在OCI VM上安装Magento2

2021-11

目录

[1. 前言 3](#_Toc87975145)

[2. 环境装备 3](#_Toc87975146)

[2.1. Magento 架构 3](#_Toc87975147)

[2.2. 环境装备 4](#_Toc87975148)

[3. 安装MySQL 5](#_Toc87975149)

[4. 安装php 6](#_Toc87975150)

[5. 安装Elasticsearch 7](#_Toc87975151)

[6. 安装magento 8](#_Toc87975152)

[7. 安装ngnix 10](#_Toc87975153)

[8. 访问magento 13](#_Toc87975154)

[9. 添加magento示例数据 14](#_Toc87975155)

[10. 生成magento测试数据 17](#_Toc87975156)

[11. 安装配置redis缓存 19](#_Toc87975157)

[11.1. 安装Redis 19](#_Toc87975158)

[11.2. 配置Magento使用Redis 20](#_Toc87975159)

[12. 安装配置Memcached Session缓存 21](#_Toc87975160)

[13. 安装配置RabbitMQ 23](#_Toc87975161)

# 前言

Magento是一个用于电子商务网站的开源内容管理系统（CMS）平台。它使商家可以添加最佳的购物车，自定义产品页面以及出色的网站主题和设计。Magento可以免费使用庞大的社区提供支持。

Magento是一个基于PHP的平台，可同时提供出色的功能和功能。超过 25万个网站使用Magento电子商务平台，该平台成为2020年第二大最受欢迎的电子商务平台，全球电子商务市场整体份额为12％。

Magento平台在CMS市场中的全球份额占全球1.9％。在这个竞争激烈的市场中，这是一个非常体面的数字。Magento平台可完全控制大量产品和多个优惠的定制。 企业版的Magento价格（由对客户的全面备份支持提供支持）也非常合理。根据电子商务评论，Magento是市场上最好的平台之一。请务必注意，从2013年到现在，Magento连续四年被评为好的平台。

本文介绍在oracle Cloud VM环境中如何安装和配置Magento网站，并使用示例数据测试网站内容。

# 环境装备

## Magento 架构

Magento 使用LAMP架构，安装过程中，其中PHP，MySQL，Elasticsearch是必须的安装的，而Redis，RabbitMQ是可选组件。

图示

描述已自动生成

## 环境装备

Magento 2系统需求：

* 操作系统 (Linux x86-64)

Linux发行版如红帽企业Linux（RHEL），CentOS，Ubuntu，Debian，等等

* 内存需求

升级的应用程序和扩展你的Magento从Magento市场和其他来源获得需要多达2GB的RAM。如果您使用少于2GB内存的系统，我们建议您创建一个交换文件；否则，您的升级可能会失败。

* Composer

最新的稳定版本

* web服务器

Apache 2.2 or 2.4

nginx 1.8 (或最新稳定版本)

* 数据库

MySQL 5.7兼容。

* PHP

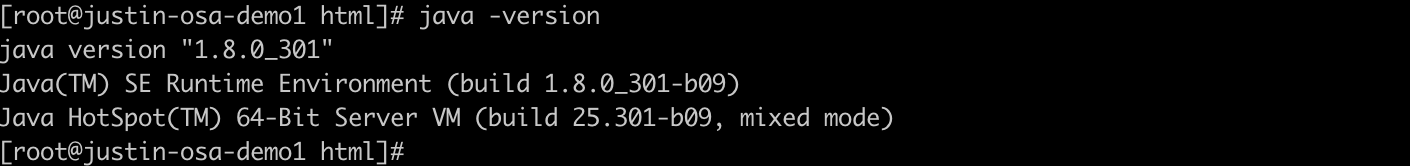
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7.0.0, 7.0.1 | 7.0.2 | 7.0.3 | 7.0.4 | 7.0.5 | 7.0.6–7.0.x | 7.1.x |
| Magento 2.1.x 系统需求 | Magento 2.1.x 系统需求 | Magento 2.1.x 系统需求 | Magento 2.1.x 系统需求 | Magento 2.1.x 系统需求 | Magento 2.1.x 系统需求 | Magento 2.1.x 系统需求 |

* Elasticsearch

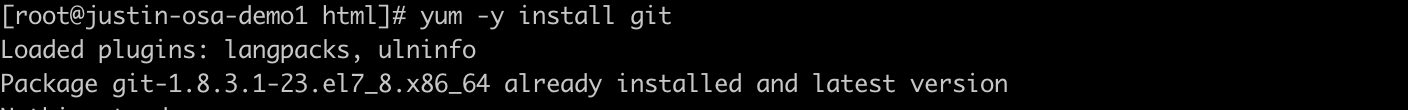
安装版本opendistroforelasticsearch-1.13.2

* Redis(可选)
* RabbitMQ（可选）

Java 1.8+



yum -y install git



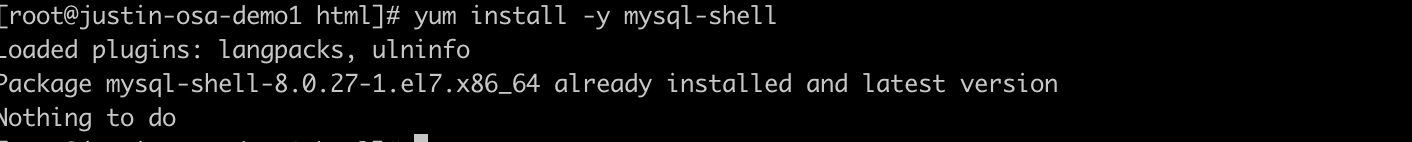
yum -y install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

# 安装MySQL

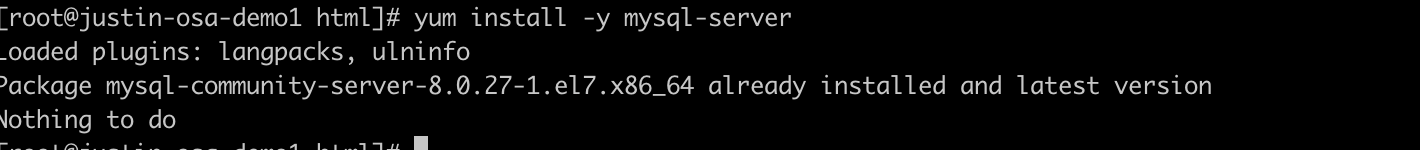
# Install MySQL Community Edition 8.0

rpm -ivh <https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm>

yum install -y mysql-shell



yum install -y mysql-server



# 安装php

yum -y install https://rpms.remirepo.net/enterprise/remi-release-7.rpm



yum-config-manager --enable remi-php74

文本

描述已自动生成

yum -y install php php-cli php-mysqlnd php-zip php-gd php-mcrypt php-mbstring php-xml php-json php-opcache php-bcmath php-soap php-pear-Net-Socket php-intl

文本

描述已自动生成

yum -y install certbot mod\_ssl

文本

描述已自动生成

需要修改php.ini的配置文件，设置如下操作：

#Vi /etc/php.ini

memory\_limit = 2G

max\_execution\_time = 1800

zlib.output\_compression = On

cgi.fix\_pathinfo=0

session.save\_path = "/var/lib/php/session"

# 安装Elasticsearch

1）获取yum repo

curl [https://d3g5vo6xdbdb9a.cloudfront.net/yum/opendistroforelasticsearch-artifacts.repo -o /etc/yum.repos.d/opendistroforelasticsearch-artifacts.repo](https://d3g5vo6xdbdb9a.cloudfront.net/yum/opendistroforelasticsearch-artifacts.repo%20-o%20/etc/yum.repos.d/opendistroforelasticsearch-artifacts.repo)

2）安装elasticsearch

yum -y install opendistroforelasticsearch

3）修改配置文件：

Vi /etc/elasticsearch/elasticsearch.yml

node.name: node-1

path.data: /var/lib/elasticsearch

path.logs: /var/log/elasticsearch

network.host: 0.0.0.0

cluster.initial\_master\_nodes: ["node-1"]

discovery.seed\_hosts: ["node-1"]

opendistro\_security.disabled: true

4）启动elasticsearch服务

systemctl restart elasticsearch.service

5) 关闭ES性能监控

# echo false | sudo tee /usr/share/elasticsearch/data/batch\_metrics\_enabled.conf

# 安装magento

1. 获取安装文件

# wget <https://github.com/magento/magento2/archive/$magento_version.tar.gz>

# tar zxvf $magento\_version.tar.gz

# rm -rf html/ $magento\_version.tar.gz

# mv magento2-\* /var/www/html

1. 配置http，开放访问

# sed -i '/AllowOverride None/c\AllowOverride All' /etc/httpd/conf/httpd.conf

1. 安装composer

# cd /usr/local/bin

# wget <https://getcomposer.org/composer-1.phar>

# chmod +x composer-1.phar

# mv composer-1.phar composer

1. 使用composer 安装相关库

# cd /var/www/html

# /usr/local/bin/composer install

图形用户界面, 文本

描述已自动生成

1. 创建magento数据库和用户

#mysqlsh --user ${admin\_username} --password=${admin\_password} --host ${mds\_ip} --sql -e "CREATE DATABASE $magentoschema;"

#mysqlsh --user ${admin\_username} --password=${admin\_password} --host ${mds\_ip} --sql -e "CREATE USER $magentoname identified by '${magento\_password}';"

#mysqlsh --user ${admin\_username} --password=${admin\_password} --host ${mds\_ip} --sql -e "GRANT ALL PRIVILEGES ON $magentoschema.\* TO $magentoname;"

1. 部署配置magento

#cd /var/www/html

#bin/magento setup:install \

--base-url=http://test.magetotest.com/ \

--db-host={mds\_ip} \

--db-name=magentodb \

--db-user= ${admin\_username} \

--db-password=${magento\_password} \

--admin-firstname=admin \

--admin-lastname=admin \

--admin-email=admin@admin.com \

--admin-user=admin \

--admin-password=admin \

--language=en\_US \

--currency=USD \

--timezone=America/Chicago \

--use-rewrites=1

图片包含 形状

描述已自动生成

文本

描述已自动生成

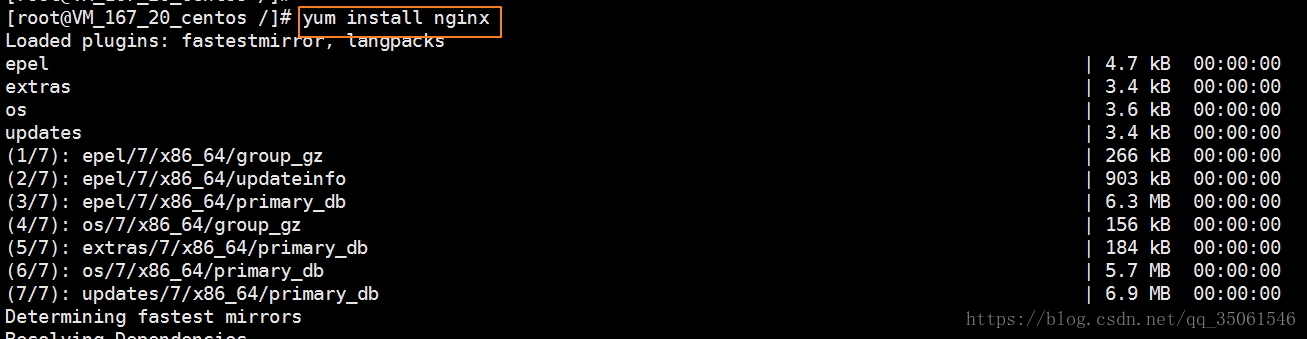
# cd /var/www

# chown apache. -R html

# 安装ngnix

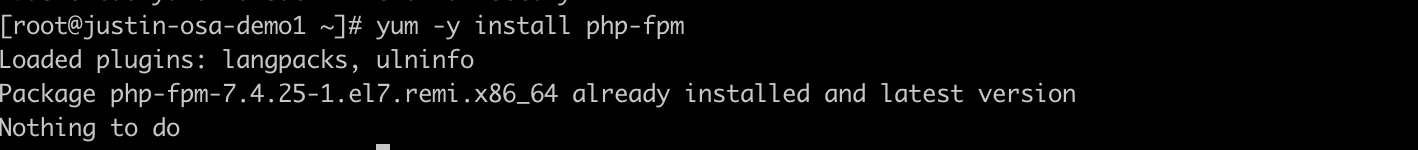
1. 安装nginx

链接上服务器后执行 yum -y install nginx



1. 安装php-fpm

# yum -y install php-fpm



3）配置php-fpm

# vim /etc/php-fpm.d/www.conf

user = apache

group = apache

listen = /run/php-fpm/php-fpm.sock

listen.owner = apache

listen.group = apache

listen.mode = 0660

取消注释环境线：

env[HOSTNAME] = $HOSTNAME

env[PATH] = /usr/local/bin:/usr/bin:/bin

env[TMP] = /tmp

env[TMPDIR] = /tmp

env[TEMP] = /tmp

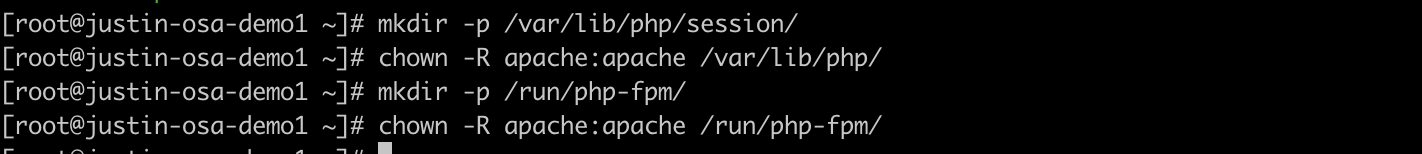
创建相关目录和修改目前权限：

mkdir -p /var/lib/php/session/

chown -R apache:apache /var/lib/php/

mkdir -p /run/php-fpm/

chown -R apache:apache /run/php-fpm/



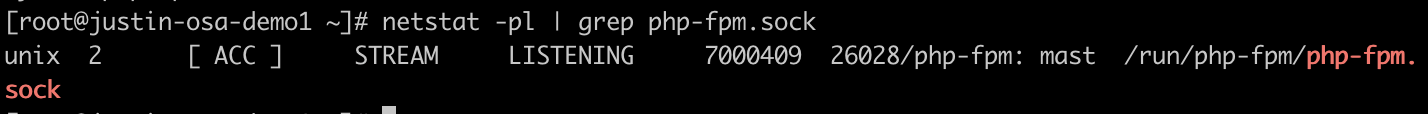
启动该php-fpm服务并将其配置为在引导时启动：

systemctl start php-fpm

systemctl enable php-fpm

验证php-fpm服务是否正在运行：如果输入啥反应都没有，就表示不在运行，有显示就表示运行成功。

netstat -pl | grep php-fpm.sock



4）配置ngnix

Vi /etc/nginx/conf.d/magento.conf

upstream fastcgi\_backend {

server unix:/run/php-fpm/php-fpm.sock;

}

server {

listen 80;

server\_name test.magetotest.com;

set $MAGE\_ROOT /var/www/html;

set $MAGE\_MODE default;

include /var/www/html/nginx.conf.sample;

}

启动该nginx服务并将其配置为在引导时启动：

systemctl start nginx

systemctl enable nginx

文本

描述已自动生成

# 访问magento

启动nginx后，就可以直接访问到magento提的web页面了。默认这里面没有商品数据。

http://test.magetotest.com/

图形用户界面, 文本, 应用程序

描述已自动生成

管理页面地址：

注意，安装完成后，会提示管理地址的URI。登陆用户名和密码是部署配置时确定的用户名和密码，登陆后密码可以修改。

图形用户界面

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

# 添加magento示例数据

$ git clone <https://github.com/magento/magento2-sample-data.git>

文本

描述已自动生成

$ sudo php -f magento2-sample-data/dev/tools/build-sample-data.php -- --ce-source="/var/www/html"

$ cd magento2-sample-data/

$ sudo chown -R :apache .

$ sudo find . -type d -exec chmod g+ws {} +

# cd /var/www/html

# php bin/magento setup:upgrade

文本

描述已自动生成

# chown -R apache. \*

# rm -rf var/cache/\* var/page\_cache/\* generated/\*

重新刷新访问magento网站，可以看到示例的商品数据已经添加成功。

http://test.magetotest.com/

图形用户界面, 网站

描述已自动生成

管理页面中，能看到销售的产品信息。

图形用户界面, 应用程序

描述已自动生成

查看mangento网站访问日志

#cd /var/log/nginx

文本

描述已自动生成

#卸载测试数据

$ sudo php -f magento2-sample-data/dev/tools/build-sample-data.php – --command=unlink --ce-source="/var/www/html"

# 生成magento测试数据

Magento可以支持自动进行测试数据的生成，具体的测试数据范围如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **Small profile** | **Medium profile** | **Large profile** | **Extra large profile** |
| websites | 1 | 1 | 3 | 5 |
| store\_groups | 1 | 2 | 3 | 5 |
| store\_views | 1 | 2 | 3 | 5 |
| simple\_products | 800 | 16,000 | 400,000 | 800,000 |
| configurable\_products | 50 | 1,000 | 25,000 | 50,000 |
| categories | 30 | 300 | 1,000 | 3,000 |
| categories\_nesting\_level | 3 | 3 | 3 | 6 |
| catalog\_price\_rules | 10 | 20 | 50 | 100 |
| catalog\_target\_rules | 2 | 5 | 10 | 50 |
| cart\_price\_rules | 10 | 20 | 50 | 100 |
| cart\_price\_rules\_floor | 2 | 2 | 2 | 5 |
| customers | 20 | 200 | 2,000 | 5,000 |
| tax rates | 40,000 | 40,000 | 40,000 | 40,000 |
| orders | 80 | 1,600 | 40,000 | 80,000 |

Mysql 设置 SET GLOBAL log\_bin\_trust\_function\_creators = 1;

文本

描述已自动生成

# cd /var/www/html

# bin/magento setup:perf:generate-fixtures /var/www/html/setup/performance-toolkit/profiles/ce/small.xml

文本

描述已自动生成

# chown -R apache. \*

# 安装配置redis缓存

## 安装Redis

Redis安装参考：<https://www.digitalocean.com/community/tutorials/how-to-install-secure-redis-centos-7>

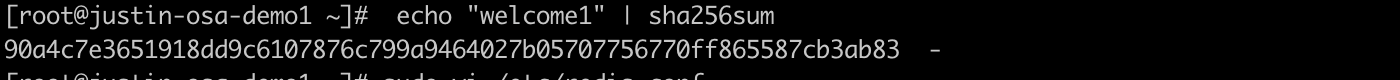
$ sudo yum install epel-release

$ sudo yum install redis -y

文本

描述已自动生成

$ echo "welcome1" | sha256sum



$ sudo vi /etc/redis.conf

图形用户界面, 文本, 电子邮件, 网站

描述已自动生成

$ sudo systemctl restart redis.service

$ redis-cli

文本

描述已自动生成

## 配置Magento使用Redis

以下示例启用 Redis 默认缓存，将主机设置为 127.0.0.1 并将数据库编号分配为 0。Redis 对所有其他参数使用默认值。

1. Configure Redis default caching

# bin/magento setup:config:set --cache-backend=redis --cache-backend-redis-server=127.0.0.1 --cache-backend-redis-db=0 --cache-backend-redis-password=90a4c7e3651918dd9c6107876c799a9464027b05707756770ff865587cb3ab83

文本

描述已自动生成

1. Configure Redis page caching

#bin/magento setup:config:set --page-cache=redis --page-cache-redis-server=127.0.0.1 --page-cache-redis-db=1 --page-cache-redis-password=90a4c7e3651918dd9c6107876c799a9464027b05707756770ff865587cb3ab83文本

描述已自动生成

# 安装配置Memcached Session缓存

yum -y update

yum install -y libevent libevent-devel

yum install -y memcached

yum install -y php-pecl-memcache

文本

描述已自动生成

# vi /etc/sysconfig/memcached

CACHESIZE="1GB"

OPTIONS="127.0.0.1"

# service memcached restart

文本

描述已自动生成

#vi /var/www/html/app/etc/env.php

修改



为

array (

'save' => 'memcached',

'save\_path' => '127.0.0.1:11211'

),

图片包含 形状

描述已自动生成

删除session缓存

# rm -rf var/cache/\* var/page\_cache/\* var/session/\*

查看 Cache情况

图形用户界面, 应用程序

描述已自动生成

# 安装配置RabbitMQ

|  |
| --- |
| # rpm --import https://www.rabbitmq.com/rabbitmq-signing-key-public.asc |

# yum -y install rabbitmq-server

文本

描述已自动生成

# vi app/etc/env.php

修改增加下面的内容

文本

低可信度描述已自动生成

重新update

#bin/magento setup:upgrade

#chown -R apache. \*