出内容来确认 python 版本信息。

2.2 关于 ipython 的使用说明

ipython 是一个增强的交互式 pyhton shell，具有 tab 补全功能、对象自省功能以及强大的历史机制，并能实现内嵌的源代码编辑等。

（1）tab 补全功能：

```
In [1]: import sys

In [2]: sys.
     sys.abiflags              sys.builtin_module_names  sys.displayhook           sys.executable
     sys.api_version           sys.byteorder             sys.dont_write_bytecode   sys.exit
     sys.argv                  sys.call_tracing          sys.exc_info              sys.flags
     sys.base_exec_prefix      sys.callstats             sys.excepthook            sys.float_info
     sys.base_prefix           sys.copyright             sys.exec_prefix           sys.float_repr_style
```

（2）对象自省：

```
In [4]: import subprocess

In [5]: dir(subprocess.call)
Out[5]:
['__annotations__',
 '__call__',
 '__class__',
 '__closure__',
 '__code__',
 '__defaults__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
```

```
'__eq__',
'__format__',
'__ge__',
'__get__',
'__getattribute__',
'__globals__',
'__gt__',
'__hash__',
'__init__',
'__kwdefaults__',
'__le__',
'__lt__',
'__module__',
'__name__',
'__ne__',
'__new__',
'__qualname__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__']
```

（3）查看历史：

```
In [7]: history
a = 1
b = 2
d = {}
e = []
for i in range(20):
    e.append(i)
    d[i]=b
print d[1]
history
```

运行 shell，必须使用转义符号：

```
In [8]: !echo $$PATH
/root/.pyenv/versions/3.4.1/bin:/root/.pyenv/libexec:/root/.pyenv/plugins/python-build/bin
```

此外，ipython 还有 debugger 接口、定义宏、定义环境等功能，可以帮助开发人员高效工作，快速定位问题。

2.3 关于 pip 的使用说明

pip 是一个 python 包管理工具，主要用于安装和管理 pypi 上的软件包。

安装包：

#pip install package_name

查看已安装软件包
#pip show –files package_name

检查需要更新的软件包：
#pip list –outdated

升级软件包：
#pip install –upgrade package_name

卸载软件包：
#pip uninstall package_name

2.4 关于 nginx 的使用说明

nginx 是一个高性能的 HTTP 和反向代理服务器，其特点是轻量级且高并发能力强。

配置文件：/etc/nginx/nginx.conf（/etc/nginx/conf.d/）
日志路径：/var/log/nginx/
常用命令：

查看版本：
#nginx –v

显示帮助信息：
#nginx –h

语法检测：
#nginx –t

启动命令：
#service nginx start

关闭命令：（先查看进程号，再 kill 掉进程）
#service nginx stop

重启命令：
#nginx –s reload

2.5 关于 mysql 的使用说明

关于 mysql 的配置和相关命令如下。

配置文件：/etc/my.cnf
数据库文件目录：/var/lib/mysql
日志文件：/var/log/mysqld.log
数据库密码存放文件路径：/usr/local/mysql.log

启动命令：
#/etc/init.t/mysqld start

连接数据库：
#/etc/init.d/mysqld stop

3 Redis 命令介绍

3.1 Redis 安装目录

安装目录：
/usr/local/redis

数据目录：
/usr/local/redis/db

日志目录：

/usr/local/redis/log

pid 目录：
/usr/local/redis/pid

配置文件：
/usr/local/redis/redis.conf

3.2 Redis 常用命令

启动命令：
/etc/init.d/redis start

关闭命令：
/etc/init.d/redis stop

连接命令：
redis-cli –h host –p port –a password

数据库备份：
127.0.0.1:6379> SAVE
127.0.0.1:6379> BGSAVE

查看配置：
#redis-cli
127.0.0.1:6379> config get loglevel
使用*号查看所有配置：
127.0.0.1:6379> config get *

编辑配置：
127.0.0.1:6379> CONFIG SET CONFIG_SETTING_NAME NEW_CONFIG_VALUE
127.0.0.1:6379> config set loglevel "notice"

```
OK
127.0.0.1:6379> config get loglevel
1) "loglevel"
2) "notice"

获取统计信息：
127.0.0.1:6379>INFO
```

3.3 Redis 配置说明

配置文件部分参数说明：

①redis 默认不以守护进程方式启动，使用 yes 修改该配置项启用守护进程

```
daemonize no
```

②指定 pid 存放路径

```
pidfile /usr/local/redis/pid/redis.pi
```

③指定 redis 监听端口， 默认监听 6379

```
port 6379
```

④绑定主机地址

```
bind 127.0.0.1
```

[注]如果修改该地址，请修改启动脚本/etc/init.d/redis 中的 HOST 变量。使用 redis-cli 连接时使用-h 指定绑定地址。

⑤客户端闲置多少秒后断开连接

```
timeout 300
```

⑥指定日志级别，redis 支持 debug，verbose，notice，warning 四种日志级别。

```
Loglevel notice
```

⑦数据库数量

```
databases 16
```

⑧在指定时间内，多少次更新时将数据同步到数据文件

默认配置如下，分别表示 900s 内有 1 个更新，300s 内有 10 个更新，60s 内有 10000 个更新。

```
save 900 1
save 300 10
save 60 10000
```

⑨指定存储至本地数据库时是否压缩

```
rdbcompression yes
```

⑩本地数据库名，默认为 dump.rdb

```
dbfilename dump.rdb
```

⑪数据库存放目录

```
dir /usr/local/redis/db
```

⑫指定 redis 连接密码，默认关闭

```
requirepass foobared
```

⑬设置同一时间最大连接数，为 0 表示不做限制，默认关闭

```
maxclients 10000
```

⑭redis 最大内存

```
maxmemory <bytes>
```

⑮当有更新操作时是否进行日志记录

```
appendonly no
```

⑯更新日志文件

```
appendfilename "appendonly.aof"
```

⑰更新日志规则，no 选项依赖操作系统同步，always 选项在每次写操作时同步，everysec 选项表示每秒同步一次。

```
appendfsync everysec
```

4 案例部署

4.1 代码连接 mysql 配置

4.1.1 python 连接 mysql 配置

连接数据库代码样例路径：/usr/local/ mysql_test.py

镜像默认使用 python3.4.1，因为需要导入 pymysql 包。下面代码以创建数据库为例进行说明：

```
#!/usr/bin/python
# coding: utf-8
# author: yangtao
# mail: yyyummy22@163.com
import pymysql

cxn = pymysql.Connect(host = '', user = '', passwd = '')
cur = cxn.cursor()
cur.execute("CREATE DATABASE junyun")
cur.close()
cxn.commit()
cxn.close()
```

其中，连接数据库的代码中 host 为数据库的 ip，user 为数据库用户名，passwd 为数据库密码。

4.1.2 django 连接 msyql 配置

django 样例路径：/usr/local/www/project

(1)首先使用如下命令创建 django 项目和 app。

```
#django-admin startporject protject_name
#django-admin startapp app_name
```

这里我们创建的项目名称和 app 名称分别为 project 和 app01。

(2)修改项目的配置文件 setting.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': '', #数据库名称
        'USER':'',  #连接数据库用户名
        'PASSWORD':'', #数据库密码
        'HOST':'', #数据库 ip 或机器名
        'PORT':'', #数据库端口号
    }
}
```

至此就完成了连接配置，接下来我们来创建数据库表。

(3)修改 app01 下的 models.py

```
#vim /usr/local/www/project/app01/models.py
class mydb(models.Model):
    name = models.CharField(max_length=50)
     sex = models.BooleanField()
```

这里我们创建两个字段，分别为 name（字符型）和性别（布尔型）。

(4)我们执行 django 命令来创建数据库表。

```
#python manage.py makemigrations
#python manage.py migrate
```

(5)这时数据库创建完成，我们可以在 mysql 命令行查看是否创建成功。

```
查看数据库：
 > show databases;
      > use db_name; #这里选择数据库名
> desc mydb;
```

4.2 django 项目部署

（1）配置 views。

```
# vim /usr/local/www/project/app01
 from django.http.response import HttpResponse
    def index(request):
return HttpResponse("君云时代科技有限公司")
```

（2）配置 url。

```
# vim /usr/local/www/project/urls.py
 from django.conf.urls import url

 from django.contrib import admin
 from app01.views import index
    urlpatterns = [
  url(r'^admin/', admin.site.urls),
  url(r'index/',index),
]
```

（3）启动 django。

```
# python manage.py runserver 0.0.0.0:8000
```

（4）这时我们在浏览器中输入：http://ip:8000/index 就可以访问到我们的页面。

4.3 nginx 反向代理

我们的 django 项目运行在 8000 端口上，需要配置 nginx 反向代理，从而通过 80 端口访问我们的服务。

```
# vim /etc/nginx/conf.d/default.conf
server{
    listen 80;
    server_name 118.190.83.187; #这里 ip 或者域名
    rewrite ^/$ /index/ break; #默认跳转至/index
    location / {
        proxy_pass http://118.190.83.187:8000; #django 的 ip 和端口
    }
}
```

配置完成后使用如下命令检查 nginx 语法，并重启 nginx。

```
#nginx –t
#nginx –s reload
```

这里我们可以在浏览器中输入 http://ip 就可以访问到我们的页面。

4.4 迁移 mysql 至数据盘

关闭 mysql 服务：
#service mysql stop

拷贝数据目录：（如果数据库比较大，这里需要较长时间）
#cd /var/lib/
#cp –a mysql /disk1/

修改配置文件：
#vim /etc/my.cnf
datadir=/var/lib/mysql 改为 datadir=/disk1/mysql
socket=/var/lib/mysql/mysql.sock 改为 socket=/disk1/mysql/mysql.sock
#vim /etc/init.d/mysqld
datadir "/var/lib/mysql"改为 datadir "/disk1/mysql"
# vim /usr/bin/mysqld_safe
DATADIR=/var/lib/mysql 改为 DATADIR=/disk1/mysql

建立 mysql.sock 的软链：

#ln –s /disk1/mysql/mysql.sock /var/lib/mysql/mysql.sock

至此，修改完成，我们启动 mysql

#service mysql start

★mysql 允许远程连接：

考虑 mysql 的相关安全性，在默认情况下只允许本地连接。在一些情况下，如果需要进行远程连接，可通过以下方法进行操作。

（1）使 root 用户可以远程连接：

```
>use mysql;
>update user set host='%' where user='root' and host='localhost';
>flush privileges;
```

（2）使特定用户从特定主机远程连接：

```
>use mysql;
>grant all privileges on *.* to 'username'@ip inentified by password with grant option;
>flush privileges;
```

我们推荐使用第二种方法，特定数据库对特定用户和 ip 授予连接权限，这种方式更加安全可靠。

4.5 迁移 redis 服务至数据盘

镜像部署的服务存储在系统盘中，随着服务量的增长，数据量也会与日俱增并大量占用系统盘空间。我们建议如果挂载数据盘，则优先考虑将数据目录迁移到数据盘中，这样将大大提高服务器的性能。

在迁移服务之前，需要先格式化、分区和挂载磁盘。具体操作见以下链接，这里不再赘述。https://help.aliyun.com/document_detail/25426.html。

1.关闭 redis

#/etc/init.d/redis stop

2.拷贝数据目录(disk1 是数据盘)

#cp –a /usr/local/redis/db /disk1/redis/

3.编辑 redis 配置文件：

#vim /usr/local/redis/redis.conf

dir /disk1/redis/db

4.启动 redis

#/etc/init.d/redis start

【功能亮点】

（1）redis 属于高性能，支持丰富数据类型，主流的 key-value 数据库。
（2）python 多版本自由切换和管理。
（3）pip 升级至新版本，轻松管理 python 包。
（4）python 连接 mysql 轻松实现。
（5）ipython 便于高效开发和 debug。
（6）产品内置 redis 启停脚本，便于操作。
（7）内置 redis 缓存数据库，增强查询性能。
（8）对内核参数进行优化设置，进一步强化镜像安全性能。
（9）健全 sshd 服务的可靠性，充分适应整体需求。

【注意事项】

（1）系统首次初始化启动较慢，请耐心等待，如长时间连不上请进入阿里云管理控制台，远程管理终端查看状态。

（2）服务器管理员账户信息：Windows 系统远程桌面默认管理员账户为 administrator，Linux 系统 SSH 默认管理员账户为 root，默认密码为新购 ECS 或者初始化系统盘时所设置。

（3）开通本实例所需的安全组策略对应端口，操作方式如下：阿里云【控制台】——【ECS 云主机】——【实例】，点开对应实例之后找到【安全组】，【配置策略】上方菜单栏选择【公网入方向】，根据策略示例设置策略开放需要的端口。

（4）更多使用说明请参照 PDF 镜像使用指南。

【小提示】

（1）请及时更改数据库密码，并做好数据备份。
（2）如挂载数据盘，请迁移数据目录到数据盘中。

【相关权限】

（1）镜像中如有收费软件，请根据软件官方说明购买使用版权，因版权问题产生的纠纷本公司概不负责。

（2）镜像操作系统为公司定制，并经过反复测试验证，请参照商品详情中信息内容使用，除镜像本身默认环境问题，均不含任何人工技术支持。

（3）部分付费镜像有安全优化，但不保证服务器绝对安全，互联网中不存在绝对安全的服务器，请做好代码安全，并培养良好的使用习惯。

【售后支持范围】

关于售后服务：

（1）确保初始环境正常使用，如出现不能正常使用情况，请及时联系售后技术支持；如用户个人需要其他配置、调试修改、故障排查，请联系在线技术支持根据实际情况下单付费处理。

（2）如发现镜像存在安全漏洞，请及时联系售后技术支持免费修复漏洞。

（2）其他相关服务参照本公司服务类商品定价，下单后联系技术支持。

售后服务时间：工作日 9:00—12:00,13:00-18:00。

关于业务范围：

服务器环境配置，故障排查（不含程序自身问题），数据库配置更改，数据库权限、账户，数据迁移，程序迁移，数据库故障排查等。

费用参考：详情参照本公司服务类商品定价，或咨询在线技术支持。

关于更多技术细节，可参考【先知云】技术文章 http://bbs.cldera.com/forum-59-1.html。

【声明】

本镜像操作系统为公司技术人员细致研发定制，并经过反复测试验证，在基本设计和性能运行上已较为完善可靠。如需使用方法、其他故障修复等技术交流或支持，本公司将本着服务客户的态度热忱解决问题，但将酌情收取人工成本费用。