□Engineering Directive(ED) ■Engineering Specification (ES) □Manufacturing System (MS)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Subject: Environment setup SOP  Document classification: Confidential | | | Doc. No.: | | | Rev.:V001 | |
| Effective Date: 2023/02/27 | | | Revision Status | |
| Description:  Environment setup SOP | | | | | | Page | Rev. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Reason for Change: | | | | | | Reviewed by/ |  |
| ----------------- | ----------------- |
| Copy to(All contents): | | | | | | | | ----------------- | ----------------- |
|  | | | | | | | |
| Approved by/ | Reviewed by/ |  | |  |  | Prepared by/  Robbin Zheng | |
| ----------------- | ----------------- | | ----------------- | ----------------- |
| ----------------- | ----------------- | | ----------------- | ---------------- |

Notice: This cover sheet forms a part of the following directive (specification) and is not to be discarded unless superseded by a revised issue

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Comment** |
| V001 | 2023/02/27 | Robbin Zheng | Initial Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

[1 预设环境 4](#_Toc128729063)

[1.1 硬件信息 4](#_Toc128729064)

[1.2 安装CentOS 4](#_Toc128729065)

[1.3 开机自动网络连接 4](#_Toc128729066)

[1.4 手动编译安装Python 3.64 5](#_Toc128729067)

[1.4.1 安装epel release 5](#_Toc128729068)

[1.4.2 安装Development Tools 5](#_Toc128729069)

[1.4.3 编译Python3.64依赖库 5](#_Toc128729070)

[1.4.4 编译安装Python 3.64 6](#_Toc128729071)

[1.5 安装docker 6](#_Toc128729072)

[1.6 安装docker-compose 6](#_Toc128729073)

[1.7 拉取项目镜像 6](#_Toc128729074)

[1.8 配置防火墙 7](#_Toc128729075)

[1.9 关闭SELinux 7](#_Toc128729076)

[2 搭建系统 8](#_Toc128729077)

[2.1 创建操作系统用户 8](#_Toc128729078)

[2.2 布置发布的Web项目 8](#_Toc128729079)

[2.2.1 复制发布的项目 8](#_Toc128729080)

[3 注意事项： 9](#_Toc128729081)

## 预设环境

目前已经有一个硬盘空间有60G的KVM虚拟机CentOS7文件。

操作系统CentOS7超级用户信息：

user: root, password: 1234qwer!@#$QWER

### 硬件信息

具体需求搭不同配置。

### 安装CentOS

有无桌面没关系。可以安装系统服务器数据中心系统，或者最小配置系统的服务器。

CentOS的版本是7.9。

查看操作系统命令：

[root@localhost ~]# cat /etc/redhat-release

CentOS Linux release 7.9.2000 (Core)

基本的分区有/, /var, /home, /boot和swap。

分区需要注意：系统数据库会安装在/var，所以需要数据库的空间。

### 开机自动网络连接

1. 查看网卡信息

[root@localhost network-scripts]# ifconfig

eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 10.41.95.85 netmask 255.255.255.0 broadcast 10.41.95.255

ether 10:60:4b:92:14:c5 txqueuelen 1000 (Ethernet)

RX packets 109521808 bytes 32829266203 (30.5 GiB)

RX errors 0 dropped 313831 overruns 0 frame 0

TX packets 31471102 bytes 17457400420 (16.2 GiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

device interrupt 17

知道了网卡的名字是ens1。

1. 修改网卡的配置ONBOOT的值。

[root@AOI ~]# cd /etc/sysconfig/network-scripts/

把ifcfg-ens1文件中的ONBOOT的值改为yes

ONBOOT=yes

1. 重开机。

### 手动编译安装Python 3.64

目前使用的Python3版本源码包是Python-3.6.4.tgz。

下载路径：<https://www.python.org/downloads/release/python-364/>

#### 安装epel release

安装epel release，可以获取到比较新的软件包。

[root@localhost sites-enabled]# yum install epel-release

#### 安装Development Tools

安装开发环境的相关工具，可以提前安装编译Python3需要的工具，比如gcc。

[root@localhost sites-enabled]# yum groupinstall "Development Tools"

如果在安装Development Tools出现以下失败信息：

Maybe run yum groups mark install

No packages in any requested group availabe to install or update

安装失败解决步骤如下：

1. 先执行以下两条命令：

[root@localhost sites-enabled]# yum groups mark install "Development Tools"

[root@localhost sites-enabled]# yum groups mark convert "Development Tools"

1. 再安装Development Tools

[root@localhost sites-enabled]# yum groupinstall "Development Tools"

#### 编译Python3.64依赖库

必须的：

[root@localhost sites-enabled]# yum install python-devel libxml2-devel pcre openssl-devel zlib-devel bzip2-devel ncurses-devel xz-devel readline-devel tk-devel gdbm-devel db4-devel libcap-devel sqlite-devel

#### 编译安装Python 3.64

使用root用户进行这一步的操作。

1. 解压Python3压缩包

[root@localhost Python]# tar -xvf Python-3.6.4.tgz

1. 进入到解压目录

[root@localhost Python]# cd Python-3.6.4/

1. 创建Python3安装路径

[root@localhost Python-3.6.4]# mkdir /usr/python3.64

1. 执行配置

[root@localhost Python-3.6.4]# ./configure --prefix=/usr/python3.64 --enable-optimizations

1. 编译

[root@localhost Python-3.6.4]# make

1. 安装

[root@localhost Python-3.6.4]# make install

### 安装docker

在线下载：yum install docker-ce docker-ce-cli containerd.io

离线包下载路径：<https://download.docker.com/linux/static/stable>

离线包安装方式rpm -ivh 路径/文件名.rpm

离线包安装顺序①containerd.io②docker-ce-cli③docker-ce

安装完成后重新启动docker：systemctl start docker

设置开机自启动： systemctl enable docker

### 安装docker-compose

有python的情况下可以使用pip install –U docker-compose

离线下载包路径：curl -L "[https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)](https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname%20-s)-$(uname%20-m))" -o/usr/local/bin/docker-compose

添加执行权限：chmod +x /usr/local/bin/docker-compose

### 拉取项目镜像

1. 前往目录 /etc/docker/下 修改daemon.json文件内容添加

{

"insecure-registries":["reg.swharbor.com","10.41.95.93","harbor.wistron.com"]

}

加载配置并重启docker服务

systemctl daemon-reload

systemctl restart docker

1. 前往目录/etc 下，修改hosts文件

结尾处添加 10.41.95.93 reg.swharbor.com

1. 拉取生产者镜像：docker pull reg.swharbor.com/hugin-l11/iperf-producer:v1

拉取消费者镜像：docker pull reg.swharbor.com/hugin-l11/iperf-consumer:v1

拉取rabbitmq镜像：docker pull reg.swharbor.com/hugin-l11/iperf-rabbitmq:latest

拉取mac镜像：docker pull reg.swharbor.com/hugin-l11/query-ip-by-mac:v1

拉取redis镜像：docker pull reg.swharbor.com/hugin-l11/l11-redis:latest

### 配置防火墙

因为是web服务器，添加service http和https。

[root@localhost example]# firewall-cmd --permanent --zone=public --add-service=http

[root@localhost example]# firewall-cmd --permanent --zone=public --add-service=https

[root@localhost example]# firewall-cmd --reload

Firewall启动和关闭

systemctl start firewalld

systemctl stop firewalld

### 关闭SELinux

为了避免Nginx被阻止导致无妨正常使用，关闭SELinux。

(venv) [root@localhost example]# vim /etc/selinux/config

设置如下：

"/etc/selinux/config" 15L, 565C# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

#SELINUX=enforcing

SELINUX=disabled

# SELINUXTYPE= can take one of three two values:

# targeted - Targeted processes are protected,

# minimum - Modification of targeted policy. Only selected processes are prootected.

# mls - Multi Level Security protection.

SELINUXTYPE=targeted

(venv) [root@localhost example]# shutdown -r now

(venv) [root@localhost example]# sestatus

SELinux status: disabled

## 搭建系统

预设环境部分准备好了，就可以进行系统搭建了。

### 创建操作系统用户

需要给Web项目新增一个专有的操作系统用户及密码。

用户：iperf

密码：1234qwer!@#$QWER

需要在root用户下才能新增成功，命令如下：

[root@localhost example]# useradd iperf

[root@localhost example]# passwd 1234qwer!@#$QWER

切换用户到iperf

[iperf@localhost example]# su iperf

将用户添加到docker用户组：

sudo gpasswd –a ${USER} docker

sudo service docker restart

newgrp docker

### 布置发布的Web项目

#### 复制发布的项目

发布的iperf项目目录是iperf,把整个目录文件拷到项目目录/home/iperf/下。更推荐使用git：

* 1. 创建目录/home/iperf，并且到该目录下；
  2. 拉取消费者:git clone ssh://git@10.41.95.141:222/esrz30/consumer\_script.git
  3. 拉取生产者:git clone ssh://git@10.41.95.141:222/esrz30/hugin-l11-producer-api.git
  4. 从mac获取信息:git clone ssh://git@10.41.95.141:222/esrz30/query-information-by-mac.git

通过docker-compose启动项目

docker-compose放于git的consumer仓库中

去到目录下的docker文件夹内/home/iperf/consumer\_script

docker-compose up –d

## 注意事项：

1. 该项目目录中，如果项目中consumer\_script,hugin-l11-producer-api,query information by mac文件不是放在同一个目录中，则需要修改consumer\_script下docker-compose文件中对应部分的文件路径挂载。使其与当前目录相对应才可正常运行。
2. Iperf中关于连接L11数据库的部分。生产者：/home/iperf/hugin-l11-producer-api/utils/config.py 和消费者：/home/iperf/consumer\_script/consumer\_script\_sql.py 中连接到L11数据库的部分根据实际需求需要更改对应的ip。

生产者：





消费者：

