**快速部署rabbitMQ教程**

**来源：**

**https://yq.aliyun.com/articles/92665?spm=5176.10695662.1996646101.searchclickresult.51672b57UJzYB8**

[云市场转载](https://yq.aliyun.com/users/1206489547028373) 2017-06-01 15:18:53 浏览17978 评论0

* [linux](https://yq.aliyun.com/tags/type_blog-tagid_10/)

* [erlang](https://yq.aliyun.com/tags/type_blog-tagid_646/)

* [镜像](https://yq.aliyun.com/tags/type_blog-tagid_713/)

* [教程](https://yq.aliyun.com/tags/type_blog-tagid_991/)

* [rabbitmq](https://yq.aliyun.com/tags/type_blog-tagid_1401/)

* [云市场](https://yq.aliyun.com/tags/type_blog-tagid_1446/)

* [部署](https://yq.aliyun.com/tags/type_blog-tagid_3616/)

**摘要：** 今天教大家利用镜像快熟部署使用RabbitMQ。RabbitMQ是一个开源的AMQP实现，服务器端用Erlang语言编写，支持多种客户端，用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。

RabbitMQ是一个开源的AMQP实现，服务器端用Erlang语言编写，支持多种客户端，如：Python、Ruby、.NET、Java、JMS、C、PHP、ActionScript、XMPP、STOMP等，支持AJAX。用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。

**部署方式**

在阿里云服务器下部署 RabbitMQ提供两种部署方式：

* RabbitMQ镜像部署
* 手动部署（源码编译安装/YUM安装）

一般推荐镜像部署适合新手使用更加快捷方便，安装包部署以及手动部署适合对Linux命令有基本了解的用户，可以满足用户个性化部署的要求。本教程主要介绍镜像和手工部署的方式。

**镜像部署**

1. 单击 **JAVA环境（CentOS7.2 Nginx Tomcat8 JDK）**进入镜像详情页。

[JAVA环境（CentOS7.3 Nginx Tomcat8 JDK）](https://market.aliyun.com/products/53400005/cmjj016483.html)



镜像集成CentOS7.4+Nginx+MySQL5.7+Tomcat8，Nginx处理静态资源，Tomcat以apr模式运行处理动态资源，大幅度的提高性能

￥0/月

[立即购买](https://market.aliyun.com/products/53400005/cmjj016483.html)

2. 单击 **\*\*立即购买\*\***，按提示步骤购买 ECS 实例。

3. 登录 **ECS 管理控制台**：<https://ecs.console.aliyun.com/#/home>

4. 在左边导航栏里，单击 **\*\*实例\*\***，进入 ECS 实例列表页。

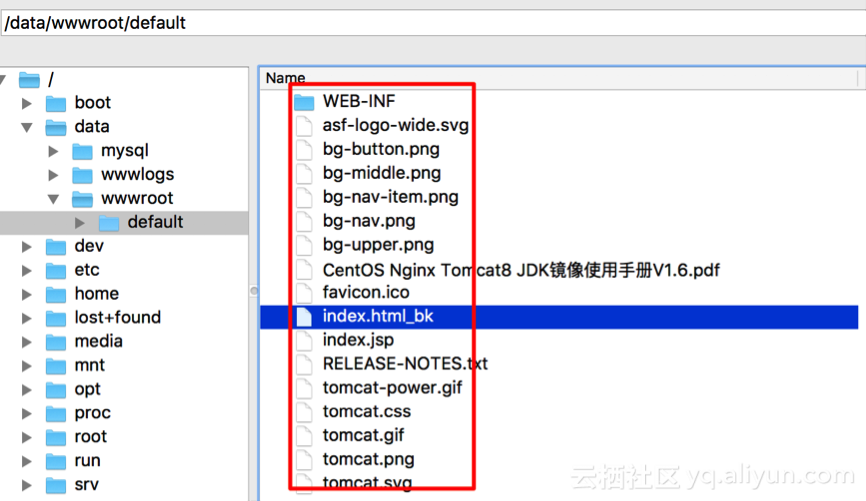
5. 选择所购 ECS 实例所在的地域，并找到所购 ECS 实例，在 **\*\*IP 地址\*\*** 列获取该实例的公网 IP 地址。

6. 在浏览器地址栏中输入公网 IP 地址，下载操作文档。



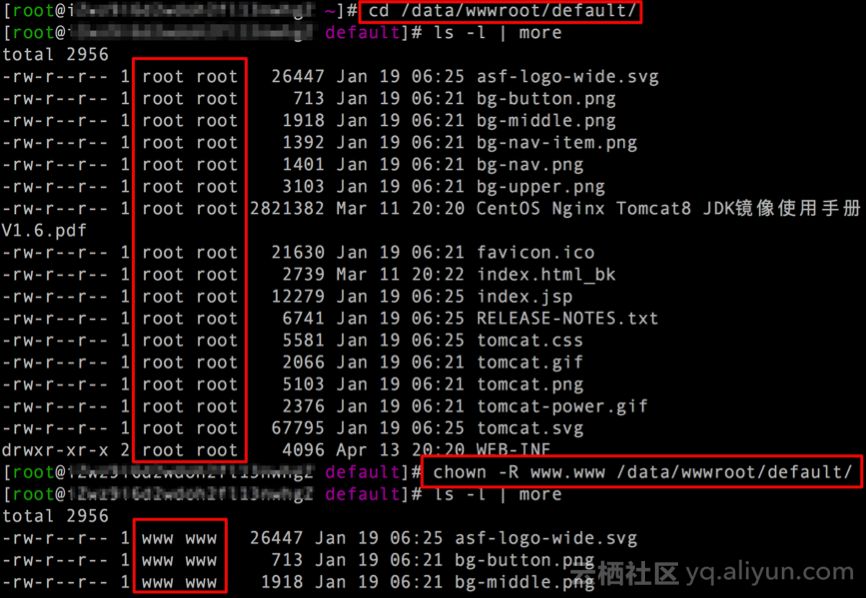
7. 使用putty登录Linux服务器，参考《[连接Linux实例](https://help.aliyun.com/document_detail/25434.html)》；忘记root密码参考《[重置实例密码](https://help.aliyun.com/document_detail/25439.html)》。

8. 使用winscp工具将Java代码放入**/data/wwwroot/default**中。



9. 默认tomcat是以一般www用户运行，将网站代码权限改为www，执行下面命令：

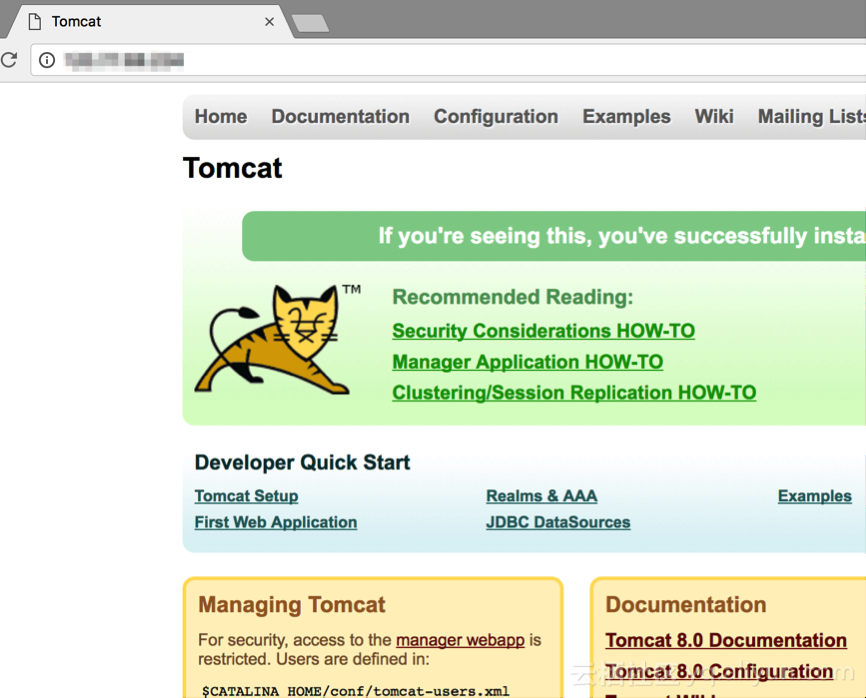
chown -R www.www /data/wwwroot



10. 重启tomcat



11. 在浏览器地址栏中输入公网 IP 地址，验证。



**手工部署**

**系统平台：**CentOS 7.3

**rabbitmq版本：**rabbitmq-server -3.6.9

**erlang版本：**erlang19.3

**JDK版本：**JDK1.8.0\_121

* **前提准备**

创建一般用户rabbitmq，运行rabbitmq

useradd rabbitmq

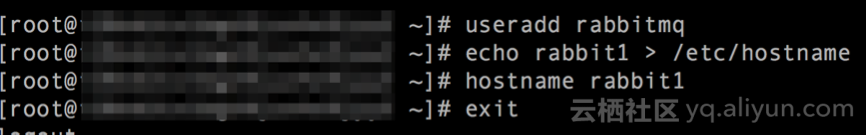
设置Linux主机名

centos7修改**/etc/hostname**，centos6修改**/etc/sysconfig/network**，下面以centos7为例

echo rabbit1 > /etc/hostname

hostname rabbit1

exit #退出重新登录



安装依赖包

yum -y install make gcc gcc-c++ m4 ncurses-devel openssl-devel unixODBC-devel

快速部署rabbitMQ教程

* **源代码下载**

wget http://erlang.org/download/otp\_src\_19.3.tar.gz

wget https://www.rabbitmq.com/releases/rabbitmq-server/v3.6.9/rabbitmq-server-generic-unix-3.6.9.tar.xz

* **安装Erlang**

tar xzf otp\_src\_19.3.tar.gz #解压

cd otp\_src\_19.3

./configure --prefix=/usr/local/erlang --enable-shared-zlib --with-ssl --enable-threads --enable-smp-support --enable-kernel-poll --enable-hipe --without-javac

make && make install

**./configure时报错：**

**QA1.configure: error: No curses library functions found**

**缺少ncurses安装包**

yum -y install ncurses-devel

QA2. crypto         : No usable OpenSSL found

QA3. odbc : ODBC library - link check failed

QA4. jinterface : No Java compiler found

这里主要注意 APPLICATIONS DISABLED 部分的提示,其他两部分是不影响编译的。

jinterface : No Java compiler found 什么？没有java编译器？其实这里我们可以选择用gcc等其他方式来编译erlang。

如果你安装了gcc，这里就用不上java编译了，因此可以在configure时增加 –disable-javac 或者--without-javac 避免第二个错误；

./configure --prefix=/usr/local/erlang --enable-shared-zlib --with-ssl --enable-threads --enable-smp-support --enable-kernel-poll --enable-hipe --without-javac -disable-javac

* **解压RabbitMQ**

1. 解压**rabbitmq-server-generic-unix-3.6.9.tar.xz**

tar xvJf rabbitmq-server-generic-unix-3.6.9.tar.xz

mv rabbitmq\_server-3.6.9 /usr/local/rabbitmq

2. rabbitmq环境变量配置

sed -i 's@^ERL\_DIR=.\*@ERL\_DIR=/usr/local/erlang/bin/@' /usr/local/rabbitmq/sbin/rabbitmq-defaults

sed -i 's@^LOG\_BASE=.\*@LOG\_BASE=/usr/local/rabbitmq/var/log/rabbitmq@' /usr/local/rabbitmq/sbin/rabbitmq-defaults

mkdir -p /usr/local/rabbitmq/var/{lib,log}/rabbitmq

3. 一般用户（rabbitmq）运行RabbitMQ

wget http://pkgs.fedoraproject.org/cgit/rpms/rabbitmq-server.git/plain/rabbitmq-script-wrapper

sed -i 's@cd /var/lib/rabbitmq@cd /usr/local/rabbitmq/var/lib/rabbitmq@g' rabbitmq-script-wrapper #更改rabbitmq数据存储目录

sed -i 's@/usr/lib/rabbitmq/bin/@/usr/local/rabbitmq/sbin/@g' rabbitmq-script-wrapper

chmod +x rabbitmq-script-wrapper

cp rabbitmq-script-wrapper /usr/sbin/rabbitmqctl

cp rabbitmq-script-wrapper /usr/sbin/rabbitmq-server

cp rabbitmq-script-wrapper /usr/sbin/rabbitmq-plugins

chown -R rabbitmq.rabbitmq /usr/local/rabbitmq/var

4. rabbitmq日志割接

cat >> /etc/logrotate.d/rabbitmq-server << EOF

/usr/local/rabbitmq/var/log/rabbitmq/\*.log {

weekly

missingok

rotate 20

compress

delaycompress

notifempty

sharedscripts

postrotate

/sbin/service rabbitmq-server rotate-logs > /dev/null

endscript

}

EOF

5. rabbitmq启动脚本

vi /etc/init.d/rabbitmq-server

#!/bin/sh

#

# rabbitmq-server RabbitMQ broker

#

# chkconfig: - 80 05

# description: Enable AMQP service provided by RabbitMQ

#

### BEGIN INIT INFO

# Provides: rabbitmq-server

# Required-Start: $remote\_fs $network

# Required-Stop: $remote\_fs $network

# Description: RabbitMQ broker

# Short-Description: Enable AMQP service provided by RabbitMQ broker

### END INIT INFO

# Source function library.

. /etc/init.d/functions

PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/erlang/bin

NAME=rabbitmq-server

DAEMON=/usr/sbin/${NAME}

CONTROL=/usr/sbin/rabbitmqctl

DESC=rabbitmq-server

USER=rabbitmq

ROTATE\_SUFFIX=

INIT\_LOG\_DIR=/usr/local/rabbitmq/var/log/rabbitmq

PID\_FILE=/var/run/rabbitmq/pid

START\_PROG="daemon"

LOCK\_FILE=/var/lock/subsys/$NAME

test -x $DAEMON || exit 0

test -x $CONTROL || exit 0

RETVAL=0

set -e

[ -f /etc/default/${NAME} ] && . /etc/default/${NAME}

[ -f /etc/sysconfig/${NAME} ] && . /etc/sysconfig/${NAME}

ensure\_pid\_dir () {

PID\_DIR=`dirname ${PID\_FILE}`

if [ ! -d ${PID\_DIR} ] ; then

mkdir -p ${PID\_DIR}

chown -R ${USER}:${USER} ${PID\_DIR}

chmod 755 ${PID\_DIR}

fi

}

remove\_pid () {

rm -f ${PID\_FILE}

rmdir `dirname ${PID\_FILE}` || :

}

start\_rabbitmq () {

status\_rabbitmq quiet

if [ $RETVAL = 0 ] ; then

echo RabbitMQ is currently running

else

RETVAL=0

# RABBIT\_NOFILES\_LIMIT from /etc/sysconfig/rabbitmq-server is not handled

# automatically

if [ "$RABBITMQ\_NOFILES\_LIMIT" ]; then

ulimit -n $RABBITMQ\_NOFILES\_LIMIT

fi

ensure\_pid\_dir

set +e

RABBITMQ\_PID\_FILE=$PID\_FILE $START\_PROG $DAEMON \

> "${INIT\_LOG\_DIR}/startup\_log" \

2> "${INIT\_LOG\_DIR}/startup\_err" \

0<&- &

$CONTROL wait $PID\_FILE >/dev/null 2>&1

RETVAL=$?

set -e

case "$RETVAL" in

0)

echo SUCCESS

if [ -n "$LOCK\_FILE" ] ; then

touch $LOCK\_FILE

fi

;;

\*)

remove\_pid

echo FAILED - check ${INIT\_LOG\_DIR}/startup\_\{log, \_err\}

RETVAL=1

;;

esac

fi

}

stop\_rabbitmq () {

status\_rabbitmq quiet

if [ $RETVAL = 0 ] ; then

set +e

$CONTROL stop ${PID\_FILE} > ${INIT\_LOG\_DIR}/shutdown\_log 2> ${INIT\_LOG\_DIR}/shutdown\_err

RETVAL=$?

set -e

if [ $RETVAL = 0 ] ; then

remove\_pid

if [ -n "$LOCK\_FILE" ] ; then

rm -f $LOCK\_FILE

fi

else

echo FAILED - check ${INIT\_LOG\_DIR}/shutdown\_log, \_err

fi

else

echo RabbitMQ is not running

RETVAL=0

fi

}

status\_rabbitmq() {

set +e

if [ "$1" != "quiet" ] ; then

$CONTROL status 2>&1

else

$CONTROL status > /dev/null 2>&1

fi

if [ $? != 0 ] ; then

RETVAL=3

fi

set -e

}

rotate\_logs\_rabbitmq() {

set +e

$CONTROL rotate\_logs ${ROTATE\_SUFFIX}

if [ $? != 0 ] ; then

RETVAL=1

fi

set -e

}

restart\_running\_rabbitmq () {

status\_rabbitmq quiet

if [ $RETVAL = 0 ] ; then

restart\_rabbitmq

else

echo RabbitMQ is not runnning

RETVAL=0

fi

}

restart\_rabbitmq() {

stop\_rabbitmq

start\_rabbitmq

}

case "$1" in

start)

echo -n "Starting $DESC: "

start\_rabbitmq

echo "$NAME."

;;

stop)

echo -n "Stopping $DESC: "

stop\_rabbitmq

echo "$NAME."

;;

status)

status\_rabbitmq

;;

rotate-logs)

echo -n "Rotating log files for $DESC: "

rotate\_logs\_rabbitmq

;;

force-reload|reload|restart)

echo -n "Restarting $DESC: "

restart\_rabbitmq

echo "$NAME."

;;

try-restart)

echo -n "Restarting $DESC: "

restart\_running\_rabbitmq

echo "$NAME."

;;

\*)

echo "Usage: $0 {start|stop|status|rotate-logs|restart|condrestart|try-restart|reload|force-reload}" >&2

RETVAL=1

;;

esac

exit $RETVAL

保存后，添加执行权限，并设置自启动

chmod +x /etc/init.d/rabbitmq-server

chkconfig --add rabbitmq-server

chkconfig rabbitmq-server on

6. 修改rabbitmq.config

特别注意默认用户名密码，请自行修改**default\_user，default\_pass，loopback\_users**

cat > /usr/local/rabbitmq/etc/rabbitmq/rabbitmq.config << EOF

[

{rabbit, [

{tcp\_listeners,[{"0.0.0.0",5672}]},

{tcp\_listen\_options, [binary, {packet,raw},

{reuseaddr,true},

{backlog,128},

{nodelay,true},

{exit\_on\_close,false},

{keepalive,true}]},

{default\_vhost, <<"/">>},

{default\_user, <<"guest">>},

{default\_pass, <<"guest">>},

{loopback\_users, ["guest"]},

{default\_permissions, [<<".\*">>, <<".\*">>, <<".\*">>]}

]}

].

EOF

7. 开启rabbitmq manager

cat > /usr/local/rabbitmq/etc/rabbitmq/enabled\_plugins << EOF

[rabbitmq\_management].

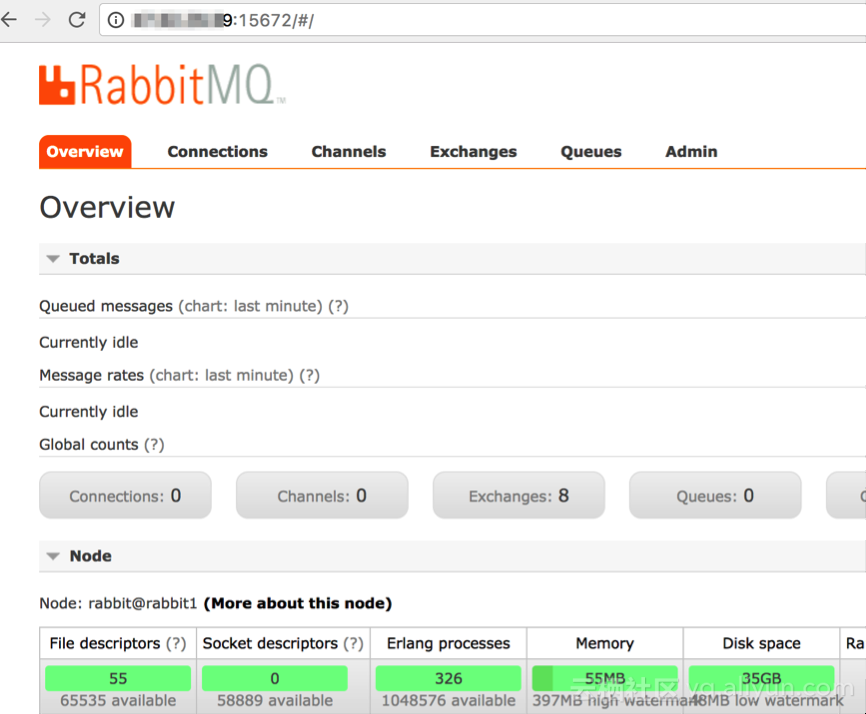
EOF

8. 启动rabbitmq

service rabbitmq-server start

9. 进入管理页面

浏览器访问**http://公网IP:15672**



**更多基础教程：**

[云市场头条基础教程频道](https://yq.aliyun.com/teams/61/type_blog-cid_156)

**更多开源软件尽在云市场：**

<https://market.aliyun.com/software>

rabbitmq简单启动方式：

/usr/local/rabbitmq/sbin目录下：

1.start: ./rabbitmq-server &

2.stop: ./rabbitmqctl stop

3.启动管理页面：

./rabbitmqctl start\_app

./rabbitmq-plugins enable rabbitmq\_management

如下提示表示管理页面启动成功：

