

**AWS DevOps para Tod@s**

# Índice de Contenidos

- Conceptos básicos de AWS
- Herramientas de Despliegue
- Terraform
- Instalación de las Herramientas
- Datos de Acceso
- AWS CLI
- Creación del Proyecto
- Base de Datos
- Despliegue del Backend
- Despliegue del Frontend

# Repositorio de Ejemplos del Taller



# Conceptos básicos de AWS

# Conceptos básicos de AWS

Amazon Web Services (en adelante AWS) es el servicio en la nube que dispone Amazon tanto para sus propios servicio como para sus clientes

# Conceptos básicos de AWS

De cara a poder interactuar con AWS será necesario darnos de alta como usuarios en la Consola de AWS

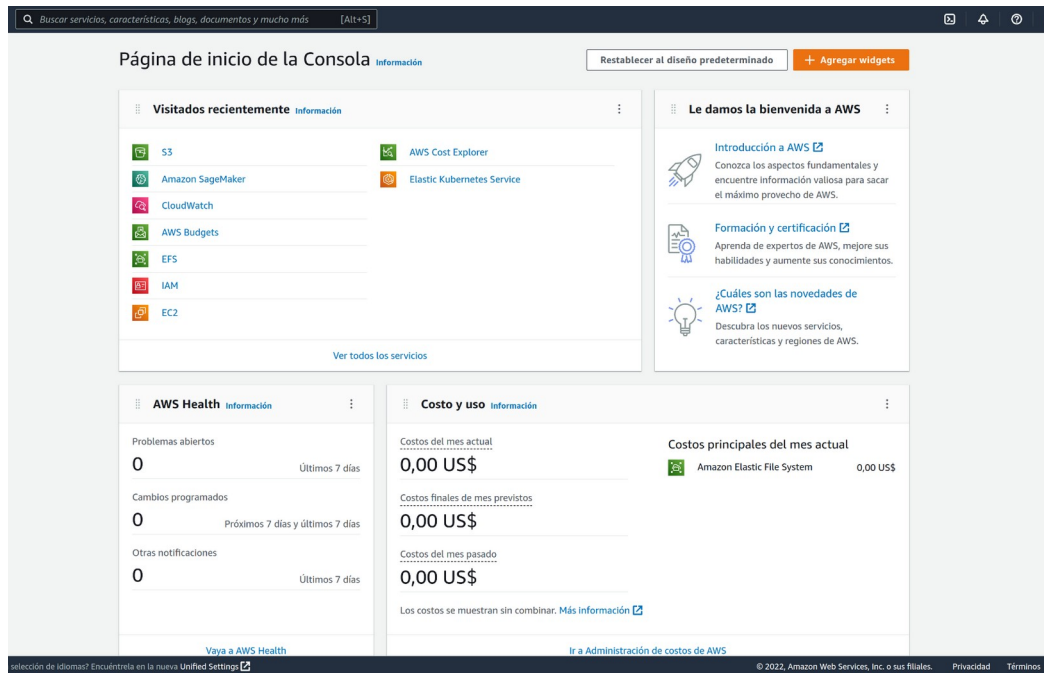


<https://aws.amazon.com/es/console/>

# Conceptos básicos de AWS

AWS dispone de cientos de servicios orientados a realizar el despliegue, monitorización y auditoría de diferentes soluciones

# Conceptos básicos de AWS



- Autenticación y Autorización
- Máquinas Virtuales
- Almacenamiento
- DNS/Rutas
- Bases De Datos
- Balanceadores de Carga
- etc...



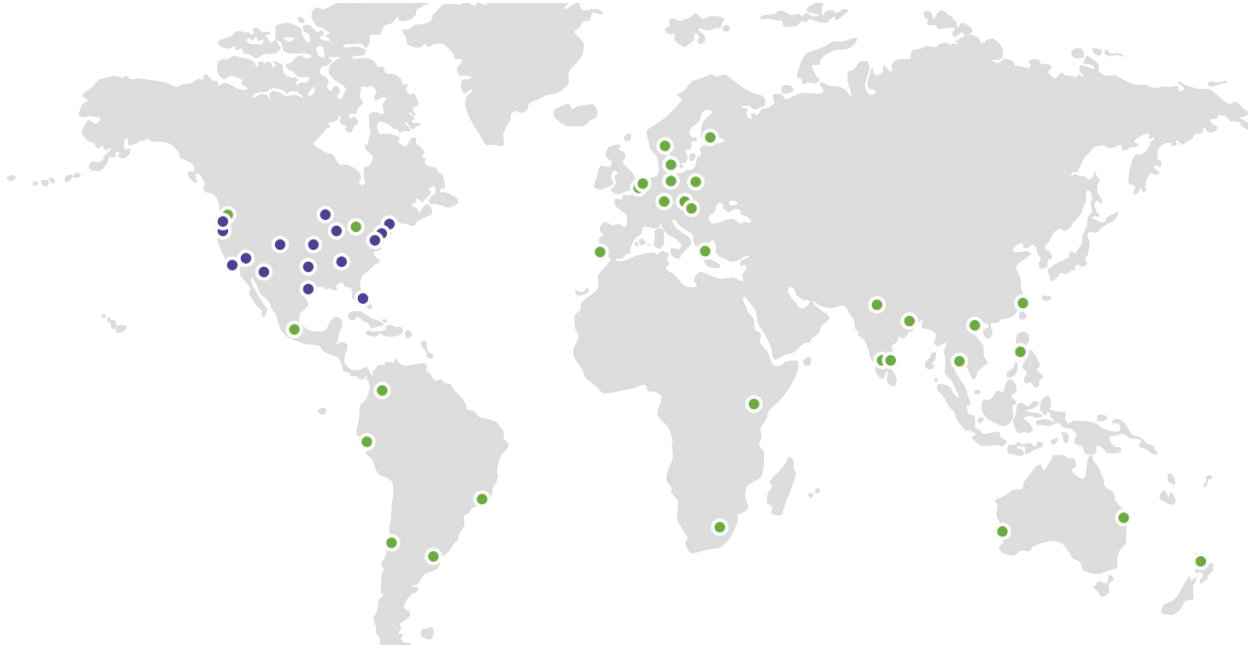
# Conceptos básicos de AWS

La escalabilidad y estabilidad de los servidores es algo que tiene en cuenta AWS mediante la localización de sus Regiones (Regions)



[https://aws.amazon.com/es/about-aws/global-infrastructure/regions\\_az/](https://aws.amazon.com/es/about-aws/global-infrastructure/regions_az/)

# Conceptos básicos de AWS



# Conceptos básicos de AWS

Las Regiones son ubicaciones físicas de los Centros de Datos de AWS y están divididas en Zonas de Disponibilidad

# Conceptos básicos de AWS

Por ejemplo en España tenemos la región

eu-south-2

# Conceptos básicos de AWS

Dentro de cada Región disponemos de mínimo tres Zonas de Disponibilidad (Availability Zones ó AZ)

# Conceptos básicos de AWS

En la región de España tenemos tres zonas de disponibilidad

eu-south-2a, eu-south-2b, eu-south-2c

# Conceptos básicos de AWS

Cada Zonas de Disponibilidad está compuesta por uno o más centros de datos con alimentación eléctrica, red y comunicaciones redundantes

# Conceptos básicos de AWS

Esto permite que cada Región puede ser redundante, o incluso a nivel de AZ también lo puede ser



# Conceptos básicos de AWS

Para entornos de alta disponibilidad será interesante hacer despliegues en diferentes AZ

# Conceptos básicos de AWS

Por lo que los despliegues que hagamos en AWS pueden ser consolidados a nivel de Región para que si un Centro de Datos tiene un problema, el resto pueda seguir dando servicio

# Conceptos básicos de AWS

Es decir, con AWS podemos conseguir un Alto Nivel de Disponibilidad de los servicios Desplegados

66

**Nunca te fíes de que un servidor  
vaya a funcionar siempre. Nunca.**

Un Servidor

# Conceptos básicos de AWS

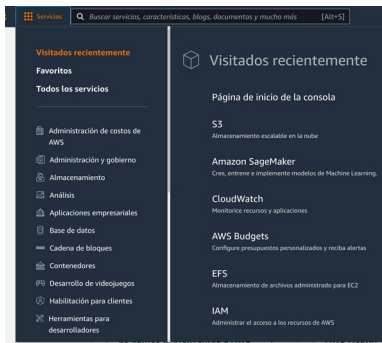
Es decir, con AWS podemos conseguir un Alto Nivel de Disponibilidad de los servicios Desplegados

# Herramientas de Despliegue

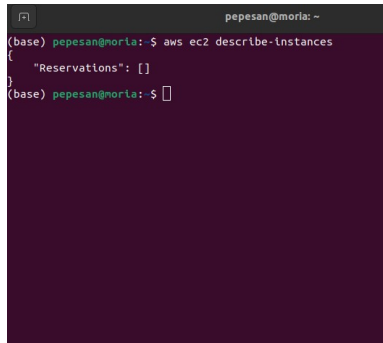
# Herramientas de Despliegue

Para poder interactuar con AWS disponemos de multitud de métodos diferentes

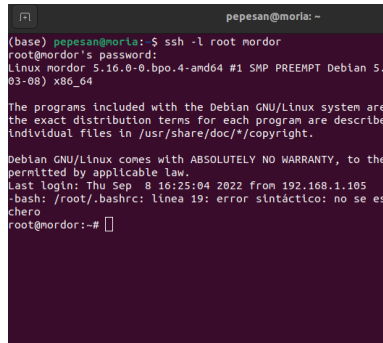
# Herramientas de Despliegue



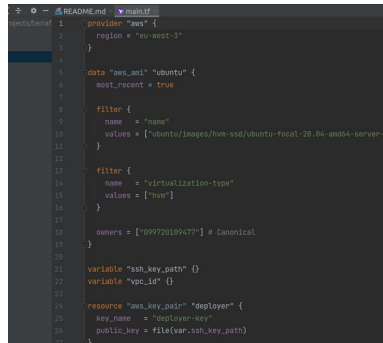
**Consola AWS**



**AWS CLI**



**SSH**



**Herramientas IaC**



# Herramientas de Despliegue

Dentro de las herramientas principales de IAC, podríamos destacar **Terraform y Ansible**

# Herramientas de Despliegue

A lo largo de este Workshop nos centraremos en la **Consola AWS y Terraform** como herramienta de **IaC** (Infraestructura como Código, en castellano)

# Terraform

# Terraform

Terraform, ó TF de aquí en adelante, es una herramienta de software libre para el aprovisionamiento y despliegue de Infraestructura como código (IaC en inglés) creada por HashiCorp



<https://www.terraform.io/>

# Terraform

Terraform nos va a permitir a partir de ficheros de código fuente, definir la infraestructura que nosotros necesitamos para luego desplegarla en un servicio en la nube o datacenter propio

# Terraform

Terraform dispone de proveedores que permiten la comunicación con los diferentes servicios en la nube así como la mayor parte de sistemas de virtualización en un datacenter



<https://registry.terraform.io/browse/providers>

# Terraform



AWS



Google Cloud



Azure



VMWare

# Terraform

Cada uno de estos proveedores de Terraform tiene su propia forma de funciona pero suelen disponer de dos tipos de maneras de interactuar: Fuentes de Datos y Recursos



# Terraform

Las Fuentes de Datos o DataSources son las que nos permitirán consultar informaciones de la plataforma: como por ejemplo imágenes de instalación o AMI's disponibles en una localización

# Terraform

Los Recursos serán todas aquellos Elementos que podemos dar de alta o gestionar desde cada plataforma conectable

# Terraform

En el caso de AWS por ejemplo dispondremos de Recursos, por ejemplo, para RDS, ECS y Route53 (El sistema DNS de AWS)

# Terraform

Como cualquier otro tipo de conexión a un servicio en la nube deberemos disponer de las credenciales adecuadas para cada proveedor, al menos una por cada nube (una por cada proveedor)

# Terraform

Una vez conseguidas estas credenciales y configurado el proveedor con ellas con los permisos adecuados podremos dar de alta y mantener nuestra infraestructura

# Terraform



- Definimos los requisitos
- Escribimos los scripts de TF
- Los lanzamos contra una nube
- Damos de alta la Infraestructura en dicha nube

# **Instalación de Herramientas**

# Instalación de Herramientas

Como comentamos en la introducción la idea de este workshop es la de realizar estos pasos y muchos más usando Terraform



# Instalación de Herramientas

Para ello lo primero que debemos hacer es realizar la instalación del comando de Terraform



<https://developer.hashicorp.com/terraform/install>

# Instalación de Herramientas

Si estamos en una máquina GNU/Linux Ubuntu,  
deberemos ejecutar en la terminal

# Instalación de Herramientas

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o  
/usr/share/keyrings/hashicorp-archive-keyring.gpg  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee  
/etc/apt/sources.list.d/hashicorp.list  
sudo apt update && sudo apt install terraform
```

# Instalación de Herramientas

Si estamos en una máquina GNU/Linux RedHat o derivados, deberemos ejecutar en la terminal

# Instalación de Herramientas

```
sudo yum install -y yum-utils  
sudo yum-config-manager --add-repo  
https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo  
sudo yum -y install terraform
```

# Instalación de Herramientas

Si estamos en una máquina Windows, deberemos descargarlo desde la página en formato Zip

Descomprimirlo en una carpeta

Colocarla dentro de la variable de sistema PATH

Así podremos ejecutarla desde cualquier lado

# Instalación de Herramientas

Si estamos en una máquina MacOSX, podemos descargarlo desde la página en formato Zip

Descomprimirlo en una carpeta

Colocarla dentro de la variable de sistema PATH

# Instalación de Herramientas

En MacOSX si disponemos de Homebrew también podemos instalarla ejecutando comandos



# Instalación de Herramientas

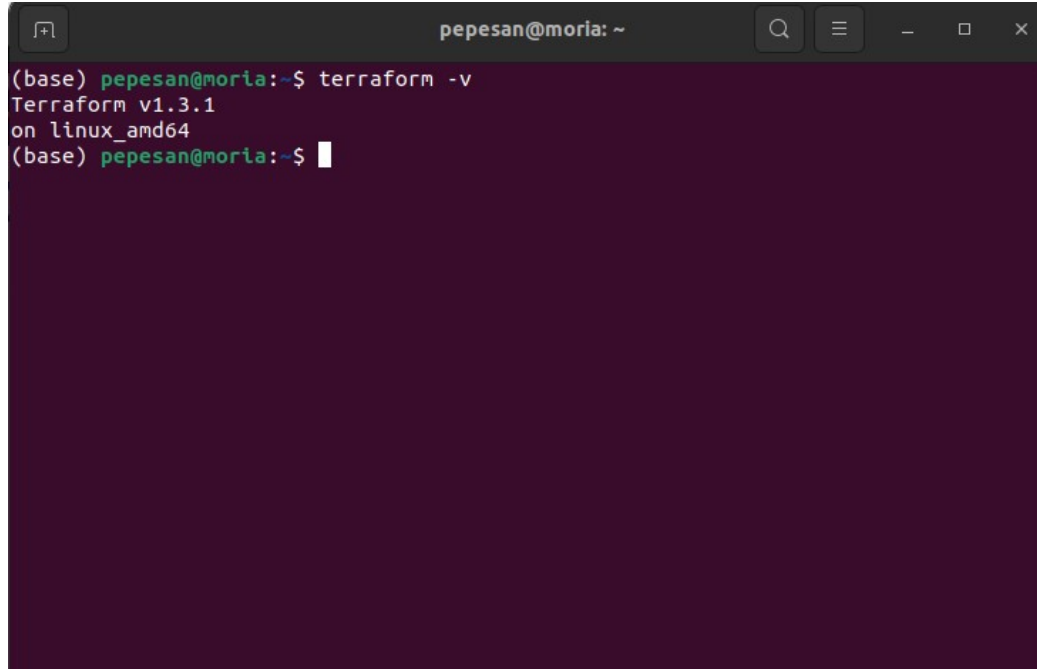
```
brew tap hashicorp/tap
```

```
brew install hashicorp/tap/terraform
```

# Instalación de Herramientas

En cualquier caso deberíamos ser capaces de ejecutar el comando desde cualquier carpeta en la terminal o el intérprete de comandos de Windows

# Instalación de Herramientas

A terminal window with a dark background and light green text. The window title is 'pepesan@moria: ~'. The command 'terraform -v' has been executed, resulting in the output 'Terraform v1.3.1 on linux\_amd64'. The prompt '(base) pepesan@moria:~\$' is visible at the end of the line.

```
(base) pepesan@moria:~$ terraform -v
Terraform v1.3.1
on linux_amd64
(base) pepesan@moria:~$
```

- Si todo va bien deberíamos poder ver una salida similar a la siguiente

# Instalación de Herramientas

Si no tienes git previamente instalado, puedes descargarlo desde su página Web



<https://git-scm.com/downloads>

# Instalación de Herramientas

No olvides inicializar el usuario y el email para git  
con los comandos:

```
git config --global user.email "tu_usuario@dominio.com"  
git config --global user.name "tuusuario"
```

# Instalación de Herramientas

Ya deberíamos de estar preparados para empezar a realizar los scripts de Terraform y Git

# **Creación de la cuenta de AWS**

# Creación de la cuenta de AWS

El primer paso que debemos hacer es crearnos una cuenta en AWS, para ello iremos a su página principal de la consola de AWS



<https://aws.amazon.com/es/>



# Creación de la cuenta de AWS

Allí, si no disponemos de una cuenta previa nos aparecerá un botón llamado (Create an AWS Account ó Cree una cuenta de AWS) que pulsaremos para ir al Asistente de Alta de Usuario

# Creación de la cuenta de AWS



Explore los productos de la capa gratuita con una cuenta de AWS nueva.

Para obtener más información, visite [aws.amazon.com/free](https://aws.amazon.com/free).



## Registrarse en AWS

Dirección de correo electrónico del usuario raíz  
Se utiliza para la recuperación de cuentas y algunas funciones administrativas

Nombre de la cuenta de AWS  
Elija un nombre para la cuenta. Podrá cambiarlo en la configuración de la cuenta después de registrarse.

Verificar la dirección de correo electrónico

o

Iniciar sesión en una cuenta de AWS existente

Rellenamos nuestro correo

Ponemos un Nombre a la cuenta de AWS para diferenciarla de otras que tengamos asociada

Pulsamos en Verificar la dirección de correo electrónico

# Creación de la cuenta de AWS



Explore los productos de la capa gratuita con una cuenta de AWS nueva.

Para obtener más información, visite [aws.amazon.com/free](https://aws.amazon.com/free).



## Registrarse en AWS

### Confirme que es usted

Garantizar que esté seguro, es lo que hacemos.

Hemos enviado un correo electrónico con un código de verificación a [pruebasopenwebminar@gmail.com](mailto:pruebasopenwebminar@gmail.com). (¿No es usted?)

Introdúzcalo a continuación para confirmar su correo electrónico.

Código de verificación

Verificar

Volver a enviar el código

¿No ha recibido el código?

- Los códigos pueden tardar hasta 5 minutos en llegar.
- Revise su carpeta de spam.

Nos llegará un correo electrónico al buzón del correo que hayamos especificado con un código de verificación

Meteremos el código Y pulsamos en Verificar

# Creación de la cuenta de AWS



Explore los productos de la capa gratuita con una cuenta de AWS nueva.

Para obtener más información, visite [aws.amazon.com/free](https://aws.amazon.com/free).



Registrarse en AWS

Cree la contraseña

✓ Hemos verificado su identidad. ✕  
La dirección de correo electrónico se ha verificado correctamente.

La contraseña proporciona acceso de inicio de sesión a AWS, por lo que es importante que este proceso se realice de forma correcta.

Contraseña de usuario raíz

Confirmar la contraseña del usuario raíz

Continuar (paso 1 de 5)

o

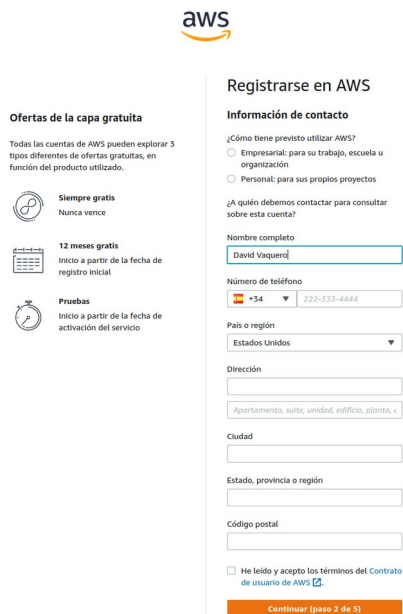
Iniciar sesión en una cuenta de AWS existente

Una vez verificada el código deberemos introducir dos veces una contraseña

Debe tener al menos 8 caracteres, mayúsculas, minúsculas, números y caracteres no alfanuméricos

Pulsa Continuar

# Creación de la cuenta de AWS



The screenshot shows the AWS account creation registration page. On the left, there are three promotional offers: 'Siempre gratis' (Always free), '12 meses gratis' (12 months free), and 'Pruebas' (Trials). The main section is titled 'Registrarse en AWS' (Sign up for AWS). It includes a 'Información de contacto' (Contact information) section with radio buttons for account type (Personal or Business) and a dropdown for account purpose. Below this are input fields for 'Nombre completo' (Full name), 'Número de teléfono' (Phone number), 'País o región' (Country or region), 'Dirección' (Address), 'Ciudad' (City), 'Estado, provincia o región' (State, province or region), and 'Código postal' (Postal code). At the bottom, there is a checkbox for 'He leído y acepto los términos del Contrato de usuario de AWS' (I have read and accept the AWS User Agreement) and a 'Continuar (paso 2 de 5)' (Continue) button.

aws

**Ofertas de la capa gratuita**

Todas las cuentas de AWS pueden explorar 3 tipos diferentes de ofertas gratuitas, en función del producto utilizado.

**Siempre gratis**  
Nunca vence

**12 meses gratis**  
Inicio a partir de la fecha de registro inicial

**Pruebas**  
Inicio a partir de la fecha de activación del servicio

**Registrarse en AWS**

**Información de contacto**

¿Cómo tiene previsto utilizar AWS?

☐ Empresarial: para su trabajo, escuela u organización

☐ Personal: para sus propios proyectos

¿A quién debemos contactar para consultar sobre esta cuenta?

Nombre completo  
David Vaquero

Número de teléfono  
+34 222-533-4444

País o región  
Estados Unidos

Dirección  
Apartamento, suite, unidad, edificio, planta, etc.

Ciudad

Estado, provincia o región

Código postal

☐ He leído y acepto los términos del [Contrato de usuario de AWS](#)

Continuar (paso 2 de 5)

Después nos pedirá  
nuestros datos  
personales, incluido  
el número de  
teléfono y el uso de  
la cuenta

Rellenaremos los  
datos, aceptamos las  
condiciones

Y pulsamos en  
Continuar

# Creación de la cuenta de AWS



**Verificación segura**

ⓘ No se cobrará el uso que esté por debajo de los límites del nivel gratuito de AWS. Podemos retener temporalmente hasta 1 USD (o una cantidad equivalente en moneda local) como transacción pendiente durante 3-5 días para verificar su identidad.



**Registrarse en AWS**

**Información de facturación**

Número de tarjeta de crédito o débito



AWS acepta todas las tarjetas de crédito y débito principales. Para obtener más información sobre las opciones de pago, consulte nuestras [preguntas frecuentes](#)

Fecha de vencimiento

Mes ▼ Año ▼

Nombre del titular de la tarjeta

Dirección de facturación

☒ Utilizar mi dirección de contacto

C/ Finisterre 19 Bajo  
Castellanos de Villiquera Salamanca 37797  
ES

☐ Utilizar una nueva dirección

**Verificar y continuar (paso 3 de 5)**

Es posible que se le redirija al sitio web de su banco para autorizar el cargo de verificación.

Después nos pedirá  
nuestros datos de la  
tarjeta de pagos,  
incluido su número,  
fecha y titular

Rellenaremos los  
datos, elegimos la  
dirección física o  
fiscal

Y pulsamos en  
Continuar

# Creación de la cuenta de AWS



**Verificación segura**

ⓘ No se cobrará el uso que esté por debajo de los límites del nivel gratuito de AWS. Podemos retener temporalmente hasta 1 USD (o una cantidad equivalente en moneda local) como transacción pendiente durante 3-5 días para verificar su identidad.



**Registrarse en AWS**

**Información de facturación**

Número de tarjeta de crédito o débito

AWS acepta todas las tarjetas de crédito y débito principales. Para obtener más información sobre las opciones de pago, consulte nuestras [preguntas frecuentes](#)

Fecha de vencimiento

Mes  Año

Nombre del titular de la tarjeta

Dirección de facturación

☒ Utilizar mi dirección de contacto

C/ Finisterre 19 Bajo  
Castellanos de Villiquera Salamanca 37797  
ES

☐ Utilizar una nueva dirección


**Verificar y continuar (paso 3 de 5)**

Es posible que se le redirija al sitio web de su banco para autorizar el cargo de verificación.

Se nos cargará 1\$ durante 3 ó 4 para verificar que la tarjeta que hemos introducido es válida

Tendremos que validar la transacción con el método definido por nuestra entidad de la tarjeta

# Creación de la cuenta de AWS



**aws**

## Registrarse en AWS

### Confirme su identidad

Para poder utilizar la cuenta de AWS, debe verificar su número de teléfono. Cuando continúe, el sistema automatizado de AWS se comunicará con usted para proporcionarle un código de verificación.

¿Cómo prefiere que le enviemos el código de verificación?

☒ Mensaje de texto (SMS)

☐ Llamada de voz

Código de país o región

España (+34) ▼

Número de teléfono móvil

Comprobación de seguridad

cf#5 r2  
ci25 12

Escriba los caracteres como se indica arriba

Enviar SMS (paso 4 de 5)

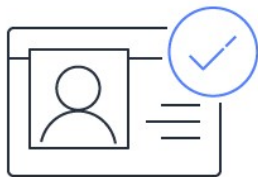
Ya por último  
deberemos validar el  
número de teléfono  
por SMS o llamada  
telefónica

Introduciremos el  
código de país o  
región, el número y  
el captcha

Pulsamos en Enviar  
(SMS)



# Creación de la cuenta de AWS



## Registrarse en AWS

### Confirme su identidad

Verificar código

**Continuar (paso 4 de 5)**

¿Tiene algún problema? A veces, se necesitan hasta 10 minutos para recibir el código de verificación. Si ha transcurrido más tiempo del mencionado, [vuelva a la página anterior](#) e inténtelo de nuevo.

Nos debería llegar un SMS o una Llamada, según lo hayamos elegido

Deberíamos introducir el código de verificación

Y pulsar en Continuar

# Creación de la cuenta de AWS



Registrarse en AWS

Seleccionar un plan de soporte

Elige un plan de soporte para su cuenta personal o empresarial. [Compare planes y ejemplos de precio](#) [Puede cambiar su plan en cualquier momento desde la consola de administración de AWS.](#)

<input checked="" type="radio"/> <b>Soporte de nivel Básico: gratis</b> <ul style="list-style-type: none"><li>Recomendado para los usuarios nuevos que recién comienzan a utilizar AWS</li><li>Acceso de autoservicio las 24 horas del día, los 7 días de la semana a los recursos de AWS</li><li>Solo para problemas de facturación y cuentas</li><li>Acceso a Personal Health Dashboard y Trusted Advisor</li></ul> 	<input type="radio"/> <b>Soporte Developer: a partir de 29 USD al mes</b> <ul style="list-style-type: none"><li>Recomendado para desarrolladores que experimentan con AWS</li><li>Acceso por correo electrónico a AWS Support durante el horario laboral</li><li>Tiempos de respuesta de 12 horas (horario laboral)</li></ul> 	<input type="radio"/> <b>Soporte Business: a partir de 100 USD al mes</b> <ul style="list-style-type: none"><li>Recomendado para ejecutar cargas de trabajo de producción en AWS</li><li>Soporte técnico las 24 horas, los 7 días de la semana por correo electrónico, teléfono y chat</li><li>Tiempos de respuesta de 1 hora</li><li>Conjunto completo de recomendaciones de prácticas de Trusted Advisor</li></ul> 
---	---	--



¿Necesita soporte de nivel Enterprise?

A partir de los 15.000 USD por mes, tendrá tiempos de respuesta de 15 minutos y una experiencia de consejería con un director técnico de cuenta asignado. [Más información](#)

Finalizar registro

Nos mostrará las diferentes alternativas de soporte

Elegimos el básico o gratuito, si no queremos pagar por ello

Pulsamos en Finalizar registro

# Creación de la cuenta de AWS



## Felicitaciones

Gracias por registrarse en AWS.

Estamos en proceso de activar su cuenta. Esto solo debe tardar unos pocos minutos. Recibirá un correo electrónico cuando se haya completado el proceso.

[Ir a la consola de administración de AWS](#)

[Regístrese para obtener otra cuenta](#) or [entre en contacto con el departamento de ventas.](#)

Con esto ya  
terminamos el  
asistente de Alta de  
Usuario en AWS

Y ya podremos  
acceder a la Consola  
de Administración de  
AWS con nuestro  
usuario

# **Datos de Acceso**

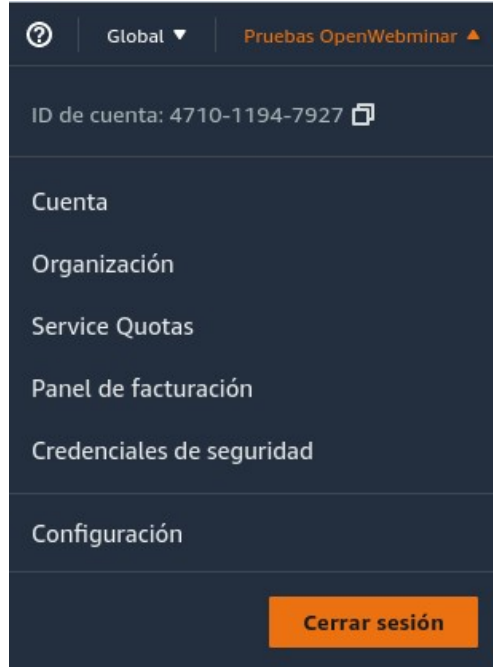
# Datos de Acceso

Si queremos usar el proveedor de TF de AWS  
necesitaremos pasarle unas credenciales para que  
pueda autenticarnos con AWS desde TF

# Datos de Acceso

Para ello debemos crear las credenciales desde IAM  
en la consola de AWS

# Datos de Acceso



Para ello debemos  
crear las  
credenciales desde  
IAM en la consola  
de AWS

# Datos de Acceso

Una vez dentro de IAM vamos al apartado de Claves de Acceso

Y pulsamos en el botón Crear una Clave de Acceso

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for 'Panel', 'Administración del acceso', 'Grupos de usuarios', 'Usuarios', 'Roles', 'Políticas', 'Proveedores de identidad', 'Configuración de cuenta', 'Informes de acceso', 'Analizador de acceso', 'Reglas de archivo', 'Analizadores', 'Configuración', 'Informe de credenciales', 'Actividad de la organización', and 'Políticas de control de servicios (SCP)'. The main content area is titled 'Sus credenciales de seguridad' and includes instructions on how to manage credentials. It features expandable sections for 'Contraseña', 'Multi-Factor Authentication (MFA)', and 'Claves de acceso (ID de clave de acceso y clave de acceso secreta)'. The 'Claves de acceso' section contains a table with columns: 'Creado', 'ID de clave de acceso', 'Último uso', and 'Última región utilizada'. Below the table is a 'Crear una clave de acceso' button. At the bottom, there are links for 'Pares de claves de CloudFront', 'certificado X.509', and 'Identificadores de cuenta'. The bottom status bar shows the account ID: 'ID de la cuenta de AWS: 471011347927'.

Identity and Access Management (IAM)

Panel

- Administración del acceso
  - Grupos de usuarios
  - Usuarios
  - Roles
  - Políticas
  - Proveedores de identidad
  - Configuración de cuenta
- Informes de acceso
  - Analizador de acceso
  - Reglas de archivo
  - Analizadores
  - Configuración
- Informe de credenciales
- Actividad de la organización
- Políticas de control de servicios (SCP)

Buscar en IAM

ID de la cuenta de AWS: 471011347927

### Sus credenciales de seguridad

Utilice esta página para administrar las credenciales de su cuenta de AWS. Para administrar las credenciales de los usuarios de AWS Identity and Access Management (IAM), utilice una lista [Consola de IAM](#).

Para obtener más información sobre los tipos de credenciales de AWS y cómo utilizarlas, consulte [Credenciales de seguridad de AWS](#) en la Referencia general de AWS.

- Contraseña
- Multi-Factor Authentication (MFA)
- Claves de acceso (ID de clave de acceso y clave de acceso secreta)

Utilice las claves de acceso para realizar llamadas mediante programación a AWS desde la CLI de AWS, las herramientas para PowerShell, los SDK de AWS o llamadas directas a la API de AWS. Puede tener 1

Para su protección, no comparta nunca las claves secretas. Como práctica recomendada, sugerimos un cambio frecuente de las claves.

La clave secreta solo se puede ver o descargar durante el proceso de creación. Cree una nueva clave de acceso si ha perdido la ya existente. [Más información](#)

Creado	ID de clave de acceso	Último uso	Última región utilizada
--------	-----------------------	------------	-------------------------

[Crear una clave de acceso](#)

Las claves de acceso del usuario raíz proporcionan acceso ilimitado a toda su cuenta de AWS. Si necesita claves de acceso a largo plazo, le recomendamos que cree un nuevo usuario de IAM con permiso

- Pares de claves de CloudFront
- certificado X.509
- Identificadores de cuenta



# Datos de Acceso

Crear una clave de acceso

✓ La clave de acceso (ID de clave de acceso y clave de acceso secreta) se ha creado correctamente.

Descargue ahora el archivo de claves, que contiene la clave de acceso secreta y el ID de clave de acceso nuevos. Si no descarga el archivo de claves ahora, no podrá recuperar la clave de acceso secreta.

Para mayor protección, almacene la clave de acceso secreta de forma segura y no la comparta.

▼ Ocultar clave de acceso

ID de clave de acceso:

AK

OU

Clave de acceso secreta:

kZ4l

Of5

Descargar archivo de claves

Cerrar

En este caso he  
anonimizado la  
captura por temas  
de seguridad  
Pero nos da dos  
valores  
El id de la clave de  
acceso  
Y la clave secreta

# Datos de Acceso

Crear una clave de acceso

✓ La clave de acceso (ID de clave de acceso y clave de acceso secreta) se ha creado correctamente.

Descargue ahora el archivo de claves, que contiene la clave de acceso secreta y el ID de clave de acceso nuevos. Si no descarga el archivo de claves ahora, no podrá recuperar la clave de acceso secreta.

Para mayor protección, almacene la clave de acceso secreta de forma segura y no la comparta.

▼ Ocultar clave de acceso

ID de clave de acceso:

AK

OU

Clave de acceso secreta:

kZ4l

Of5

Descargar archivo de claves

Cerrar

Guarda de manera segura esos valores

Por ejemplo en un gestor de contraseñas tipo BitWarden ó 1Password

No compartas esos datos con NADIE

# Datos de Acceso

Crear una clave de acceso

✓ La clave de acceso (ID de clave de acceso y clave de acceso secreta) se ha creado correctamente.

Descargue ahora el archivo de claves, que contiene la clave de acceso secreta y el ID de clave de acceso nuevos. Si no descarga el archivo de claves ahora, no podrá recuperar la clave de acceso secreta.

Para mayor protección, almacene la clave de acceso secreta de forma segura y no la comparta.

▼ Ocultar clave de acceso

ID de clave de acceso:

AK

OU

Clave de acceso secreta:

kZ4l

Of5

Descargar archivo de claves

Cerrar

Guarda de manera segura esos valores

Por ejemplo en un gestor de contraseñas tipo BitWarden ó 1Password

No compartas esos datos con NADIE

# Datos de Acceso

Crear una clave de acceso

✓ La clave de acceso (ID de clave de acceso y clave de acceso secreta) se ha creado correctamente.

Descargue ahora el archivo de claves, que contiene la clave de acceso secreta y el ID de clave de acceso nuevos. Si no descarga el archivo de claves ahora, no podrá recuperar la clave de acceso secreta.

Para mayor protección, almacene la clave de acceso secreta de forma segura y no la comparta.

▼ Ocultar clave de acceso

ID de clave de acceso: AKOU

Clave de acceso secreta: kZ4lOf5

Descargar archivo de claves

Cerrar

No dejes que nadie  
los vea  
Ni cuelgues estos  
datos en un  
repositorio Git  
Cualquiera podría  
hacer cualquier  
cosa con tu cuenta  
de AWS y gastar  
dinero de tu tarjeta

# Datos de Acceso

Crear una clave de acceso

✓ La clave de acceso (ID de clave de acceso y clave de acceso secreta) se ha creado correctamente.

Descargue ahora el archivo de claves, que contiene la clave de acceso secreta y el ID de clave de acceso nuevos. Si no descarga el archivo de claves ahora, no podrá recuperar la clave de acceso secreta.

Para mayor protección, almacene la clave de acceso secreta de forma segura y no la comparta.

▼ Ocultar clave de acceso

ID de clave de acceso: AKOU

Clave de acceso secreta: kZ4lOf5

Descargar archivo de claves

Cerrar

Como puedes ver puedes descargar también un fichero con los dos datos pulsando el botón Descargar archivo de claves

# Datos de Acceso

Ahora ya con las credenciales ya podríamos generar un fichero llamado .env donde metamos el contenido de estas credenciales

# Datos de Acceso

# Fichero .env

export AWS\_ACCESS\_KEY\_ID=KEYDEACCESODEEJEMPLO

export

AWS\_SECRET\_ACCESS\_KEY=KeySECRETAdACCESO

En el fichero .env  
guardaremos las  
variables de  
entorno

En concreto las  
credenciales de  
AWS dentro de las  
variables  
AWS\_ACCESS\_KEY y  
AWS\_SECRET\_ACCE  
SS\_KEY

# Datos de Acceso

```
# Fichero .env  
export AWS_ACCESS_KEY_ID=KEYDEACCESODEEJEMPLO  
export  
AWS_SECRET_ACCESS_KEY=KeySECRETAdACCESO
```

Edita el fichero .env  
y coloca después  
de los = cada uno  
de los valores que  
te han dado desde  
la Consola de AWS



# Datos de Acceso

```
# Fichero .env  
export AWS_ACCESS_KEY_ID=KEYDEACCESODEEJEMPLO  
export  
AWS_SECRET_ACCESS_KEY=KeySECRETAdedeACCESO
```

Es muy importante  
que estos  
ficheros .env no  
acaben nunca  
publicándose!

# Datos de Acceso

Antes de ejecutar cualquier comando de terraform como: plan, apply o destroy deberemos ejecutar:

```
source .env
```

# Datos de Acceso

En el caso de estar en Windows podríamos meter estas variables dentro de las variables de sistema

**AWS CLI**

# AWS CLI

Empezaremos realizando la instalación del AWS CLI



<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

# AWS CLI

En una distribución GNU/Linux podemos hacer la instalación ejecutando estos comandos

# AWS CLI

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

# AWS CLI

En el caso de usar Windows (\$deity no lo quiera)  
ejecutaríamos el siguiente comando con los permisos  
adecuados



# AWS CLI

```
msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

# AWS CLI

En el caso de usar MacOSX deberíamos descargar el  
PKG



<https://awscli.amazonaws.com/AWSCLIV2.pkg>

# AWS CLI

Con el Instalador de MacOSX sigue las instrucciones que aparecen en pantalla

# AWS CLI

Con todo ya instalado deberíamos ser capaces de ejecutar el comando

# AWS CLI

```
aws --version
```

# AWS CLI

Ahora sería cuando deberíamos de configurar las credenciales de acceso a AWS desde el AWS CLI

# AWS CLI

Para ello usaremos el comando

```
aws configure
```

# AWS CLI

→ ~ aws configure

AWS Access Key ID [\*\*\*\*\*702N]:

AWS Secret Access Key [\*\*\*\*\*osCa]:

Default region name [eu-west-3]: eu-south-2

Default output format [None]:



# AWS CLI

Primero nos pedirá el AWS Access Key ID, luego el  
AWS Secret Access key

# AWS CLI

Aquí deberás meter los datos correspondientes a tu usuario

# AWS CLI

Elige la Región eu-south-2 y deja en formato de salida por defecto

# AWS CLI

Para probar el funcionamiento del comando y ver que tenemos acceso a los datos de la cuenta ejecutaremos

```
aws iam list-users
```

# Creación del Proyecto

# Creación del proyecto

Con todo ya instalado podemos empezar a crear el proyecto Terraform para hacer el despliegue de las aplicaciones

# Creación del proyecto

Empezaremos creando una carpeta en nuestro disco duro por ejemplo: despliegue\_aws

# Creación del proyecto

Por ejemplo en GNU/Linux sería ejecutando el comando desde una terminal:

**`mkdir despliegue_aws`**



# Creación del proyecto

Para después meternos en dicha carpeta:

**`cd despliegue_redhat`**

# Creación del proyecto

Lo siguiente que debemos hacer es crear el primer **fichero** de terraform llamado **main.tf**

# Creación del proyecto

Para editar este tipo de ficheros debemos hacerlo con un editor de código que tengamos a mano

# Creación del proyecto

Por ejemplo podríamos usar el JetBrains IntelliJ Idea Community Edition con el plugin de Terraform

# Creación del proyecto

En las siguientes diapositivas veremos código  
Terraform

# Creación del proyecto

Empezemos viendo el fichero **main.tf**

# Creación del proyecto

```
# Fichero main.tf

provider "aws" {
  region = "eu-west-3"
}
```

En este fichero indicamos que queremos usar el proveedor de AWS y en qué región daremos de alta los recursos que definamos

# Creación del proyecto

En este caso estamos metiendo un comentario con el # y definiendo que queremos usar el proveedor de aws



# Creación del proyecto

Como comentamos anteriormente existen proveedores para AWS, Google Cloud Platform, Azure, VMware, etc..

# Creación del proyecto

En este taller usaremos el de aws



<https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

# Creación del proyecto

Cada proveedor funciona de manera diferente y tiene su propia página de documentación

# Creación del proyecto

Lo suyo sería que una vez configurado el fichero main.tf ejecutemos la inicialización de Terraform en el directorio del fichero main.tf

```
terraform init
```

# Creación del proyecto

Esto crearía las configuraciones básicas como por ejemplo la carpeta `.terraform` que incluirá todos los proveedores, plugins y demás que necesite para ejecutar el proyecto

# Creación del proyecto

Recuerda que ante cualquier modificación de los ficheros con extensión tf debería volver a ejecutar por si acaso el comando

```
terraform init
```

# Creación del proyecto

Suele ser una buena práctica crear un fichero llamado **versions.tf** donde introducimos todas las dependencias del proyecto Terraform (TF)

# Creación del proyecto

```
# Fichero versions.tf
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 5.70.0"
    }
  }
}
```

En este fichero indicamos que requerimos del proveedor de AWS, su fuente y la versión mínima de funcionamiento



# Creación del proyecto

```
→ despliegue_aws terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.70.0"...
- Installing hashicorp/aws v5.70.0...
- Installed hashicorp/aws v5.70.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Si todo va bien  
deberíamos  
poder ver una  
salida similar a  
la siguiente

# Creación del proyecto

```
→ despliegue_aws terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.70.0"...
- Installing hashicorp/aws v5.70.0...
- Installed hashicorp/aws v5.70.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Como vemos  
inicializa el  
backend  
descarga el  
proveedor

# Creación del proyecto

```
→ despliegue_aws terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.70.0"...
- Installing hashicorp/aws v5.70.0...
- Installed hashicorp/aws v5.70.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Y crea el fichero  
terraform.lock.hcl

# Creación del proyecto

Lo suyo sería empezar a guardar el código en un repositorio git

# Creación del proyecto

De esta manera podremos gestionar las versiones del código que vayamos generando

# Creación del proyecto

Inicializamos el repositorio git con el comando en la carpeta a gestionar los ficheros TF:

```
git init
```

# Creación del proyecto

A partir de este momento ya podemos empezar a gestionar los ficheros de la carpeta con git

# Creación del proyecto

Si gestionas el proyecto con git no te olvides de crear un fichero `.gitignore` para evitar que guarde lo que no queremos



# Creación del proyecto

```
# Fichero .gitignore
.env
.idea
.vscode/*
.terraform
.terraform.*
terraform.tfstate*
terraform.tfvars
```

En este fichero indicamos  
ficheros y carpetas no  
queremos que se  
guarden en el repositorio  
git

# Creación del proyecto

```
# Fichero .gitignore  
.env  
.idea  
.vscode/*  
.terraform  
.terraform.*  
terraform.tfstate*  
terraform.tfvars
```

En mi caso he incluido la  
carpeta .idea

# Creación del proyecto

```
# Fichero .gitignore
.env
.idea
.vscode/*
.terraform
.terraform.*
terraform.tfstate*
terraform.tfvars
```

Si usas VSCode  
mete .vscode/\*

# Creación del proyecto

```
# Fichero .gitignore
.env
.idea
.vscode/*
.terraform
.terraform.*
terraform.tfstate*
terraform.tfvars
```

En el fichero .env  
guardaremos las  
variables de entorno

# Creación del proyecto

Ahora ya podremos añadir los ficheros al repositorio  
git:

```
git add .gitignore main.tf versions.tf
```

# Creación del proyecto

Y por fin podremos crear nuestro primer commit:

```
git commit -m "primer commit"
```

# Creación del proyecto

Cada cambio que vayas haciendo sería bueno que añadas los ficheros e hicieras un commit nuevo para consolidar los cambios

# Creación del proyecto

Aunque Git es opcional en este taller, funcionará igual si no usas Git, es altamente recomendable si lo vas a usar TF en serio



# Creación del proyecto

Para comprobar que todo funciona de manera correcta contra AWS y que hemos configurado correctamente el AWS CLI y por lo tanto el Proveedor de Terraform para AWS

# Creación del proyecto

Ejecutaremos

`terraform plan`

# Creación del proyecto

Si todo ha ido bien dirá que ha comprobado la infraestructura y que está todo correcto

# **Base de datos**

# Base de Datos

Veamos ahora cómo crear una instancia de la bbdd  
en Lightsail

# Base de Datos

Lightsail es el servicio más sencillo de usar de AWS porque está preparado para la gente que comienza en la plataforma

# Base de Datos

En nuestro caso veremos cómo dar de alta diferentes servicios pero usando Terraform

# Base de Datos

Empecemos creando el fichero `variables_db.tf` donde definiremos las variables de entrada a nuestro proyecto



# Base de Datos

```
variable "db_name" {  
    description = "El nombre de la base de datos"  
    type        = string  
    default     = "mydatabase"  
}
```

En este caso  
empezamos  
definiendo  
una variable  
de tipo string  
llamada  
db\_name

# Base de Datos

```
variable "db_name" {  
  description = "El nombre de la base de datos"  
  type        = string  
  default     = "mydatabase"  
}
```

Este variable  
nos permitirá  
saber el  
nombre del  
recurso de  
bbdd  
generado

# Base de Datos

```
variable "db_user" {  
  description = "El nombre de usuario para la base de datos"  
  type        = string  
  default     = "admin"  
}
```

```
variable "db_password" {  
  description = "La contraseña para la base de datos"  
  type        = string  
  sensitive   = true  
  default     = "mypassword"  
}
```

Haremos lo mismo con el usuario y la contraseña de acceso

# Base de Datos

```
variable "project_name" {  
  description = "El nombre del proyecto"  
  type        = string  
  default     = "test"  
}
```

Y también  
con el  
nombre del  
proyecto

# Base de Datos

```
db_name="test"  
project_name="terraform"  
db_user="root"  
db_password="rootroot"
```

Ahora  
debertemos  
crear un fichero  
llamado  
terraform.tfvars

# Base de Datos

```
db_name="test"  
project_name="terraform"  
db_user="root"  
db_password="rootroot"
```

En este fichero  
definiremos las  
variables que  
usaremos en  
concreto en este  
proyecto

# Base de Datos

```
resource "random_string" "snapshot_name" {  
  length = 8  
  special = false  
}
```

Seguimos con el  
fichero  
database.tf

# Base de Datos

```
resource "random_string" "snapshot_name" {  
  length = 8  
  special = false  
}
```

En este caso  
definimos un  
recurso random  
para los  
nombres de  
recursos a  
generar



# Base de Datos

```
resource "awsLightsailDatabase" "test" {
  relational_database_name = var.db_name
  availability_zone        = "eu-west-3a"
  master_database_name     = var.db_name
  master_password          = var.db_password
  master_username          = var.db_user
  # aws lightsail get-relational-database-blueprints
  blueprint_id             = "mysql_8_0"
  # aws lightsail get-relational-database-bundles
  bundle_id                = "micro_2_0"
  publicly_accessible      = true
  # Esto nos sirve para el taller pero en producción no deberíamos ponerlo
  skip_final_snapshot      = true
  # en producción deberíamos crear una snapshot antes de borrar la bbdd
  final_snapshot_name      = "${var.db_name}-snap-${
    random_string.snapshot_name.result}"
}
```

Definimos el  
recurso de  
lightsail para  
database

# Base de Datos

```
resource "aws_lightsail_database" "test" {
  relational_database_name = var.db_name
  availability_zone        = "eu-west-3a"
  master_database_name     = var.db_name
  master_password          = var.db_password
  master_username          = var.db_user
  # aws lightsail get-relational-database-blueprints
  blueprint_id             = "mysql_8_0"
  # aws lightsail get-relational-database-bundles
  bundle_id                = "micro_2_0"
  publicly_accessible      = true
  # Esto nos sirve para el taller pero en producción no deberíamos ponerlo
  skip_final_snapshot      = true
  # en producción deberíamos crear una snapshot antes de borrar la bbdd
  final_snapshot_name      = "${var.db_name}-snap-${
    random_string.snapshot_name.result}"
}
```

Definimos  
nombre de  
bbdd, zona de  
disponibilidad  
usuario,  
contraseña

# Base de Datos

```
resource "aws_lightsail_database" "test" {
  relational_database_name = var.db_name
  availability_zone        = "eu-west-3a"
  master_database_name     = var.db_name
  master_password          = var.db_password
  master_username          = var.db_user
  # aws lightsail get-relational-database-blueprints
  blueprint_id             = "mysql_8_0"
  # aws lightsail get-relational-database-bundles
  bundle_id                = "micro_2_0"
  publicly_accessible      = true
  # Esto nos sirve para el taller pero en producción no deberíamos ponerlo
  skip_final_snapshot      = true
  # en producción deberíamos crear una snapshot antes de borrar la bbdd
  final_snapshot_name      = "${var.db_name}-snap-${
    random_string.snapshot_name.result}"
}
```

Metemos  
también el  
motor de bbdd  
que queremos  
usar y el tipo de  
instancia

# Base de Datos

```
output "db_instance_fqdn" {  
  value = awsLightsailDatabase.test.master_endpoint_address  
}  
output "db_instance_port" {  
  value = awsLightsailDatabase.test.master_endpoint_port  
}
```

Ahora en el  
fichero  
outputs.tf  
definiremos las  
salidas de  
terraform

# Base de Datos

```
output "db_instance_fqdn" {  
  value = awsLightsailDatabase.test.master_endpoint_address  
}  
output "db_instance_port" {  
  value = awsLightsailDatabase.test.master_endpoint_port  
}
```

Aquí veremos el nombre fqdn del servicio de bbdd y el puerto

# Base de Datos

Ahora es cuando deberíamos aplicar los cambios con  
`terraform apply`

# Base de Datos

Si todo ha ido bien y en unos 10 minutos deberíamos de disponer ya de la bbdd en Lightsail y debería ofrecernos una salida con los parámetros

## **Datos de Contacto**

**david@cursosdedesarrollo.com**

**+34 656 66 14 78**



**Licencia**  
**CC/BY/SA/NC**  
**David Vaquero Santiago**

**<https://creativecommons.org/licenses/by-nc-sa/3.0/es/deed.es>**