

Alibaba Cloud **云服#器 ECS**

チュートリアル

Document Version20200213

目次

- 1 ソフトウェア開発環境を構築する..... 1**
- 2 ウェブサイトの構築.....2**
- 3 アプリケーションの構築.....3**
- 4 LNMP のデプロイ.....4**
 - 4.1 CentOS 6 の下での LNMP 環境の構築..... 4
- 5 Java Web の設定..... 13**
 - 5.1 Java Web プロジェクトのデプロイ.....13
- 6 ECS 上での Magento Web サイトの構築.....18**
- 7 ECS インスタンス上での FTP サイト構築.....26**
 - 7.1 Linux ECS インスタンス上での FTP サイトの構築.....26
 - 7.2 Windows ECS インスタンス上での FTP サイトの構築..... 30

1 ソフトウェア開発環境を構築する

2 ウェブサイトの構築

3 アプリケーションの構築

4 LNMP のデプロイ

4.1 CentOS 6 の下での LNMP 環境の構築

ここでは、基本設定済みの ECS インスタンス上の CentOS で、どのようにして LNMP 環境を構築するかを説明します。

- ・ **Linux:** 無料でオープンソースの、UNIX 系のソフトウェア OS (オペレーティングシステム)ファミリー。
- ・ **Nginx:** 軽量のHTTP 兼リバースプロキシサーバー。
- ・ **MySQL:** リレーショナルデータベース管理システム。
- ・ **PHP:** 特に Web 開発に最適なスクリプト言語。

対象

本手法は、Linux には習熟しているものの、Alibaba Cloud ECS を使った Web サイト構築の経験は浅い個人ユーザーを対象としています。

手順

次の手順で、ECS インスタンス上に LNMP 環境を構築します。

1. コンパイル環境を準備します。
2. Nginx をインストールします。
3. MySQL をインストールします。
4. PHP-FPM をインストールします。
5. テスト。

ステップ 1: コンパイル環境の準備

次の手順でコンパイル環境を準備します。LNMP イメージを [Cloud Market](#) で購入して ECS インスタンスを開始し、Web サイトをすばやく構築することもできます。

1. オペレーティングシステムのバージョンを確認します。

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



注:


```

--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

```

3. SysV の起動スクリプトを追加します。

```

# 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
```

- このスクリプトを実行する権限を付与します。

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

- Nginx** をサービス管理リストに追加し、起動時に自動的にスタートするよう設定します。

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

- サービスを開始します。

```
# service nginx start
```

- http://Public IP** アドレスを使って、インスタンスにアクセスします。下記のようなページが表示されれば、**Nginx** のインストールは成功しています。



ステップ 3: MySQL のインストール

- コンパイルの環境を準備します。

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

- MySQL** のデータを保存するディレクトリを作成します。

```
# 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)
```

3. データ保存用ディレクトリの所有者とグループを変更します。

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

4. 安定版のソースコードパッケージを [MySQL official website](#) からダウンロードして展開、コンパイルします。ここでは、バージョン **5.6.24** を使います。

```
# 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
```

5. インストールディレクトリのグループを **mysql** に変更します。

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

6. データベースを初期化します。

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



注:

CentOS 6.5 オペレーティングシステムの最小構成インストールを完了すると、**my.cnf** というファイルが **/etc** ディレクトリの下に生成されます。このファイルの名前を変更します。たとえば、**/etc/my.cnf.bak** という名前にします。そうしないと、このファイルが、ソースコードからインストールされた **MySQL** の正しい設定に干渉してしまい、**MySQL** が起動できなくなります。

7. 設定ファイルと起動スクリプトをコピーします。

```
# 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
```

8. 起動時の自動スタートを設定します。

```
# chkconfig mysqld on
```

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

9. 設定ファイルの中のインストールパスとデータ保存パスを変更します。

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

- 10.PATH 環境変数を設定します。

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

- 11.サービスを開始します。

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

ステップ 4: PHP-FPM のインストール

Nginx は **PHP** を処理できません。**Nginx** は、**Web** サーバーとしてリクエストを受け取っても、外部プログラムを直接呼び出すことも解析することもサポートしていません。そのようなプログラムを呼び出すには、**FastCGI** を使用しなければなりません。ただし、**PHP** のリクエストの場合、**Nginx** はリクエストを **PHP** インタープリターに転送し、その結果をクライアントに返します。**PHP-FPM** は **FastCGI** のプロセスマネージャーであり、**PHP** コードの解析をサポートします。**PHP-FPM** により、**PHP** のプロセスマネジメントのメソッドが向上するため、メモリと処理を効率的に制御し、**PHP** 設定ファイルのスムーズなロードが実現します。

1. 依存関係パッケージをインストールします。

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

2. 公式サイトからダウンロードしたソースコードパッケージを展開、コンパイル、インストールします。

```
# 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/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
# make && make install
```

3. **PHP** と **PHP-FPM** の設定ファイルを追加します。

```
# 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
```

4. **PHP-FPM** の起動スクリプトを追加します。

```
# 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
```

5. **PHP-FPM** をサービスリストに追加し、起動時に自動スタートするよう設定します。

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

6. サービスを開始します。

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

7. 以下のステップに従って **Ngix** を設定し、**fastcgi** をサポートします。デフォルトの設定ファイルをバックアップします。

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

/etc/nginx/nginx.conf を編集します。次のように、**PHP** 形式のホームページを追加して、サポートされるホームページ形式に変換します。

```
location / {
    root    /usr/local/nginx/html;
    index  index.php index.html index.htm;
}
```

次のコンテンツの前のコメントを削除します。

```
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;
```

```
}

```

Nginx の設定ファイルを再度読み込みます。

```
# service nginx reload

```

usr/local/nginx/html/ の下に index.php のテストページを作成します。その内容は次のとおりです。

```
# 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();
?>

```

http://Public IP address/index.php を使って、インスタンスにアクセスします。次のページが表示されれば、LNMP 環境の構築は成功です。

PHP Version 5.6.23 	
System	Linux iZuf66k0f52wt2c81bplg2Z 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-crypt' '--with-bz2'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc

5 Java Web の設定

5.1 Java Web プロジェクトのデプロイ

ここでは、**Java Web** プロジェクトを基本的設定済みの **Linux** インスタンスにデプロイする方法を説明します。この方法は、**ECS** を使った **Web** サイト構築の経験がないユーザー向けです。

設定の必要条件

この **Java Web** プロジェクトのデプロイには、例として次のプログラムを使います。

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

準備

- **CentOS 7.4** では、ファイアウォールがデフォルトで有効になっています。ファイアウォールを無効にするか、または、公式ドキュメントを参照してファイアウォールに関するルールを追加して、**80**、**443**、**8080** のいずれかのポートを開放してインバウンドアクセスができるようにします。

- ファイアウォールを無効化します。

```
systemctl stop firewalld.service
```

- 起動時にファイアウォールが自動的に有効とにならないように設定します。

```
systemctl disable firewalld.service
```

- **Tomcat** を実行するためにユーザー **www** を追加します。

```
useradd www
```

- **8080** ポートを開放して **HTTP** アクセスができるようにするため、セキュリティグループルールを追加します。詳細は、「[セキュリティグループルールの追加](#)」をご参照ください。

- **Java Web** プロジェクト用に **root** ディレクトリを作成します。

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

- **Tomcat** テストページを作成します。

```
echo Tomcat test > /data/wwwroot/default/index.jsp
```

```
chown -R www.www /data/wwwroot
```

ソースコードのダウンロード

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

ソースコードは常時アップグレードされています。インストールパッケージは、<https://mirrors.aliyun.com/apache/tomcat/tomcat-8/> にあります。

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

ソースコードは常時アップグレードされています。インストールパッケージは、<http://mirrors.linuxeye.com/jdk/> にあります。

JDK のインストール

JDK をインストールするには、次の手順を実行します。

1. `mkdir /usr/java` を実行して、ディレクトリを作成します。

```
mkdir /usr/java
```

2. 次のコマンドを実行し、`jdk-8u141-linux-x64.tar.gz` を `/usr/java` ディレクトリに展開します。

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

3. 次の手順で、環境変数を設定します。

- a. `vi /etc/profile` を実行します: `vi /etc/profile`
- b. `i` キーを押して、編集モードに入ります。
- c. `/etc/profile` ファイルに次の行を追加します。

```
#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
```

- d. `Esc` キーを押し、「`:wq`」と入力して、ファイルを保存して閉じます。

4. `source /etc/profile` を実行して、新しい環境変数を読み込みます。
5. JDK のバージョンを確認します。JDK のバージョンが表示されたら、JDK のインストールは成功しています。

```
java -version
```

```
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)
```

Tomcat のインストール

Tomcat をインストールするには、以下の手順を実行します。

1. 次のコマンドを順番に実行して **apache-tomcat-8.5.23.tar.gz** を展開し、**Tomcat** ディレクトリの名前を変更してユーザーパーミッションを設定します。

```
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/
```



注:

/usr/local/tomcat/ ディレクトリには、以下のディレクトリが作成されます。

- ・ **bin** ディレクトリは **Tomcat** のスクリプトファイルを保存するためのディレクトリです。中には、**Tomcat** サービスの有効化や無効化を行うためのスクリプトもあります。
- ・ **conf** ディレクトリは **Tomcat** サーバーのグローバル設定ファイルを保存します。中でも重要なのは、**server.xml** と **web.xml** です。
- ・ **webpapps** ディレクトリは **Tomcat** のメインの **Web** パブリッシング用ディレクトリで、デフォルトでは **Web** アプリケーション用ファイルの保存場所となっています。
- ・ **logs** ディレクトリはログファイルの保存場所です。

2. 次の手順で **server.xml** ファイルを設定します。

a. **/usr/local/tomcat/conf/** ディレクトリに移動します: `cd /usr/local/tomcat/conf/`

。

b. **server.xml** ファイルの名前を変更します: `mv server.xml server.xml_bk``。

c. 新たに **server.xml** ファイルを作成します。

A. `vi server.xml` を実行します。

B. `i` キーを押して、編集モードに入ります。

C. 次の内容を追加します。

```
<? 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.ThreadLocalLeakPreve
ntionListener"/> <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.MemoryUserDatabaseFa
ctory" 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" useBodyEncodingForUR
I="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.AccessLogV
alve" directory="logs" prefix="localhost_access_log." suffix=".txt
" pattern="%h %l %u %t &quot;%r&quot;%s %b" /> </Host> </Engine>
</Service> </Server>
```

3. 次の手順で **JVM** メモリのパラメーターを設定します。

- a. `vi /usr/local/tomcat/bin/setenv.sh` を実行します
- b. `i` キーを押して編集モードに入ります。
- c. 次の内容を追加します。

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

- d. `Esc` キーを押して「`:wq`」と入力し、ファイルを保存して閉じます。

4. 次の手順で、**Tomcat** の自動起動ファイルを設定します。

- a. 次のコマンドを実行してスクリプトをダウンロードします: `wget https://github.com/lj2007331/oneinstack/raw/master/init.d/Tomcat-init`
- b. 次のコマンドを実行して **Tomcat-init** の名前を変更します: `mv Tomcat-init /etc/init.d/tomcat`
- c. パーミッションを追加します: `chmod +x /etc/init.d/tomcat`
- d. 起動クリプト **JAVA_HOME** を設定します。

```
sed -i 's@^export JAVA_HOME=.*@export JAVA_HOME=/usr/java/jdk1.8.
0_141@' /etc/init.d/tomcat
```

5. 自動起動を設定します。

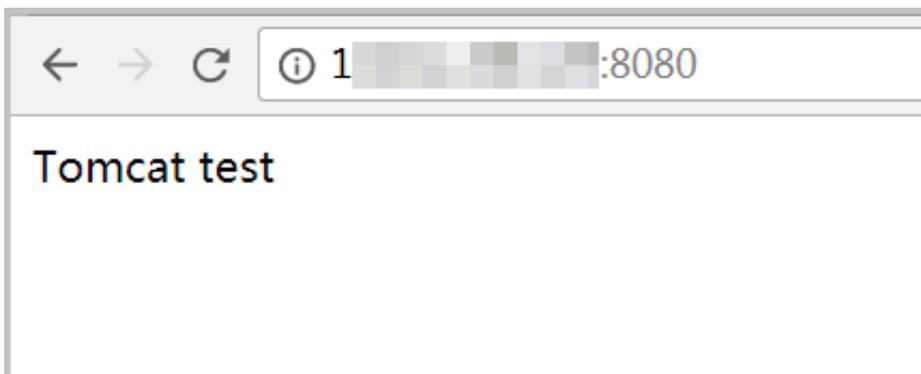
```
chkconfig --add tomcat
```

```
chkconfig tomcat on
```

6. **Tomcat** を起動します。

```
service tomcat start
```

7. **http://Public IP アドレス: 8080** でインスタンスにアクセスします。次のページが表示されれば、**Tomcat** のインストールは成功です。



6 ECS 上での Magento Web サイトの構築

Magento は PHP で書かれたオープンソースの E コマースプラットフォームです。B2B や B2C の E コマースプラットフォーム構築によく使われています。このチュートリアルでは、単一の ECS インスタンス上に **Magento** プラットフォームを構築する方法を説明します。

このチュートリアルでは、次のツールをインストールします。

- ・ MySQL バージョン: 5.7
- ・ PHP バージョン: 7.0
- ・ Magento バージョン: 2.2

前提条件

[ECS インスタンスを作成します](#)。そのインスタンスは次の条件を満たすものとします: オペレーティングシステム: CentOS 7.2 64ビット。最低限のスペックとして、2コア CPU、4 GB のメモリ、40 GB の Ultra Cloud Disk をシステムディスクとして使用。VPC 接続。VPC ネットワークがない場合は、ECS インスタンスの作成時に作成されます。インスタンスにアサインされたパブリック IP アドレス。

ECS インスタンスの TCP ポート 80 に対するインバウンドトラフィックが許可されています。詳細は、「[ECS インスタンスの作成](#)」と「[セキュリティグループルールの追加](#)」をご参照ください。

サービス	ルールの方向	権限付与ポリシー	プロトコルタイプ	ポート範囲	権限付与タイプ	権限付与オブジェクト	優先度
HTTP	インバウンド	許可	ユーザー定義の TCP	80/80	アドレスフィールドアクセス	0.0.0.0/0	1
MySQL	インバウンド	許可	ユーザー定義の TCP	3306/3306	アドレスフィールドアクセス	0.0.0.0/0	1

手順

次のステップで、ECS を使用して **Magento** の Web サイトを構築します。

ステップ 1: LAMP を ECS にインストールする

ステップ 2: データベースを設定する

ステップ 3: **Composer** をインストールして設定する

ステップ 4: **Magento** をインストールして設定する

ステップ 5: インストールをテストする

ステップ 1: **LAMP (Linux、Apache、MySQL、および PHP)** を ECS にインストールする

このセクションでは、**LAMP** プラットフォームを手動でインストールする方法を説明します。**LAMP** イメージを購入して[クラウドマーケット](#)から直接 ECS インスタンスを起動することで、すぐに **Web** サイトを構築することもできます。

1. **ECS** インスタンスに接続し、**Apache** と **MySQL** をインストールします。

```
# yum -y update
# yum -y install httpd
# rpm -Uvh http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
# yum -y install mysql-community-server
```

2. **Apache** と **MySQL** をサービスとして起動し、起動時に有効にします。

```
# systemctl start httpd
# systemctl enable httpd
# systemctl start mysqld
# systemctl enable mysqld
```

3. **Apache** の設定ファイル `/etc/httpd/conf/httpd.conf` を設定します。

a. `vim /etc/httpd/conf/httpd.conf` を実行します。

b. `i` キーを押します。

c. `Include conf.modules.d/*.conf` の下に、`conf.modules.d/*.conf` を追加します。そして次のセクションの `AllowOverride None` を `AllowOverride all` に置き換えます。

```
Options Indexes FollowSymLinks
#
# AllowOverride controls what directives may be placed in .
# htaccess files.
# It can be "All", "None", or any combination of the keywords:
# Options FileInfo AuthConfig Limit
#
AllowOverride None
```

d. `Esc` キーを押し、「`wq:`」と入力してファイルを保存して終了します。

4. 次のコマンドで、**MySQL** のインストール時に、ルートアカウントの一時パスワードを取得します。

```
# grep 'temporary password' /var/log/mysqld.log.
```

```
2016-12-13T14:57:47.535748Z 1 [Note] A temporary password is
generated for root@localhost: p0/G28g>lsHD
```

5. 次のとおり、MySQL のセキュリティ設定を完了します。

- ・ ルートアカウントのパスワードのリセット
- ・ リモートのルートログインの無効化
- ・ 匿名ユーザーの削除
- ・ テストのデータベースとテストのデータベースに対するアクセスの削除

詳細については、[公式情報](#)をご参照ください。

```
# mysql_secure_installation
Securing the MySQL server deployment.
Enter password for user root: # Enter your temporary root password
that is recorded in the previous step
The 'validate_password' plugin is installed on the server.
The subsequent steps will run with the existing configuration of the
plugin.
Using existing password for root.
Estimated strength of the password: 100
Change the password for root? (Press y|Y for Yes, any other key for
No): Y
New password: # Enter a new strong password. The password can be
[8, 30] characters in length. It must contain uppercase letters,
lowercase letters, and numbers. The following special characters are
allowed: ()`~! @#$%^&*-+=|{}[]:;'&lt;>,.? /
Re-enter new password: # Repeat the new password to confirm it
Estimated strength of the password: 100
Do you wish to continue with the password provided?( Press y|Y for
Yes, any other key for No) : Y
By default, a MySQL installation has an anonymous user, allowing
anyone to log into MySQL without having to have a user account
created for them. This is intended only for testing, and to make
the installation go a bit smoother. You should remove them before
moving into a production environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No): Y
Success.
Normally, root should only be allowed to connect from 'localhost'.
This ensures that someone cannot guess at the root password from the
network.
Disallow root login remotely? (Press y|Y for Yes, any other key for
No): Y
Success.
By default, MySQL comes with a database named 'test' that anyone can
access.
This is also intended only for testing, and should be removed before
moving into a production
environment.
Remove test database and access to it? (Press y|Y for Yes, any other
key for No): Y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for
No): Y
Success.
```

```
All done!
```

6. PHP 7をインストールします。

```
# yum install -y http://dl.iuscommunity.org/pub/ius/stable/CentOS/7/
x86_64/ius-release-1.0-14.ius.centos7.noarch.rpm
# yum -y update
# yum -y install php70u php70u-pdo php70u-mysqlnd php70u-opcache
php70u-xml php70u-gd php70u-mcrypt php70u-devel php70u-intl php70u-
mbstring php70u-bcmath php70u-json php70u-iconv
```

7. PHP のインストールを検証します。

```
# php -v
PHP 7.0.13 (cli) (built: Nov 10 2016 08:44:18) (NTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
with Zend OPcache v7.0.13, Copyright (c) 1999-2016, by Zend
Technologies
```

8. /etc/php.ini ファイルを編集してタイムゾーンを設定します。

- a. vim/etc/php.iniを実行します。
- b. i キーを押します。
- c. date.timezone で始まる行を見つけます。

```
which is commented out by default, and add the correct time zone.
If your site is in China, add date.timezone = Asia/Shanghai.
```

9. httpd を下記のコマンドで再起動します。

```
systemctl restart httpd.
```

ステップ 2 : データベースの設定

次のような手順でデータベースを設定します:

1. データベースとユーザーを作成します。次のコマンドを実行します (**mysql>** プロンプトの後のコマンドも含めてください)。

```
# mysql -u root -p
Enter password:
mysql> CREATE DATABASE magento;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL ON magento. * TO YourUser@localhost IDENTIFIED BY '
YourPass';
Query OK, 0 rows affected, 1 warning (0.00 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

2. exit を実行して MySQL を終了します。

3. 新しいユーザーをテストします。

```
# mysql -u YourUser -p
mysql> show databases;
```



```
Switched to a new branch '2.1.0'
```

3. インストールファイルを **Apache** のルートディレクトリに移動します。このステップをスキップすると、**Magento** サービスは、`http://your-server-ip /magento2` でのみアクセスできるようになります。.

```
# shopt -s dotglob nullglob && mv /var/www/html/magento2/* /var/www/html/ && cd ..
```

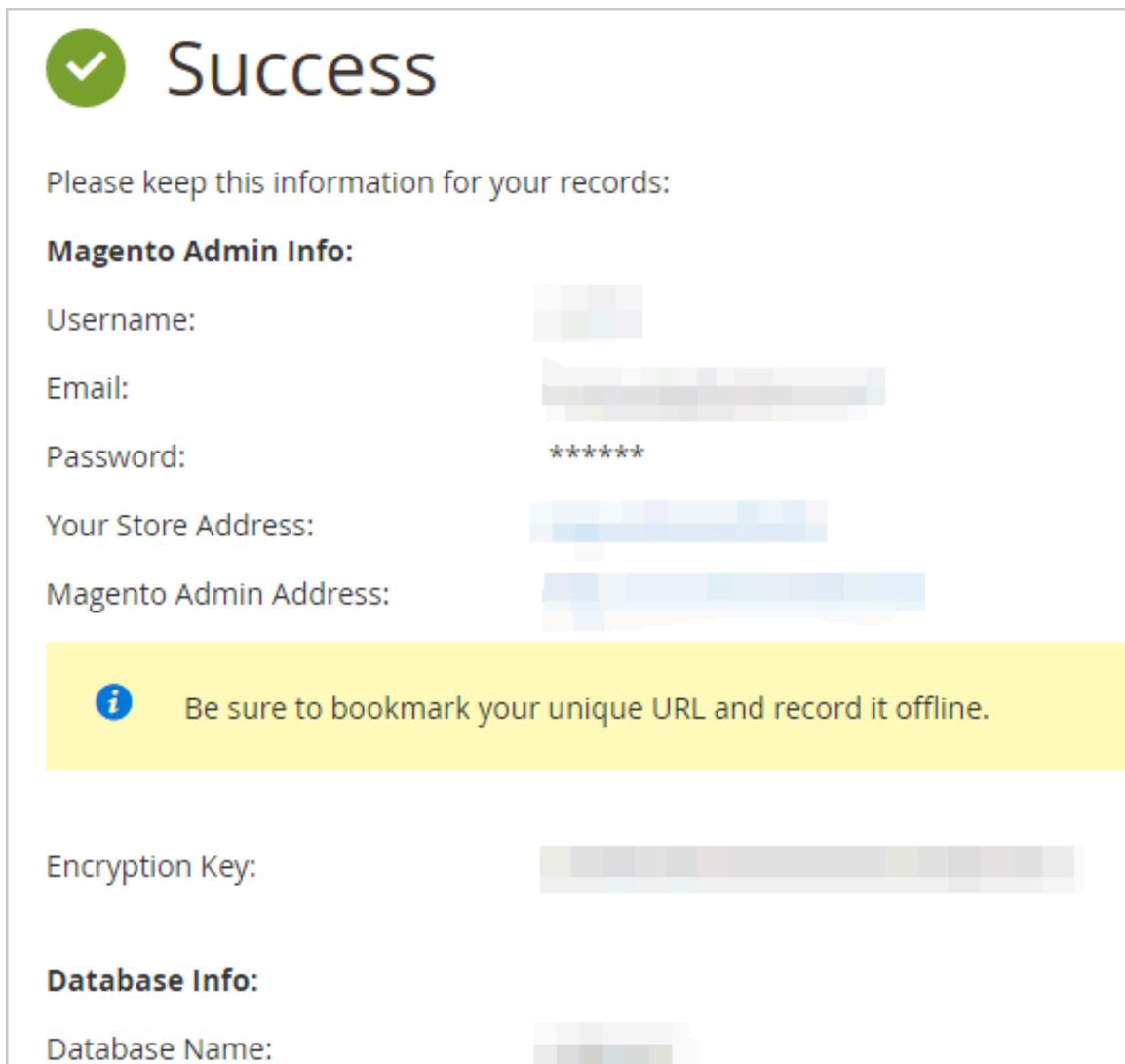
4. **Magento** のファイルのアクセス許可を以下のように設定します。

```
# chown -R :apache /var/www/html
# find /var/www/html -type f -print0 | xargs -r0 chmod 640
# find /var/www/html -type d -print0 | xargs -r0 chmod 750
# chmod -R g+w /var/www/html/{pub,var}
# chmod -R g+w /var/www/html/{app/etc,vendor}
# chmod 750 /var/www/html/bin/magento
```

5. `composer install` を実行して **Magento** をインストールします。
6. ブラウザーでサーバー (`http://ECS インスタンスのパブリック IP アドレス`) にアクセスします。下のような、画面が表示されます。



7. [同意して **Magento** を設定] をクリックし、データベース情報、**Web** 設定、アカウントを次のように入力します。以下のような画面が表示されれば、インストールは成功です。



ステップ 5: cron ジョブの設定

1. `crontab -u apache -e` を実行します。
2. `/etc/crontab` ファイルに以下を追記します。

```
*/10 * * * * php -c /etc /var/www/html/bin/magento cron:run
*/10 * * * * php -c /etc /var/www/html/update/cron.php
*/10 * * * * php -c /etc /var/www/html/bin/magento setup:cron:run
```

詳細は、[公式ドキュメント](#)をご参照ください。

次にすること

`http://ECS` インスタンスのパブリック IP アドレス にアクセスして、デフォルトのホームページを表示します。

`http://ECS` インスタンスのパブリック IP アドレス/`admin` にアクセスして、インストール時に設定したユーザー名とパスワードを使用して、ダッシュボードにログインします。

Magento の設定について詳しくは、[公式ドキュメント](#)を参照してください。

7 ECS インスタンス上での FTP サイト構築

7.1 Linux ECS インスタンス上での FTP サイトの構築

vsftpd は Linux 用の軽量で使いやすい FTP サーバーです。Linux のあらゆるバージョンを通じて、最も人気のある FTP サーバーです。ここでは、CentOS 7.2 x64 を稼働中の Linux ECS インスタンス上に **vsftpd** をインストールする方法を説明します。ここでは、CentOS 7.2 x64 を稼働中の Linux ECS インスタンス上に **vsftpd** をインストールする方法を説明します。

Linux ECS インスタンス上に FTP サイトを構築するには、以下のステップを実行します。

- ・ ステップ 1. **vsftpd** のインストール
- ・ ステップ 2. **vsftpd** の設定
- ・ ステップ 3. セキュリティグループの設定
- ・ ステップ 4. テスト

ステップ 1. vsftpd のインストール

1. `#unique_13`.
2. 次のコマンドを実行して **vsftpd** をインストールします。

```
yum install -y vsftpd
```

次のコマンドを実行して

3. `etc/vsftpd` を開き、閲覧します。

```
cd /etc/vsftpd  
ls
```



注:

- ・ `/etc/vsftpd/vsftpd.conf` がコアの設定ファイルです
- ・ `/etc/vsftpd/ftpusers` はブラックリストです。このブラックリストのユーザーは FTP サーバーにアクセスできなくなります
- ・ `/etc/vsftpd/user_list` はホワイトリストです。このホワイトリストのユーザーだけが FTP サーバーへのアクセスを許可されます。

4. 次のコマンドを実行して、**vsftpd** が起動時に自動的に開始するようにします。

```
systemctl enable vsftpd.service
```

5. 次のコマンドを実行して **FTP** サービスを起動します。

```
systemctl start vsftpd.service
```

6. 次のコマンドを実行して **FTP** サービスのポートを閲覧します。

```
netstat -antup | grep ftp
```

ステップ 2. vsftpd の設定

vsftpd をインストールすると、匿名 **FTP** がデフォルトで有効になります。匿名 **FTP** 機能を使うと、ユーザーはユーザー名とパスワードなして **FTP** サーバーにログインできますが、ファイルの修正やアップロードはできません。

このセクションでは、**vsftpd** の設定方法と、それに対応したパラメーター記述を説明します。

- 匿名ユーザーへのファイルアップロードアクセス許可の付与
- ローカルユーザーのログイン設定
- vsftpd.conf** のパラメーターの紹介

匿名ユーザーへのファイルアップロードアクセス許可の付与

vsftpd.conf 設定ファイルのオプションを修正すれば、匿名ユーザーに付与するアクセス許可を増やせます。

1. **/etc/vsftpd/vsftpd.conf** を次の手順で修正します。

- vim /etc/vsftpd/vsftpd.conf** を実行します。
- i** キーを押して編集モードに入ります。
- write_enable=YES** と設定します。
- anon_upload_enable=YES** と設定します。
- Esc** キーを押し、「**:wq**」と入力して、ファイルを保存して閉じます。

2. 次のコマンドを実行して **/var/ftp/pub** ディレクトリのパーミッションを変更し、**FTP** ユーザーに書き込み許可を与え、設定ファイルを再読み込みします。

```
chmod o+w /var/ftp/pub/systemctl restart vsftpd.service
```

ローカルユーザーのログイン設定

ローカルユーザーのログインとは、ユーザーが **Linux** オペレーティングシステムのユーザー名とパスワードを使って、**FTP** サーバーにログインすることです。

vsftpd をインストールした段階では、匿名 FTP ログインだけがサポートされています。もし **Linux** のユーザー名でログインしようとする、**vsftpd** へのアクセスは拒否されます。ただし **vsftpd** 設定ファイルを修正して、ユーザー名とパスワードによるログインを可能にできます。次の手順を実行します。

1. 次のコマンドを実行して、**ftptest** ユーザーを作成します。

```
useradd ftptest
```

2. 次のコマンドを実行して、**ftptest** ユーザーのパスワードを変更します。

```
passwd ftptest
```

3. 次の手順で `/etc/vsftpd/vsftpd.conf` を修正します。

- a. `vim/etc/vsftpd/vsftpd.conf` を実行します。
- b. **i** キーを押して編集モードに入ります。
- c. `anonymousenable=NO` と設定します。
- d. `local_enable=YES` と設定します。
- e. **Esc** キーを押し、「`:wq`」と入力して、ファイルを保存して閉じます。

4. 次のコマンドを実行して、設定ファイルを再読み込みします。

```
systemctl restart vsftpd.service
```

vsftpd.conf のパラメーターの概要

`cat /etc/vsftpd/vsftpd.conf` を実行して、設定ファイルの中身を見ます。

次の表は、ユーザーログイン制御に関連するすべてのパラメーターをまとめたものです。

パラメーター	説明
anonymous_enable=YES	匿名ログインを許可します。
no_anon_password=YES	ログイン時、匿名ユーザーはパスワードの入力を求められません。
anon_root=(none)	匿名ユーザーのルートディレクトリ。
local_enable=YES	ローカルユーザーのログインを許可します。
local_root=(none)	ローカルユーザーのルートディレクトリ。

次の表は、ユーザーアクセス許可制御に関連するすべてのパラメーターをまとめたものです。

パラメーター	説明
write_enable=YES	ファイルアップロードを許可します (グローバル制御I)。
local_umask=022	ローカルユーザーのファイルアップロードに対して umask を行います。
file_open_mode=0666	ファイルアップロード権限に対して umask を使います。
anon_upload_enable=NO	匿名ユーザーに対してファイルアップロードを許可します。
anon_mkdir_write_enable=NO	匿名ユーザーにディレクトリの作成を許可します。
anon_other_write_enable=NO	匿名ユーザーにファイルやディレクトリの変更や削除を許可します。
chown_username=lightwiter	匿名ユーザーにユーザー名を付与します。

匿名でアップロードされたファイルのユーザー名

FTP サイトを構築したら、FTP ポートを開くためのルールを追加する必要があります。詳細は、「[セキュリティグループルールの追加](#)」をご参照ください。

ステップ4. テスト

[ローカルコンピューター] 上で、ftp://public IP address:FTP port (ポート番号を入力しなければ、デフォルトのポート番号 **21** が使われます) を使って FTP サイトにアクセスします。たとえば、「ftp://0.0.0.0:20」と入力します。設定がうまくいっていれば、ユーザー名とパスワードを入力するように求められます。ユーザー名とパスワードを正しく入力すると、権限に従って、関連する FTP ファイル操作を行うことができます。



注:

この方法でクライアントから FTP サイトにアクセスする場合、FTP フォルダーを開くように **Internet Explorer** の設定を変更する必要があります。 **Internet Explorer** を開き、[ツール] > [インターネット オプション] > [詳細設定] をクリックします。 [FTP サイト用のフォルダービューを使用する] を選択し、 [パッシブ FTP を使用する] をクリアします。

次にすること

FTP サービスのセキュリティ向上の対策を実行します。 詳細は、「[FTP 匿名ログインと弱いパスワードの脆弱性](#)」をご参照ください。

7.2 Windows ECS インスタンス上での FTP サイトの構築

ここでは、**Windows ECS** インスタンス上に **FTP** サイトを構築する方法を説明します。この方法は **Windows Server 2008** とその後継バージョンで有効です。ここでは、**Windows Server 2008 R2** を使用します。

Windows ECS インスタンス上に **FTP** サイトを構築する手順は以下のとおりです。

- ・ ステップ 1. **IIS** と **FTP** サービスのロールを追加
- ・ ステップ 2. **FTP** のユーザー名とパスワードを作成
- ・ ステップ 3. 共有ファイルへのパーミッションを設定
- ・ ステップ 4. **FTP** サイトを追加して設定
- ・ ステップ 5. セキュリティグループとファイアウォールの設定
- ・ ステップ 6. テスト

ステップ 1. IIS と FTP サービスのロールを追加

FTP サイト構築の前に、**IIS** と **FTP** サービスをインストールします。

1. [#unique_15](#).
2. [スタート] > [すべてのプログラム] > [管理ツール] > [サーバー マネージャー] をクリックします。
3. 左側のナビゲーションウィンドウから[ロール]をクリックし[ロールを追加]をクリックします
4. ダイアログボックスの [次へ] をクリックします
5. [ウェブサーバー (IIS)] をクリックし、次に [次へ] をクリックします
6. [IIS 管理コンソール]と [FTP サーバー]を選択し、[次へ] をクリックして [インストール] をクリックします。

ステップ 2. FTP のユーザー名とパスワードを作成

匿名ユーザーがこの **FTP** にアクセスすることを許可する場合は、このステップはスキップしてください。

1. [スタート] > [管理ツール] > [サーバー マネージャー] をクリックします。
2. [設定] > [ローカルユーザーとグループ] > [ユーザー]をクリックし、空白部分で右クリックして、[新しいユーザー]を選択します。[新しいユーザー] ダイアログボックスで、新しいユーザーの情報を入力します。ここでは **ftptest** を使うことにします。



注:

パスワードには、大文字、小文字、数字を含める必要があります。そうしなければ、パスワードは無効です。

ステップ 3. 共有ファイルへの権限を設定

FTP サイトのユーザーに共有するフォルダーには、読み出し、書き込み、実行の権限を付与する必要があります。

1. FTP サイト用のフォルダーを作成し、そのフォルダーを右クリックして、[プロパティ] を選択します。
2. [セキュリティ] をクリック、[ユーザー] を選択、[編集] をクリックします。
3. [ユーザーのアクセス許可] を編集します。ここでは、[すべてのアクセス許可] を付与します。

ステップ 4. FTP サイトを追加して設定

次のステップで FTP サイトをインストールします。

1. [スタート] > [すべてのプログラム] > [管理ツール] > [インターネット インフォメーション サービス (IIS) マネージャー] をクリックします。
2. 左側のナビゲーションウィンドウで インスタンス ID をクリックし、[サイト] を右クリック、[FTP サイトの追加] をクリックします。
3. ダイアログボックスで、FTP サイト名と、共有フォルダーの物理的パスを設定し、[次へ] をクリックします。
4. IP アドレスの [既定値] を使用し、このインスタンスのポート番号を入力します。デフォルトの FTP ポート番号は 21 です。
5. SSL 設定を選択します。
 - ・ **SSL を許可:** この FTP サイトに、クライアントとの間で SSL と非 SSL 両方の通信を許可します。
 - ・ **SSL が必要:** FTP サーバーとクライアントの間の通信に、SSL 暗号化を必要とします。
 - ・ **SSL なし:** SSL 暗号化が必要ない場合に [SSL なし] を選択します。
6. 1 つ以上の認証方法を選択します。
 - ・ **匿名:** [anonymous] または [ftp] というユーザー名を入力することで、どのユーザーでも共有フォルダーにアクセスできるようにします。
 - ・ **基本:** 共有コンテンツにアクセスする前に、有効なユーザー名とパスワードの入力を必須とします。基本認証方法は、ネットワークを通じて、暗号化されていないパスワードを送信します。そのため、この認証方式は、クライアントと FTP サーバーの間の通信が安全であると確信できる場合、たとえば SSL が使用されている場合にだけ使用してください。

7. [権限付与] リストの次の選択肢から **1** つを選び、アクセス許可を設定します。

- ・ **すべてのユーザー:** 関連するコンテンツに、すべてのユーザー (匿名ユーザーと身元が特定されているユーザーの両方) がアクセスできます。
- ・ **匿名ユーザー:** 関連するコンテンツに匿名ユーザーがアクセスできます。
- ・ **指定されたロールまたはユーザー グループ:** 関連するコンテンツに、特定のロールグループまたはユーザーグループがアクセスできます。対応する入力欄にロールグループやユーザーグループを入力します。
- ・ **指定されたユーザー:** 関連するコンテンツに、指定されたユーザーのみがアクセスできます。対応する入力欄にユーザー名を入力します。

8. 権限のあるユーザーに対して付与する [読み出し] または [書き込み] パーミッションを選択します。 [終了] をクリックします。

ステップ 5. セキュリティグループとファイアウォールの設定

FTP サイトを構築したら、FTP のポートに対するインバウンドトラフィックを許可するために、セキュリティグループにルールを追加しなければなりません。詳細は、「[セキュリティグループのルールの追加](#)」をご参照ください。

デフォルトでは、TCP ポート **21** が FTP サービスのためにサーバーファイアウォールで開いています。別のポート番号を入力した場合は、このポートがファイアウォールで開くように、インバウンドルールを追加しなければなりません。

ステップ 6. テスト

[ローカル コンピューター] 上で、ftp://IP address:FTP port (ポート番号を入力しなければ、デフォルトのポート番号 **21** が使われます) を使って、FTP サイトにアクセスします。たとえば、「ftp://0.0.0.0:20」と入力します。設定がうまくいっていれば、ユーザー名とパスワードを入力するように求められます。ユーザー名とパスワードを正しく入力すると、パーミッションに従って、関連する FTP ファイル操作を行うことができます。



注:

この方法でクライアントから FTP サイトにアクセスする場合、FTP フォルダーを開けるように **Internet Explorer** の設定を変更する必要があります。 **Internet Explorer** を開き、 [ツール] > [インターネット オプション] > [詳細設定] をクリックします。 [FTP サイト用のフォルダービューを使用する] を選択し、 [パッシブ FTP を使用する] の選択を解除します。

次にすること

[FTP サービスのセキュリティ向上の対策](#)を実行します。

詳細は、「[FTP 匿名ログインと弱いパスワードの脆弱性](#)」をご参照ください。