

# Instalar ABCD/iah en Linux/Apache en un servidor virtual AWS

## Notas sobre comandos y recomendaciones

Preparado por Wenke Adam – 1ª versión de 12.11.2017

### 1. ¿Qué es AWS?

AWS – Amazon Web Services es un conjunto de servicios web mantenidos por Amazon. Véalos en <http://aws.amazon.com>

Son servicios “en la nube” donde todo es elástico y escalable se adapta automáticamente al volumen de recursos necesarios para correr cualquier tipo de aplicaciones y se pueden expandir o reducir en espacio sin intervención nuestra. Durante el primer año AWS ofrece una parte de estos servicios gratis para clientes nuevos, dentro de ciertos límites de espacio y tráfico. Si el cliente excede estos límites, se le empieza a cobrar automáticamente por el excedente. Una instalación ABCD típica no excede los límites.

Si usted tiene una cuenta en Amazon (por ejemplo para comprar libros en Kindle) la misma cuenta sirve para abrir cuenta en AWS. Si no la tiene, cree una cuenta gratuita en <http://aws.amazon.com>

AWS tiene una enorme cantidad de documentación y tutoriales en inglés, español y otros idiomas. La interfaz de cada servicio está en inglés y tiene ayudas contextuales para cada paso.

#### 1.a. Ventajas y desventajas de AWS para ABCD:

##### Ventajas:

Permite crear servidores virtuales en que todo el software, desde el sistema operativo, el apache y el ABCD mismo son instalados y configurados por el administrador (el usuario “dueño” del servidor virtual).

##### Desventajas:

Permite crear servidores virtuales en que todo el software, desde el sistema operativo, el apache, y el ABCD mismo son instalados y configurados por el administrador (el usuario “dueño” del servidor virtual).

##### O sea, ¡AWS tiene una curva de aprendizaje!

Además, los servicios en la nube (y hay muchos) tienden a crear sus propias terminologías y muchísimas siglas, lo cual complica un poco al principio. Por ejemplo, en AWS un servidor se llama una “instancia”.

#### 1.b. ¿Qué es y qué no es este manual?

Este manual está destinado a colegas que tienen poca experiencia con linux pero alguna experiencia con apache y ABCD en windows. (como yo).

El manual recoge los comandos linux (y algunas explicaciones y recomendaciones) usados por la autora para instalar en AWS / EC2 un servidor virtual con **Linux versión Amazon** (similar a CentOS), un paquete **LAMP** con apache, php, y mysql, y una versión “recortada” de **ABCD** que contiene solo el **iah, cgi-bin y bases**, para servir de OPAC de una entidad pequeña. La base de datos propia se enlazará a un sitio web externo tipo wordpress, ya existente, por lo que no se incluye el módulo **site**.

En el manual solo se menciona brevemente la creación de una instancia (espacio para el servidor) en AWS/EC2, un proceso que está bien documentado (en inglés) en la propia interfaz interactiva de EC2, y en manuales y tutoriales en español. También hay numerosos foros web dedicados al tema de los servicios AWS.

### 2. Preparativos

Antes de comenzar, debe tener en su PC local los siguientes programas. Googléelos, son gratuitos:

**PUTTY** para comunicarse a distancia con linux y darle comandos para la creación de directorios, archivos, dar permisos, arrancar y detener el apache, etc.

**WinSCP** para comunicarse en modo gráfico desde windows con la estructura de directorios de su servidor, subir archivos desde el PC, etc. Es como un c-panel light, pero para instalarlo localmente.

**Notepad++** para editar archivos de configuración y otros y evitar que queden con códigos windows no legibles por linux. Notepad++ permite mantener abiertos dos o más archivos del mismo nombre residentes en diferentes carpetas, abrir un conjunto de archivos para hacer búsqueda y reemplazo de texto en todos ellos, etc.

### 3. Crear el servidor virtual (instancia)

1. Abra una cuenta en [aws.amazon.com](https://aws.amazon.com). Puede cambiar el idioma del sitio a español, funcionará para los manuales.
2. Abra la consola (Sign in to the console) y vaya a EC2 (abreviatura de Elastic Compute Cloud).
3. Abra el tutorial **Getting started guide** en la columna derecha. Si le aparece en inglés, cambie nuevamente el idioma a español y cambiará a este idioma. Mantenga abierto el tutorial para consultarlo paso a paso.
4. En la consola elija **Launch Instance** (botón grande, azul).
5. Se abre el Step 1 (Paso 1). Elija la versión linux **Amazon Linux AMI 2017.09.0 (HVM), SSD Volume Type - ami-c5062ba0**. Como ya se dijo, es una versión de CentOS.
6. Siga las instrucciones del tutorial hasta el fin hasta crear su primera instancia (servidor virtual).
7. Como cliente gratuito, sólo puede tener **una instancia en operación**. Tome nota de la URL y IP de la instancia. Detenga y reactive la instancia (Actions->**Stop** y Actions->**Restart**), y verá que la URL y la IP cambiaron. Esto sería grave si usted ya estuviera en producción. Hacer un Action->**Reboot** no pierde la IP. Se recomienda crear una **IP elástica**, que mantiene su valor después de detener la instancia con Stop. Vea información más abajo.
8. Termine la instancia como instruido en el tutorial (Actions->Terminate), espere un minuto hasta que aparezca terminada, y repita la operación de crear y terminar una instancia unas dos o tres veces hasta familiarizarse con el proceso. ¡Esta parte es sólo para entrenamiento!
9. Al crear la primera instancia también se creará un Grupo de Seguridad y un Par de Claves. Sin embargo, antes de empezar a cargar LAMP y ABCD conviene crear por lo menos **un usuario con privilegios de administrador**, y volver a entrar a la consola con estas credenciales a crear la instancia definitiva. Lea la parte del manual [Configuración con Amazon EC2](#) que le dará más información.
10. Antes de crear su primera (y única) instancia definitiva, borre todos los Grupos de Seguridad y Pares de Claves que haya creado antes y vuelva a empezar de cero. Los Grupos y Claves son específicos para una instancia determinada. Si deja los antiguos y después se confunde de Grupo o Clave (como me pasó a mí al principio, de puro entusiasmo) no podrá acceder de nuevo a la instancia.

#### 3.a. Crear una instancia definitiva

**3.a.1. Al crear la instancia, elija el Amazon Linux de AWS como indicado arriba.**

##### 3.a.2. Grupo de Seguridad

Hay uno que se crea por defecto. También puede crear otro grupo con un nombre a su gusto. Tiene que crear 3 reglas de entrada (Inbound) en el Grupo de Seguridad activo. Verificar y crear en la consola AWS.

```
ssh      tcp    22
http     tcp    80
https    tcp    443
```

##### 3.a.3. Par de claves

Cree su par de claves para esta instancia, dándole un nombre a su gusto. Descargue la llave a su PC (y guarde otra copia en lugar seguro) y abra **PUTTYgen** (un programita que viene con PUTTY) para crear una llave tipo PUTTY, con extensión .ppk. Siga las instrucciones del manual. Cree la clave .ppk con **Save**, no la genere con Generate, no va a funcionar!

##### 3.a.4. IP elástica

Recomiendo vivamente crear una IP elástica en este punto del proceso. Usted tiene derecho a una IP

elástica por cada instancia activa, solo le cobran si la crea y no la usa. Siga las instrucciones del manual AWS y conecte la IP elástica a su instancia. A partir de allí, la IP elástica será permanente y su valor estará también incluido en la URL. Si la IP es 123.45.67.89, su URL será ec2-123-45-67-89.us-east-2.compute.amazonaws.com. Note que clientes de Chile se adjudicarán automáticamente a la zona de Ohio, USA.

## 4. Instalación de LAMP (apache, php, mysql) desde PUTTY

Configure el PUTTY para que se comuniquen con su instancia. Va a necesitar: su URL, el nombre de usuario que es siempre EC2-user, la ruta a la clave .ppk en su PC, y una contraseña de entrada. Siga las instrucciones del manual y de PUTTY.

### Aquí empezamos a dar comandos a Linux via PUTTY

Primero instalaremos LAMP con php 5.6.

**Nota: Sudo** es un comando linux que indica que estamos hablando con la autoridad de un administrador y tenemos acceso a todo el contenido del servidor. **¡Úselo con cuidado!**

#### 4.a. Update de LAMP (buscar actualizaciones de LAMP antes de la instalación):

```
sudo yum update -y
```

#### 4.b. Instalar paquete LAMP:

```
sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
```

**Nota: si prefiere php70, ingrese:**

```
sudo yum install -y httpd24 php70 mysql56-server php70-mysqlnd
```

Y si no quiere mysql, instale solo httpd24 y php70.

Recomiendo instalar todo, por si más adelante quisiera instalar el resto de ABCD conEmpWeb.

#### 4.c. Verificar instalación del paquete:

```
sudo yum list installed httpd24 php56 mysql55-server php56-mysqlnd
```

Muestra lista similar a:

Loaded plugins: priorities, update-motd, upgrade-helper

Installed Packages

httpd24.x86_64	2.4.25-1.68.amzn1	@amzn-updates
mysql56-server.x86_64	5.6.35-1.23.amzn1	@amzn-updates
php70.x86_64	7.0.14-1.20.amzn1	@amzn-updates
php70-mysqlnd.x86_64	7.0.14-1.20.amzn1	@amzn-updates

#### 4.d. Iniciar Apache:

```
sudo service httpd start
```

#### 4.e. Para que Apache se reinicie en cada boot:

```
sudo chkconfig httpd on
```

#### 4.f. Confirmar que Apache está corriendo:

```
chkconfig --list httpd
```

Muestra lista:

```
httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

## 5. Probar el apache en el navegador

### 5.a. Usaremos una IP / URL ficticia como ejemplo.

Con Public DNS:

```
ec2-123-45-67-89.us-east-2.compute.amazonaws.com
```

Con IPv4 Public IP (Es elastic IP):

```
123.45.67.89
```

Debiera aparecer la página de saludo de Apache.

### 5.b. ¿Dónde están los archivos de Apache y home? :

```
ls -l /var/www
debe aparecer algo así como:
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug 7 00:02 error
drwxr-xr-x 2 root root 4096 Jan 6 2012 html
drwxr-xr-x 3 root root 4096 Aug 7 00:02 icons
drwxr-xr-x 2 root root 4096 Aug 7 21:17 noindex
```

El doc root por defecto es: `/var/www/html/`, pero usaremos `/var/www/vhosts/ABCD/`

## 6. Configurando Apache y ABCD

### 6.a. Crear directorio vhosts:

```
sudo mkdir /var/www/vhosts/
```

### 6.b. Crear un subdir por cada aplicación que se quiere instalar (por ahora sólo ABCD):

```
sudo mkdir /var/www/vhosts/ABCD/
```

### 6.c. Podemos crear desde ya el subdir www de ABCD

```
sudo mkdir /var/www/vhosts/ABCD/www/
```

### 6.d. Agregar ownership y permissions

#### 6.d.1. Agregar ec2-user al apache group: (OJO: se puede crear más usuarios de apache después)

```
sudo usermod -a -G apache ec2-user
```

#### 6.d.2. Hacer log out y login de PUTTY para actualizar y verificar membresía en apache group:

Log out: `exit`

Se desconecta PUTTY.

Login: Reconectar PUTTY a la instancia (login desde PUTTY).

#### 6.d.3. Ver estado del grupo apache:

```
groups
```

debe aparecer:

```
ec2-user wheel apache
```

#### 6.d.4. Cambiar ownership de var/www y su contenido al grupo apache:

```
sudo chown -R ec2-user:apache /var/www
```

#### 6.d.5. Agregar permisos de escritura a var/www y prepararlo para futuras subdirs:

```
sudo chmod 2775 /var/www
```

```
find /var/www -type d -exec sudo chmod 2775 {} \;
```

#### 6.d.6. Agregar permisos recursivos a subdirs:

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

#### 6.d.7. Crear un doc test PHP en doc root:

```
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Si no funciona echo, hacer logut/login para que apache tome los nuevos permisos.

#### 6.d.7. En el navegador, probar el php:

```
http://ec2-123-45-67-89.us-east-2.compute.amazonaws.com/phpinfo.php
```

```
http://123.45.67.89/phpinfo.php
```

Debiera verse la página PHP info.

#### 6.d.8. Borrar el archivo phpinfo.php: (¡Después de terminar todas las configuraciones!)

```
rm /var/www/html/phpinfo.php
```

## 7. Configurar el VirtualHost

Ya tenemos el directorio `var/www/vhosts/ABCD/` creado para contener ABCD.

#### 7.a. Crear un archivo vhost.conf en el directorio /etc/ de configuraciones del apache:

```
cd /etc/httpd/conf.d
sudo touch vhost.conf
```

#### 7.b. Abrir el editor nano para agregar contenido al archivo vhost.conf:

```
sudo nano /etc/httpd/conf.d/vhost.conf
```

#### 7.c. Con nano, agregar un VirtualHost para ABCD y otro genérico que sigue la config por defecto.

```
<VirtualHost *:80>
    ServerName ec2-123-45-67-89.us-east-2.compute.amazonaws.com
    ServerAlias 123.45.67.89
    DocumentRoot /var/www/vhosts/ABCD/www/htdocs/
    ScriptAlias /cgi-bin/ "/var/www/vhosts/ABCD/www/cgi-bin/"
    ServerAdmin mimail@gmail.com
    ErrorLog /var/www/vhosts/logs/ABCD_error_log

    <Directory /var/www/vhosts/ABCD>
        AllowOverride All
    </Directory>

    <Directory "/var/www/vhosts/ABCD/www/cgi-bin/">
        Options +ExecCGI
        AddHandler cgi-script .cgi
    </Directory>

</VirtualHost>

<VirtualHost *:80>
    ServerName default:80
    DocumentRoot /var/www/html/
    ServerAdmin mimail@gmail.com
    ErrorLog /var/www/vhosts/logs/error_log
</VirtualHost>
```

#### 7.d. Guardar el archivo vhost.conf y Cerrar el nano usando:

`Ctrl-x` y luego `Y` (yes) para confirmar guardar.

#### 7.e. Nota: Los logs son opcionales. Si los declaramos en VirtualHosts, también hay que crear el subdir logs y los dos archivos antes de refrescar el apache.

```
cd /var/www/vhosts
sudo mkdir /var/www/vhosts/logs
sudo touch error_log
sudo touch ABCD_error_log
```

#### 7.f. Refrescar el apache

```
sudo service httpd restart
```

## 8. Subir ABCD al servidor:

Primero hay que configurar WinSCP para comunicarse con el servidor. Siga las instrucciones del manual AWS y Ayudas de WinSCP.

Sugiero preparar en el PC local una instalación mínima de ABCD para iah solamente, verificar que todo funcione en localhost, y subir las carpetas bases, htdocs y cgi-bin de allí al servidor (mantenemos la estructura original de ABCD más que nada para facilitar la consulta de ejemplos de manuales y colegas de la lista ISIS). Así, si algo funcionó antes y deja de funcionar en el servidor, usted sabrá que no es por problemas técnicos serios, sino que serán simples errores de rutas o de haber cargado el cgi-bin o las bases en versión Windows en vez de linux, por ejemplo.

La instalación debe tener como mínimo los siguientes directorios de ABCD:

cgi-bin	Copiar el directorio cgi-bin de la versión linux de ABCD.
bases	Contiene las bases biblo y la propia, el directorio par y los archivos de configuración. La base Biblo se incluye para probar el iah, por si la base propia tiene problemas. Copiar bases desde la versión linux de ABCD. Recuerde que la base propia que usa en Windows debe pasar por crunchmt y crunchif antes de subir al servidor.
htdocs	Copiar el directorio iah completo.

Subir estas 3 carpetas al directorio `/var/www/vhosts/ABCD/www/` con WinSCP.  
Las carpetas cgi-bin y bases deben subir como binarios, htdocs como texto.

Con WinSCP dar los siguientes permisos generales y recursivos (O con PuTTY usando sudo chmod y find):

```

cgi-bin  2775
bases    0777
htdocs   0664

```

## 8.a. Configurar el ABCD/iah

Editar los archivos de configuración de ABCD si no lo hizo antes. Puede usar nano o el editor de WinSCP en el servidor, o si los edita antes de subir, hágalo con Notepad++ localmente y luego los sube. Los archivos son:

```

/var/www/vhosts/ABCD/www/htdocs/iah/scripts/iah.php.ini
/var/www/vhosts/ABCD/www/bases/abcd.def
/var/www/vhosts/ABCD/www/bases/bases.dat

```

## 8.b. Configurar el iah.def.php

```

<?php
[PATH]
PATH_DATA=/iah/
PATH_CGI-BIN=/var/www/vhosts/ABCD/www/htdocs/iah/scripts/
PATH_DATABASE=/var/www/vhosts/ABCD/www/bases/
PATH_DEF=/var/www/vhosts/ABCD/www/bases/par/
[APPEARANCE]
BODY BACKGROUND COLOR=#F7F5D0 (en mi caso, elegí un fondo crema claro)
/* Please adjust /css/stylesheet.css */
[HEADER]
LOGO IMAGE=logo-ad.jpg
LOGO URL=^1http://www.midominio.cl^2http://www.midominio.cl^3http://www.
midominio.cl^4http://www.midominio.cl
HEADER IMAGE=online.gif
HEADER URL=^1http://www.midominio.cl^2http://www.midominio.cl^3http://www.
midominio.cl^4http://www.midominio.cl

[IAH]
MANAGER E-MAIL=mimail@gmail.com
REVERSE MODE=OFF
MULTI-LANGUAGE=OFF
AVAILABLE LANGUAGES=es
?>

```

En este ejemplo, sustituí los links que venían por defecto, que enlazaban con la BVS de Bireme (logo) y página principal de site (header) que no usaremos, colocando en su lugar la URL (ficticia) de un sitio web externo `www.midominio.cl` desde donde se llamará a la base con un enlace.

### 8.c. Configurar el abcd.def

```
VERSION=1.4      (o la que usted esté usando)
LEGEND1=Nombre de la institución
LEGEND2=Nombre de la institución
URL1=http://www.midominio.cl
URL2=http://www.midominio.cl
dubcore=unicode
unicode=unicode
diglib=ffi
```

En este caso, no parece ser necesario editar el archivo abcd.def porque iah no lo usa, sólo se activa en el módulo central y site. Pero editelo por si acaso.

### 8.d. Configuración de bases.dat

Incluya las bases que tiene, por ejemplo BIBLO y MIBASE.

```
biblo|CEPAL|Y
mibase|MIBASE|Y
```

### 8.e. Probar en el navegador que wxis esté funcionando:

```
http://123.45.67.89/cgi-bin/wxis?hello
```

#### Wxis debe responder:

```
CISIS Interface v5.7e/G/PC/W/M/32767/16/60/I/64bits - XML IsisScript WWWISIS
7.1f
CISIS Interface v5.7e/.iy0/Z/GIZ/DEC/ISI/UTL/INVX/B7/FAT/CIP/CGI/MX/W
Copyright (c)BIREME/PAHO 2010. [http://reddes.bvsalud.org/projects/cisis]
WXIS release date: Aug  5 2015
```

```
WXIS|missing error|parameter|IsisScript|
```

### 8.f. Luego podemos probar si wxis está viendo al iah:

```
http://123.45.67.89/cgi-bin/wxis?IsisScript=getenv.xis
```

#### Wxis debe responder algo así:

```
CGI Environment Variables
AUTH_TYPE=
CONTENT_LENGTH=
CONTENT_TYPE=
GATEWAY_INTERFACE=CGI/1.1
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp
,image/apng,*/*;q=0.8
HTTP_USER_AGENT=Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36
PATH_INFO=
PATH_TRANSLATED=
QUERY_STRING=IsisScript=getenv.xis
REMOTE_HOST=
REMOTE_ADDR=152.172.80.148
REMOTE_IDENT=
REQUEST_METHOD=GET
REMOTE_USER=
SERVER_NAME=123.45.67.89
SERVER_SOFTWARE=Apache/2.4.27 (Amazon) PHP/5.6.31
SERVER_PORT=80
SERVER_PROTOCOL=HTTP/1.1
SCRIPT_NAME=/cgi-bin/wxis
IsisScript Environment Variables
PATH_WXIS=/var/www/vhosts/ABCD/www/cgi-bin/
```

La prueba de getenv.xis muestra que wxis está funcionando correctamente y reconoce a iah.

Si wxis no agarra el script getenv.xis, mostrará mensaje de error:

```
WXIS|file error|file open|Isis_Script|
```

## 8.f. Probar en el navegador que las configuraciones de iah y la base BIBLO funcionan:

`http://123.45.67.89/cgi-bin/wxis/iah/scripts/?IsisScript=iah.xis&lang=es&base=BIBLO`

Debiera aparecer la página de búsqueda simple de la base Biblio.

Si algo no funciona, es casi siempre debido a un problema de rutas mal configuradas en algún archivo de configuración de apache (vhost.conf) o del ABCD.

## 8.g. Para probar la base propia, basta con substituir BIBLO por MIBASE en el navegador.

`http://123.45.67.89/cgi-bin/wxis/iah/scripts/?IsisScript=iah.xis&lang=es&base=MIBASE`

Ahora puede guardar esta dirección en la barra de marcadores, como MIBASE AWS por ejemplo, para poder abrirla las veces que quiera hasta haber completado la conexión de esta misma URL al sitio web madre.

## 9. Próximos pasos (en estudio)

### 9.a. Crear un repositorio fuera de EC2 para imágenes y otros recursos estáticos de la base.

Aún no lo he hecho, lo estoy estudiando. Mi base tiene más de 5.000 imágenes jpg en resolución baja más unos 500 pdfs, además de las respectivas 5.500 miniaturas que se muestran en los registros. Los subí por ahora a subcarpetas de iah, que es donde los tenía en mi PC local. Mi instalación ABCD/iah pesa cerca de 2,5 GB. Eso, más el linux y LAMP, están lejos de ocupar los 5 GB a que tengo derecho en el contrato gratuito del primer año.

El almacenamiento del repositorio en un "bucket" o contenedor de archivos estáticos del módulo **S3** de Amazon es gratuito hasta para 5 GB el primer año, y después será mucho más barato que en EC2. Hay varias otras opciones para repositorios mayores.

### 9.b. Asociar un subdominio de `www.midominio.cl` a esta instancia.

Mi dominio y sitio web en wordpress están hospedados en Hostgator. Entiendo que la asociación se hace a través del administrador de DNS de AWS llamado **Amazon Route 53**. Implica cambiar configuraciones en NIC Chile, Hostgator y Route 53. Estudiar el procedimiento.

### 9.c. Migrar la instalación wordpress de Hostgator a AWS.

Estudiar el procedimiento. ¡Aterrorador! Pero factible. Hay muchos foros sobre migración de wordpress a AWS. ¡Deséenme suerte!

## 10. Conclusiones

Si bien al principio la cosa parece complicada, descubrí que, con un poco de práctica, todo el proceso de crear la instancia, instalar el software y configurar todo hasta tener el ABCD/iah corriendo, podía tomar unas 3 horas (más las 6 horas que dejé corriendo la carga del repositorio durante la noche, sin atención).

Pasado el primer año gratuito, calculo que esta instalación podrá costar unos 5,00 USD por mes, lo que es aceptable en este caso, comparado con lo que cobran los hostings tradicionales por mantener un servidor virtual dedicado tipo VPN. Hay que tener atención a que Amazon cobra por absolutamente todos los recursos que se usan, pero los servicios de AWS están pensados para grandes sistemas con enorme tráfico (por ejemplo, AWS hospeda a Netflix, entre muchos otros gigantes), por lo que las necesidades de un simple catálogo online de una entidad pequeña estará casi siempre dentro de los cobros mínimos.