

Elastic Compute Service

Tutorials

Tutorials

Deploy LNMP

Build LNMP environment under CentOS 6

This article describes how to build LNMP environment under CentOS on an ECS instance with the basic configuration.

LNMP means:

- Linux: A family of free and open-source UNIX-like software operating systems (OS).
- Nginx: A lightweight HTTP and reverse proxy server.
- MySQL: A relational database management system.
- PHP: A scripting language that is especially suited for web development.

This method is applicable to individual users who are familiar with Linux, but new to website construction by using Alibaba Cloud ECS.

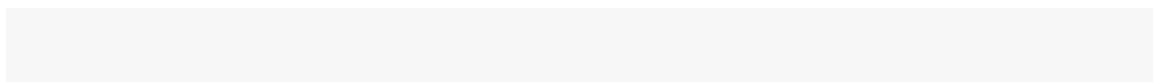
Follow these steps to build LNMP environment on an ECS instance:

1. Prepare the compiling environment
2. Install Nginx
3. Install MySQL
4. Install PHP-FPM
5. Test

Step 1. Prepare the compiling environment

Follow these steps to prepare the compiling environment.

Check the version of the operating system.



```
# cat /etc/redhat-release
CentOS release 6.5 (Final)
```

Note: This article is based on a Linux instance running CentOS 6.5. You may have different OS versions. The same is applicable to the Nginx, MySQL, and PHP versions mentioned in the following paragraphs.

Disable SELINUX: Run the command to modify the configuration file, which will permanently take effect after restarting the service.

```
# sed -i 's/SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
```

Run the command to make the configuration take effect immediately.

```
# setenforce 0
```

Add a security rule to accept Internet access to the Web server on the instance. For more information, see [Add a security group rule](#).

Step 2: Install Nginx

Nginx is a small and highly-efficient Web server based on Linux. Follow these steps to install Nginx:

Add a user to run the Nginx service process.

```
# groupadd -r nginx
# useradd -r -g nginx nginx
```

Download the source code package, decompress it, and then compile.

```
# wget http://nginx.org/download/nginx-1.10.2.tar.gz
# tar xvf nginx-1.10.2.tar.gz -C /usr/local/src
# yum groupinstall "Development tools"
# yum -y install gcc wget gcc-c++ automake autoconf libtool libxml2-devel libxslt-devel perl-devel perl-ExtUtils-Embed pcre-devel openssl-devel
# cd /usr/local/src/nginx-1.10.2
# ./configure \
--prefix=/usr/local/nginx \
--sbin-path=/usr/sbin/nginx \
--conf-path=/etc/nginx/nginx.conf \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
```

```
--pid-path=/var/run/nginx.pid \  
--lock-path=/var/run/nginx.lock \  
--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 \  
--user=nginx \  
--group=nginx \  
--with-pcre \  
--with-http_v2_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_mp4_module \  
--with-http_gunzip_module \  
--with-http_gzip_static_module \  
--with-http_random_index_module \  
--with-http_secure_link_module \  
--with-http_stub_status_module \  
--with-http_auth_request_module \  
--with-mail \  
--with-mail_ssl_module \  
--with-file-aio \  
--with-ipv6 \  
--with-http_v2_module \  
--with-threads \  
--with-stream \  
--with-stream_ssl_module  
# make && make install  
# mkdir -pv /var/tmp/nginx/client
```

Add a SysV startup script.

```
# vim /etc/init.d/nginx  
#!/bin/sh  
#  
# 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  
# config: /etc/sysconfig/nginx  
# pidfile: /var/run/nginx.pid  
# Source function library.  
./etc/rc.d/init.d/functions  
# Source networking configuration.  
./etc/sysconfig/network  
# Check that networking is up.  
[ "$NETWORKING" = "no" ] && exit 0
```

```
nginx="/usr/sbin/nginx"
prog=$(basename $nginx)
NGINX_CONF_FILE="/etc/nginx/nginx.conf"
[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx
lockfile=/var/lock/subsys/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
killall -9 nginx
}
restart() {
configtest || return $?
stop
sleep 1
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
```

Grant the permission to run the script.

```
# chmod +x /etc/init.d/nginx
```

Add Nginx to the service management list, and set it to automatically start on startup.

```
# chkconfig --add nginx
# chkconfig nginx on
```

Start the service.

```
# service nginx start
```

Access the instance by using <http://Public IP address>. If the following page appears, Nginx is installed successfully.



Step3. Install MySQL

Follow these steps to install MySQL.

Prepare the compiling environment.

```
# yum groupinstall "Server Platform Development" "Development tools" -y
# yum install cmake -y
```

Create a directory to store the data of MySQL.

```
# mkdir /mnt/data
# groupadd -r mysql
# useradd -r -g mysql -s /sbin/nologin mysql
# id mysql
uid=497(mysql) gid=498(mysql) groups=498(mysql)
```

Change the owner and group of the data directory.

```
# chown -R mysql:mysql /mnt/data
```

Decompress and compile the stable source code package downloaded from MySQL official website. In this article, we use version 5.6.24.

```
# tar xvf mysql-5.6.24.tar.gz -C /usr/local/src
```

```
# cd /usr/local/src/mysql-5.6.24
# cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql \
-DMySQL_DATADIR=/mnt/data \
-DSYSCONFDIR=/etc \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DWITH_ARCHIVE_STORAGE_ENGINE=1 \
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
-DWITH_READLINE=1 \
-DWITH_SSL=system \
-DWITH_ZLIB=system \
-DWITH_LIBWRAP=0 \
-DMySQL_TCP_PORT=3306 \
-DMySQL_UNIX_ADDR=/tmp/mysql.sock \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci
# make && make install
```

Change the group of the installation directory to mysql.

```
# chown -R mysql:mysql /usr/local/mysql/
```

Initialize the database.

```
# /usr/local/mysql/scripts/mysql_install_db --user=mysql --datadir=/mnt/data/
```

Note: After completing the minimum installation of the CentOS 6.5 operating system, a my.cnf file is generated under the /etc directory. You must rename this file. For example, rename it as /etc/my.cnf.bak. Otherwise, this file will interfere with the correct configuration for MySQL source code installation, leading to MySQL start failure.

Copy the configuration file and startup script.

```
# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
# chmod +x /etc/init.d/mysqld
# cp support-files/my-default.cnf /etc/my.cnf
```

Set automatic start on startup.

```
# chkconfig mysqld on
# chkconfig --add mysqld
```

Modify the installation path and data storage path in the configuration file.

```
# echo -e "basedir = /usr/local/mysql\ndatadir = /mnt/data\n" > /etc/my.cnf
```

Set the PATH environment variable.

```
# echo "export PATH=$PATH:/usr/local/mysql/bin" > /etc/profile.d/mysql.sh  
# source /etc/profile.d/mysql.sh
```

Start the service.

```
# service mysqld start  
# mysql -h 127.0.0.1
```

Step 4. Install PHP-FPM

Nginx cannot process PHP. As a Web server, when Nginx receives a request, it does not support directly calling or parsing the external program. It must use FastCGI to call such programs. However, in case of PHP requests, Nginx will transfer the request to a PHP interpreter, and return the result to the client. PHP-FPM is a FastCGI process manager that supports parsing PHP code. PHP-FPM provides better PHP process management methods, which can effectively control the memory and process, and can support smoothly reloading PHP configuration.

Follow these steps to install PHP-FPM.

Install dependency package.

```
# yum install libmcrypt libmcrypt-devel mhash mhash-devel libxml2 libxml2-devel bzip2 bzip2-devel
```

Decompress the source code package downloaded from the official website, and then compile and install it.

```
# tar xvf php-5.6.23.tar.bz2 -C /usr/local/src  
# cd /usr/local/src/php-5.6.23  
# ./configure --prefix=/usr/local/php \  
--with-config-file-scan-dir=/etc/php.d \  
--with-config-file-path=/etc \  
--with-mysql=/usr/local/mysql \  
--with-mysqli=/usr/local/mysql/bin/mysqli_config \  
--enable-mbstring \  
--with-freetype-dir \  
--with-jpeg-dir \  
--with-png-dir \  
--with-zlib \  
--with-libxml-dir=/usr \  
--with-openssl \  

```

```
--enable-xml \  
--enable-sockets \  
--enable-fpm \  
--with-mcrypt \  
--with-bz2  
# make && make install
```

Add the PHP and PHP-FPM configuration files.

```
# cp /usr/local/src/php-5.6.23/php.ini-production /etc/php.ini  
# cd /usr/local/php/etc/  
# cp php-fpm.conf.default php-fpm.conf  
# sed -i 's@;pid = run/php-fpm.pid@pid = /usr/local/php/var/run/php-fpm.pid@' php-fpm.conf
```

Add the PHP-FPM startup script.

```
# cp /usr/local/src/php-5.6.23/sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm  
# chmod +x /etc/init.d/php-fpm
```

Add PHP-FPM to the service list, and set it to automatically start on startup.

```
# chkconfig --add php-fpm  
# chkconfig --list php-fpm  
# chkconfig php-fpm on
```

Start the service.

```
# service php-fpm start
```

Follow these steps to configure Nginx to support fastcgi:

- i. Back up the default configuration file.

```
# cp /etc/nginx/nginx.conf /etc/nginx/nginx.confbak  
# cp /etc/nginx/nginx.conf.default /etc/nginx/nginx.conf
```

- ii. Edit /etc/nginx/nginx.conf:

- i. Add a home page in the PHP format into the supported home page formats as shown:

```
location / {  
    root /usr/local/nginx/html;
```

```
index index.php index.html index.htm;
}
```

ii. Delete comments in front of the following content:

```
location ~ \.php$ {
    root /usr/local/nginx/html;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME /usr/local/nginx/html/$fastcgi_script_name;
    include fastcgi_params;
}
```

iii. Reload the Nginx configuration file.

```
# service nginx reload
```

Create an index.php test page under /usr/local/nginx/html/, the content of which is shown as follows:

```
# cat index.php
<?php
$conn=mysql_connect('127.0.0.1','root','');
if ($conn){
    echo "LNMP platform connect to mysql is successful!";
}else{
    echo "LNMP platform connect to mysql is failed!";
}
phpinfo();
?>
```

Step 5. Test

Access the instance by using <http://Public IP address/index.php>. If the following page appears, LNMP environment is built successfully.

PHP Version 5.6.23 	
System	Linux iZuf66k0f5Qwt2c8lbp1g2Z 2.6.32-573.22.1.el6.x86_64 #1 SMP Wed Mar 23 03:35:39 UTC 2016 x86_64
Build Date	Dec 12 2016 21:27:46
Configure Command	'./configure' '--prefix=/usr/local/php' '--with-config-file-scan-dir=/etc/php.d' '--with-config-file-path=/etc' '--with-mysql=/usr/local/mysql' '--with-mysqli=/usr/local/mysql/bin/mysql_config' '--enable-mbstring' '--with-freetype-dir' '--with-jpeg-dir' '--with-png-dir' '--with-zlib' '--with-libxml-dir=/usr' '--with-openssl' '--enable-xml' '--enable-sockets' '--enable-fpm' '--with-mcrypt' '--with-bz2'
Server API	FFM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc

Configure Java Web

Deploy a Java Web project

This article describes how to deploy a Java Web project on a Linux instance with the basic configuration. This method is applicable to individual users who are new to website construction by using ECS.

Configuration requirements

The following programs are used as examples to deploy the Java Web project:

- OS: CentOS 7.4
- Tomcat: Tomcat 8.5.23
- JDK: JDK 1.8.0_141

Preparations

The firewall is enabled by default for CentOS 7.4. You can disable the firewall, or add rules on the firewall by referring to official documents to open Ports 80, 443, or 8080 for inbound access.

- Disable the firewall.

```
systemctl stop firewalld.service
```

- Set the firewall not to be enabled automatically at startup.

```
systemctl disable firewalld.service
```

Create a user www to run Tomcat.

```
useradd www
```

Add a security group rule to open Port 8080 for HTTP access. For more information, see [Add a security group rule](#).

Creates a root directory for the Java Web project.

```
mkdir -p /data/wwwroot/default
```

Create a Tomcat test page.

```
echo Tomcat test > /data/wwwroot/default/index.jsp  
chown -R www.www /data/wwwroot
```

Download source code

Run the following command to download the tomcat package.

```
wget https://mirrors.aliyun.com/apache/tomcat/tomcat-8/v8.5.23/bin/apache-tomcat-8.5.23.tar.gz
```

Note: The source code is constantly upgraded. You can find the installation package at: <https://mirrors.aliyun.com/apache/tomcat/tomcat-8/>.

Run the following command to download the JDK package.

```
wget http://mirrors.linuxeye.com/jdk/jdk-8u141-linux-x64.tar.gz
```

Note: The source code is constantly upgraded. You can find the installation package at: <http://mirrors.linuxeye.com/jdk/>.

Install JDK

To install JDK, follow these steps:

Run `mkdir /usr/java` to create a directory.

Run the following command to decompress `jdk-8u141-linux-x64.tar.gz` to the `/usr/java` directory.

```
tar xzf jdk-8u141-linux-x64.tar.gz -C /usr/java
```

Follow these steps to set environment variables:

- i. Run `vi /etc/profile`.
- ii. Press the `i` key to enter the Edit mode.
- iii. Add the following lines into the `/etc/profile` file:

```
#set java environment
export JAVA_HOME=/usr/java/jdk1.8.0_141
export CLASSPATH=$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib
export PATH=$JAVA_HOME/bin:$PATH
```

- iv. Press the Esc key, and then type `:wq` to save and close the file.

Run `source /etc/profile` to load the new environment variable.

Check the version of JDK. When the JDK version is displayed, it indicates that JDK has been installed successfully.

```
java -version
java version "1.8.0_141"
Java(TM) SE Runtime Environment (build 1.8.0_141-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.141-b15, mixed mode)
```

Install Tomcat

To install Tomcat, follow these steps:

Run the following commands one by one to decompress `apache-tomcat-8.5.23.tar.gz`, rename the Tomcat directory, and set user permissions.

```
tar xzf apache-tomcat-8.5.23.tar.gz
mv apache-tomcat-8.5.23 /usr/local/tomcat/
chown -R www.www /usr/local/tomcat/
```

Note:

In the `/usr/local/tomcat/` directory:

- The `bin` directory stores some Tomcat script files, including scripts for enabling and disabling Tomcat service.
- The `conf` directory stores various global configuration files for Tomcat server, the most important of which are `server.xml` and `web.xml`.
- The `webapps` directory is the main Web publishing directory of Tomcat, which stores Web application files by default.

- The logs directory stores Tomcat log files.

Follow these steps to configure the server.xml file:

- Switch to the /usr/local/tomcat/conf/ directory: `cd /usr/local/tomcat/conf/`.
- Rename the server.xml file: `mv server.xml server.xml_bk`.
- Create a new server.xml file:
 - Run `vi server.xml`.
 - Press the `i` key to enter the Edit mode.
 - Add the following content.

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8006" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>
  <Listener className="org.apache.catalina.core.AprLifecycleListener"/>
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml"/>
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080"
      protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443"
      maxThreads="1000"
      minSpareThreads="20"
      acceptCount="1000"
      maxHttpHeaderSize="65536"
      debug="0"
      disableUploadTimeout="true"
      useBodyEncodingForURI="true"
      enableLookups="false"
      URIEncoding="UTF-8"/>
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost" appBase="/data/wwwroot/default" unpackWARs="true" autoDeploy="true">
        <Context path="" docBase="/data/wwwroot/default" debug="0" reloadable="false"
          crossContext="true"/>
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
          prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />
      </Host>
    </Engine>
  </Service>
</Server>
```

Follow these steps to set JVM memory parameters:

- i. Run `vi /usr/local/tomcat/bin/setenv.sh`.
- ii. Press the `i` key to enter the Edit mode.
- iii. Add the following content.

```
JAVA_OPTS=' -Djava.security.egd=file:/dev/./urandom -server -Xms256m -Xmx496m -
Dfile.encoding=UTF-8'
```

- iv. Press the Esc key, and then type `:wq` to save and close the file.

Follow these steps to set Tomcat automatic startup script:

- i. Run the command to download the script.

```
wget https://github.com/lj2007331/oneinstack/raw/master/init.d/Tomcat-init
```

- ii. Run the command to rename Tomcat-init.

```
mv Tomcat-init /etc/init.d/tomcat
```

- iii. Add the permission.

```
chmod +x /etc/init.d/tomcat
```

- iv. Set the startup script JAVA_HOME.

```
sed -i 's@^export JAVA_HOME=.*@export JAVA_HOME=/usr/java/jdk1.8.0_141@' /etc/init.d/tomcat
```

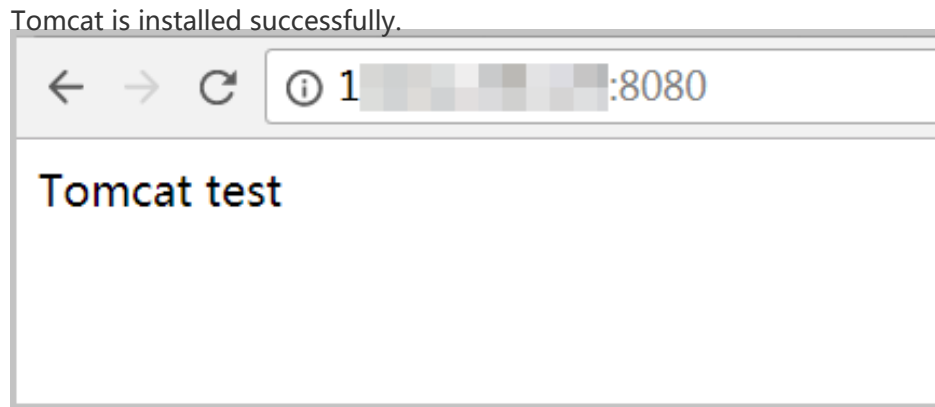
Set automatic startup.

```
chkconfig --add tomcat
chkconfig tomcat on
```

Start Tomcat.

```
service tomcat start
```

Access the instance by using `http://Public IP address:8080`. If the following page appears,



Build a WordPress website

This document describes how to create a WordPress website by using an image available on the Alibaba Cloud Marketplace. The image contains both an operating system and all applications required to start your WordPress website, including CentOS, Nginx, MySQL, PHPWind, and phpMyAdmin.

WordPress is a popular personal blog and website builder. Alibaba Cloud Elastic Compute Service (ECS) makes publishing a WordPress site simple and straightforward. You can build a WordPress by creating an ECS instance and performing a few simple configurations.

As your business develops and your website attracts more visitors, you can scale your service capacity, both vertically and horizontally, by combining other Alibaba Cloud products. For example:

- Adding ECS instances and using Server Load Balancer to more evenly process your workload.
- Using Auto Scaling to automatically add or remove instances according to traffic conditions.
- Using Object Storage Service (OSS) to store static web pages, massive pictures, and videos.

Software and versions

The applications in the image include:

- Nginx 1.10.1: high-performance web server software
- MySQL 5.7.13: a relational database management system
- PHP 5.4.45: a popular server-side scripting language
- phpMyAdmin 4.4.15.7: a web GUI for the administration of MySQL
- OpenSSH-server 6.6: a secure remote console for server management
- OpenSSH-sftp-server 6.6: a secure FTP for file uploading
- WordPress 4.5.3: a content management system for building websites

Note: The versions listed are included as of the publishing date of this document. Your versions may be different.

Install the WordPress image

Log on to the ECS console.

Go to Alibaba Cloud Marketplace.

Click **LEMP on CentOS7.2 64bits**.

Click **Choose Your Plan**.

Pay-as-you-go

Monthly subscription

ECS Usage

\$0.023 USD/Hour

Software

\$0.00 USD/Hour

FREE

Total

\$0.023 USD/Hour

Choose Your Plan

Choose initial configuration mode.

If you want to configure only essential parameters of the instance and use the default settings for others, click **Quick Buy**. If you want to configure all parameters, click **Advanced**

Buy. In this document we use **Quick Buy** as an example.

Infrastructure & Developer > Wordpress on LEMP CentOS7.2 64bits



Wordpress on LEMP CentOS7.2 64bits

Linux

Architecture:64-bit

INFINITY GLOBAL SOLUTIONS PTE LTD

Quick Buy

Advanced Buy

Click 'Accept Terms & Buy Now' to launch this software with the settings below.
Once you accept the terms, you will have access to launch any version of this software in any supported region.

VERSION	1.0	Release Date	11/08/2016
		Release Note	Initial release v1.0

Choose an image version.

VERSION	1.0	Release Date	11/08/2016
		Release Note	Initial release v1.0

Choose a region. For more information about these parameters, see [Create an instance running Linux](#).

REGION	Hong Kong
--------	-----------

Choose an ECS instance type.

After choosing a type, you can see the corresponding details.

ECS INSTANCE TYPE

Applicable Instance Type

ecs.n1.small

ecs.n2.small

ecs.e3.small

ecs.n1.medium

ecs.n2.medium

Generation	Generation II
CPU	1 core
Memory	2 GB
I/O Optimization	Yes
System Disk	40 GB

Note: Please set your instance password through the [ECS Management Console](#).

Choose a network type.

Network Type varies for different ECS features, but both of these network access services are BGP lines.

NETWORK

classic

Network Performance

Medium

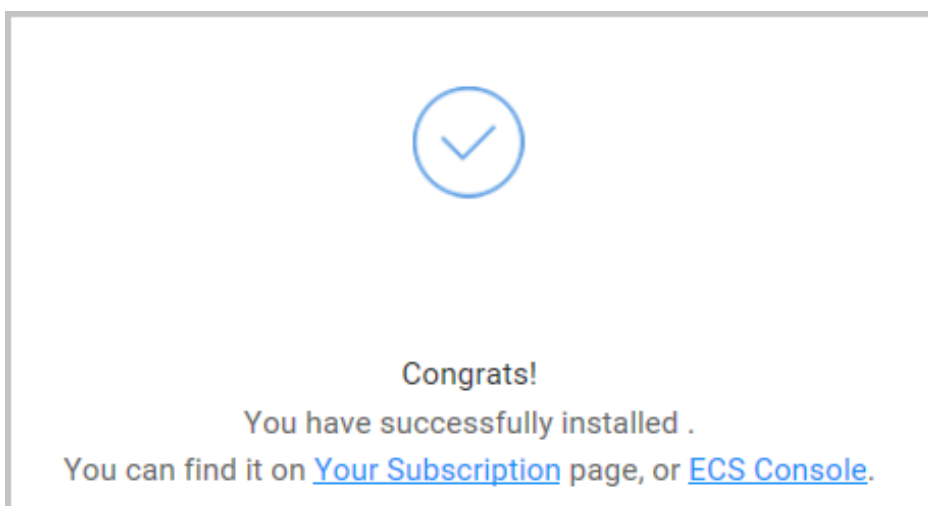
Choose the network performance.

Choose a purchasing plan: **Subscription** or **Pay-As-You-Go**.

Click **Agree Terms and Buy Now** to buy the instance.

Wait several minutes for the image to install.

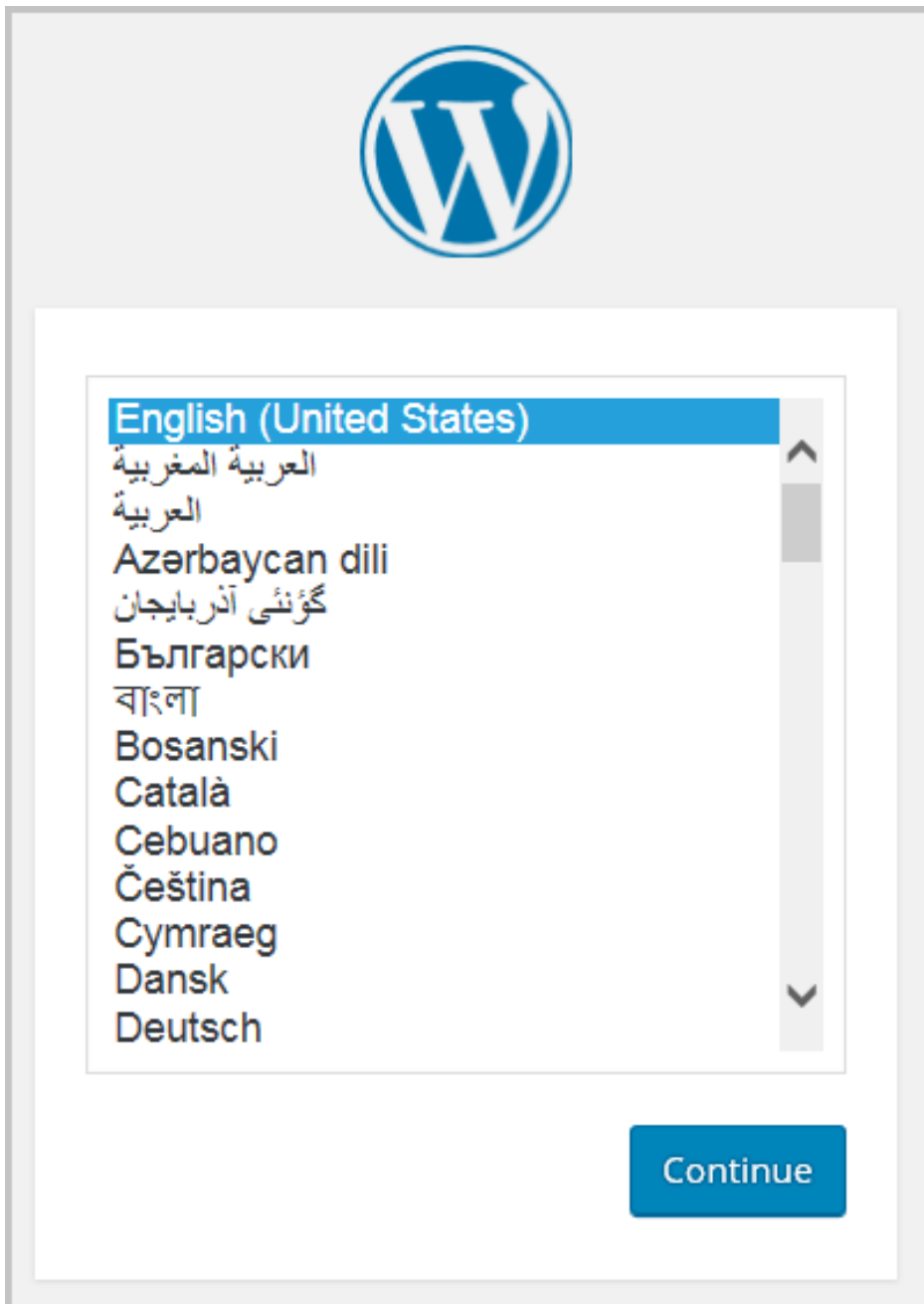
When the image is installed, the following message appears:



Log on to the ECS console, locate the instance you bought, and note down the public IP address. You must change the password and then restart the instance.

Enter the public IP address in the browser, for example, <http://47.89.30.144>. You are redirected to install WordPress.

Choose a language, and then click **Continue**.



Enter the following configurations for WordPress:

- Site title: Specify the title of your website.
- User name: Specify your user name for WordPress.
- Password: Specify your password for WordPress.
- Your Email: Specify your email address.



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username
Names can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password
Strong
Important: You will need this password to log in. Please store it in a secure location.

Your Email
Double-check your email address before continuing.

Search Engine Visibility ☐ Discourage search engines from indexing this site
It is up to search engines to honor this request.

Click **Install WordPress**.

When installation is finished, the following page appears. Click **Log In**.



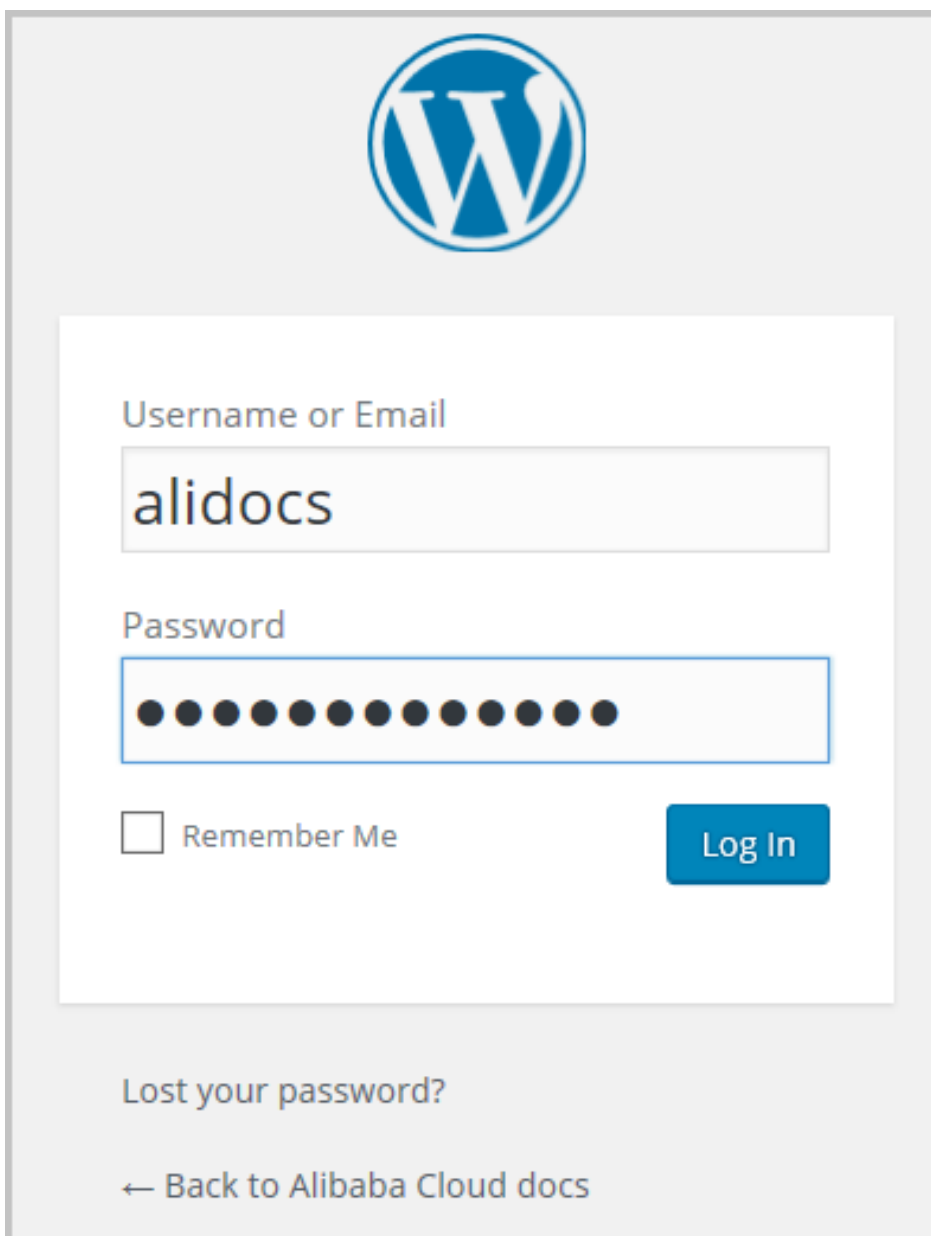
Success!

WordPress has been installed. Thank you, and enjoy!

Username alidocs

Password *Your chosen password.*

Enter the configured user name and password to log on to WordPress.

The image shows a WordPress login interface. At the top center is the WordPress logo, a blue 'W' inside a circle. Below the logo is a white rectangular box containing the login fields. The first field is labeled 'Username or Email' and contains the text 'alidocs'. The second field is labeled 'Password' and contains a series of 12 black dots. Below the password field is a checkbox labeled 'Remember Me'. To the right of the checkbox is a blue button with the text 'Log In'. Below the login box, there is a link 'Lost your password?' and a link '← Back to Alibaba Cloud docs'.

You can now customize your website on the WordPress Dashboard.

Congratulations! You have successfully created your WordPress site on Alibaba Cloud! You can now start designing and using your site.

For more instructions on how to use WordPress, go to <https://wordpress.org>.

ICP Filing

If your website is hosted in mainland China, you must complete ICP filing before users can access your website. For more information on ICP filing, see the [ICP Filing Guide](#).