



Clase 34. Programación Backend

Productos Cloud: AWS



OBJETIVOS DE LA CLASE

- Presentar AWS (Amazon Web Services) y crear una cuenta.
- Recorrer e implementar a partir de la plataforma Elastic Beanstalk de AWS.
- Desarrollar un proyecto de Node en la plataforma de AWS.

CRONOGRAMA DEL CURSO

Clase 33



Productos cloud: Heroku

Clase 34



Productos cloud: AWS

Clase 35



Envío de mensajes y
Seguridad

PLATAFORMA AWS



¿Qué es?

- Amazon Web Services (AWS) es una plataforma en la nube muy adoptada y completa.
- Ofrece más de 200 servicios integrales de centros de datos a nivel global. Muchas empresas lo utilizan y con esto reducen los costos, aumentan su agilidad e innovan de forma más rápida.



En esta clase utilizaremos la **plataforma *Elastic Beanstalk*** de AWS para implementar aplicaciones NodeJS en la nube.

👉 Ten en cuenta que al crearnos una cuenta en AWS nos piden una tarjeta de crédito. Sin embargo, si hacemos las pruebas y luego lo borramos será sin cargo.



Para crear una cuenta en AWS, la plataforma solicitará los datos de una tarjeta de crédito.

Si hacemos las pruebas y luego lo borramos no generará cargos. De todos modos, a fines de este, curso *no* será *obligatorio* que lo hagas.

👉 Si decides no hacerlo, presta atención al proceso porque puede resultarte sumamente enriquecedor a futuro.



Plataforma *Elastic Beanstalk*



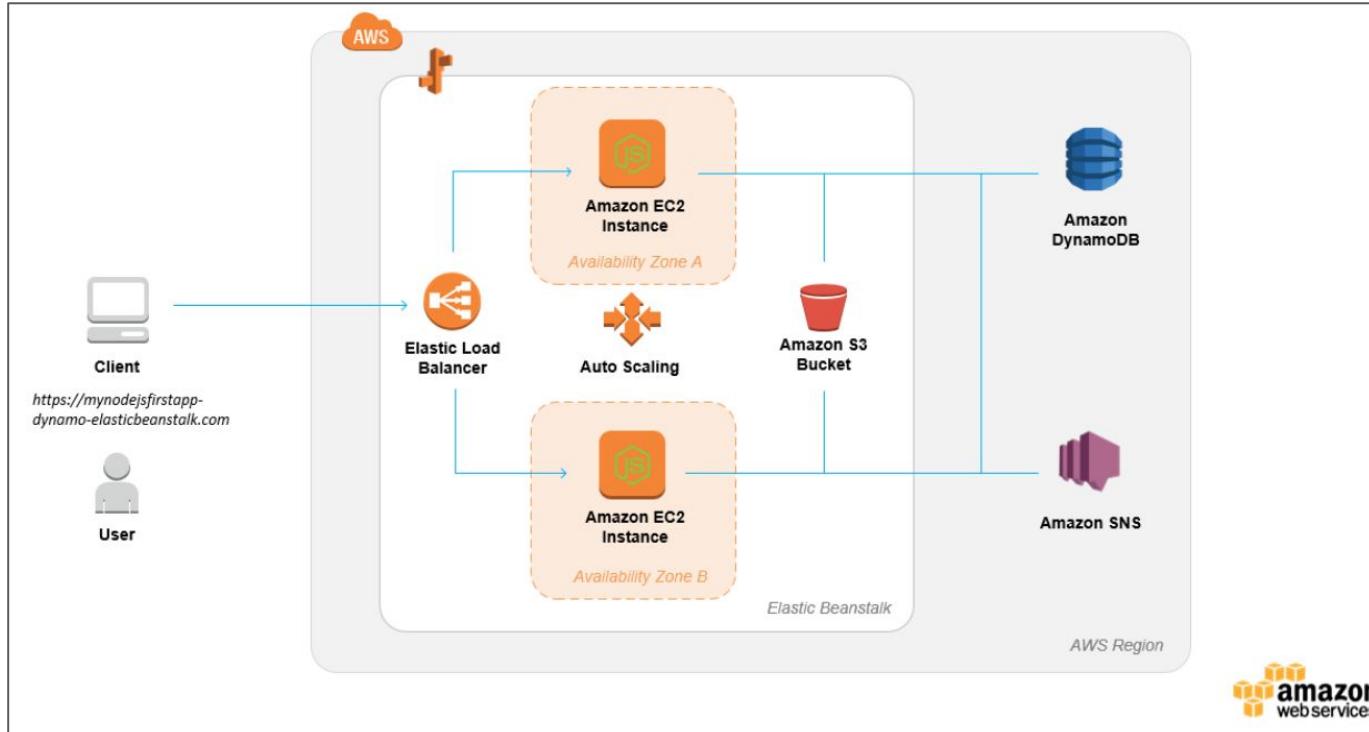
- Administra de manera automática la implementación de nuestra aplicación (desde el aprovisionamiento de la capacidad, el balanceo de carga y el auto escalamiento hasta la monitorización del estado) ingresando únicamente el código.
- Ajusta el escalado de la aplicación automáticamente en función de las necesidades específicas de las aplicaciones. Para ello utiliza una configuración de *Auto Scaling* que se puede adaptar con facilidad.



Usaremos también **Amazon DynamoDB** que es un servicio de base de datos NoSQL rápido y flexible, completamente administrado en la nube, compatible con modelos de almacenamiento de valor de clave y de documentos.



Diagrama de arquitectura de Elastic Beanstalk



CODER HOUSE

CREAR NUESTRA CUENTA EN AWS



Crear cuenta en AWS

Ejemplo
en vivo



- Para crear una cuenta en AWS entramos a <https://aws.amazon.com/es/>

The screenshot shows the AWS homepage in Spanish. At the top right, there is a red rectangular box highlighting the 'Cree una cuenta de AWS' (Create an AWS account) button. The main content area features a large orange banner titled 'Proyectos en AWS' with the heading 'Implementar una aplicación web de Node.js'. Below the heading, it says 'Lance y ejecute una aplicación web de Node.js con alta disponibilidad en AWS'. A 'Comenzar con el proyecto' button is visible, along with a note indicating '6 pasos | 60 minutos'. At the bottom of the banner, there are three tabs: 'Información general', 'Servicios y costos', and 'Preguntas frecuentes'. A descriptive paragraph at the bottom explains the project's goal: 'En este proyecto, aprenderá a implementar una aplicación web de Node.js de alta disponibilidad con AWS Elastic Beanstalk y Amazon DynamoDB. La aplicación de muestra que implementará utilizará Node.js, Express y una base de datos NoSQL. Si utiliza Elastic Beanstalk, solo tiene que cargar el código'.



Crear cuenta en AWS

Ejemplo
en vivo



- Completamos el mail, contraseña y elegimos el nombre de la cuenta. Luego continuamos al siguiente paso.

aws

Registrarse en AWS

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

Para obtener más información, visite aws.amazon.com/free.



Dirección de correo electrónico
Utilizará esta dirección de correo electrónico para iniciar sesión en su nueva cuenta de AWS.

Contraseña

Confirmar la contraseña

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

Continuar (paso 1 de 5)

[Iniciar sesión en una cuenta de AWS existente](#)



Crear cuenta en AWS

Ejemplo
en vivo



- Completamos nuestros datos de nombre y dirección y continuamos.

aws

Registrarse en 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

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

Número de teléfono

Introduzca el código de país y el número de teléfono.

+1 222-333-4444

País o región

Argentina

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)



Crear cuenta en AWS

Ejemplo
en vivo



- Luego, debemos poner los datos de una tarjeta de crédito. Pero si los proyectos que implementamos luego los eliminamos, no nos cobran nada.

Verificación segura

ⓘ No cobraremos el uso que esté por debajo de los límites de la capa gratuita de AWS. Retenemos temporalmente 1 USD/EUR como transacción pendiente por un periodo de 3 a 5 días para verificar su identidad.



Registrarse en AWS

Información de facturación

Número de tarjeta de crédito o débito

VISA MASTERCARD AMEX DISCOVER

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

Araoz 276
Buenos Aires Buenos Aires 1414
AR

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.



Crear cuenta en AWS

Ejemplo
en vivo



- Ahora debemos poner nuestro número de teléfono para que nos pueda enviar AWS un SMS como verificación de identidad.
- Luego, ponemos el código de verificación que nos llega por SMS y continuamos.

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.

Código de país o región

Argentina (+54)

Número de teléfono móvil

11

Comprobación de seguridad

353nap

Escriba los caracteres como se indica arriba

Enviar SMS (paso 4 de 5)

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 intételo de nuevo](#).



Crear cuenta en AWS

Ejemplo
en vivo



- Finalmente, elegimos el plan de soporte que queremos, en este caso el gratuito, y luego finalizamos el registro.

Registrarse en AWS

Seleccionar un plan de soporte

Elija 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.

Soporte de nivel Basic: gratis

- Recomendado para los usuarios nuevos que recién comienzan a utilizar AWS
- Acceso de autoservicio las 24 horas del día, los 7 días de la semana a los recursos de AWS
- Solo para problemas de facturación y cuentas
- Acceso a Personal Health Dashboard y Trusted Advisor

Soporte Developer: a partir de 29 USD al mes

- Recomendado para desarrolladores que experimentan con AWS
- Acceso por correo electrónico a AWS Support durante el horario laboral
- Tiempos de respuesta de 12 horas (horario laboral)

Soporte Business: a partir de 100 USD al mes

- Recomendado para ejecutar cargas de trabajo de producción en AWS
- Soporte técnico las 24 horas, los 7 días de la semana por correo electrónico, teléfono y chat
- Tiempos de respuesta de 1 hora
- Conjunto completo de recomendaciones de prácticas de Trusted Advisor

¿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 consejo con un director técnico de cuenta asignado. [Más información](#)

Finalizar registro



Crear cuenta en AWS

Ejemplo
en vivo



- De esta forma nos queda creada la cuenta, ya podemos seguir para crear una App en AWS.
- Nos llega también un mail de bienvenida a AWS.

The screenshot shows the AWS homepage with a dark header. The header includes the AWS logo, navigation links like 'Productos', 'Soluciones', 'Precios', etc., and a prominent orange button 'Inicie sesión en la consola'. Below the header, there's a large central area featuring a rocket launching from a cloud icon, followed by the text '¡Felicitaciones!' and 'Gracias por registrarse en AWS.' A message below states: 'Estamos activando la cuenta. Tardaremos solo unos minutos. Recibirá un correo electrónico cuando hayamos acabado.' At the bottom, there are two buttons: a yellow one labeled 'Ir a la consola de administración de AWS' and a blue one labeled 'Iniciar sesión en otra cuenta o Entrar en contacto con el departamento de ventas'.

CREAR NUESTRA APP EN ELASTIC BEANSTALK

Ejemplo
en vivo



Crear App en Elastic Beanstalk

- Para comenzar la implementación debemos iniciar sesión en la consola de AWS entrando al siguiente link [Consola AWS](#)

The screenshot shows the AWS console homepage with a specific project highlighted. The project title is "Implementar una aplicación web de Node.js". Below the title, it says "Lance y ejecute una aplicación web de Node.js con alta disponibilidad en AWS". A "Comenzar con el proyecto" button is visible. To the right, there's an illustration of a laptop with a JS logo and a wrench. At the bottom, there are tabs for "Información general", "Servicios y costos", and "Preguntas frecuentes". A detailed description of the project follows:

En este proyecto, aprenderá a implementar una aplicación web de Node.js de alta disponibilidad con AWS Elastic Beanstalk y Amazon DynamoDB. La aplicación de muestra que implementará utiliza Node.js, Express y una base de datos NoSQL. Si utiliza Elastic Beanstalk, solo tiene que cargar el código y Elastic Beanstalk administrará de manera automática la implementación, desde el aprovisionamiento de la capacidad, el equilibrio de carga y el escalado automático hasta la monitorización del estado de la aplicación. Elastic Beanstalk ajusta el escalado de la aplicación automáticamente en función de la necesidad específica de la aplicación, para lo que usa una configuración de Auto Scaling que se puede adaptar con facilidad. Amazon

Crear App en Elastic Beanstalk

Ejemplo
en vivo



- Luego, elegimos la opción de usuario raíz, ponemos mail y contraseña e ingresamos.

The image shows two screenshots of the AWS sign-in interface. A large grey arrow points from the first screenshot on the left to the second screenshot on the right.

Screenshot 1 (Left): The "Iniciar sesión" (Sign In) page. It features the AWS logo and a "Backup y restauración con AWS" (AWS Backup and Recovery) banner. The "Usuario raíz" (Root user) option is selected, and the "Correo electrónico" (Email address) field contains "hombradeusuario@ejemplo.com". Below the form, there's a "Siguiente" (Next) button and a note about accepting the AWS Customer Agreement. At the bottom, there's a "Create a new AWS account" link.

Screenshot 2 (Right): The "Inicio de sesión de usuarios de cuentas raíces" (Root user sign-in) page. It shows the same "Backup y restauración con AWS" banner. The "Contraseña" (Password) field is filled with a masked password. Below it is an "Iniciar sesión" (Sign In) button. There are also links for "Iniciar sesión con una cuenta diferente" (Sign in with a different account) and "Crear una cuenta de AWS" (Create a new AWS account). The bottom of the page includes a copyright notice for Amazon Web Services and a "Español" (Spanish) language link.



Crear App en Elastic Beanstalk

- En la consola de AWS elegimos la opción de Elastic Beanstalk y luego vamos al botón de crear aplicación.

The screenshot shows the AWS Management Console interface. On the left, the 'Servicios de AWS' sidebar has 'Elastic Beanstalk' highlighted with a red box. A large arrow points from this sidebar to the main content area, which displays the 'AWS Elastic Beanstalk' landing page. This page also features a red box around the 'Create Application' button.

Consola de administración de AWS

Manténgase conectado a sus recursos de AWS en cualquier lugar

Elastic Beanstalk

Explore AWS

Cree una solución

Lance una máquina virtual

Diseñe con servidores virtuales

Ejecute contenedores sin servidores con AWS Fargate

AWS Elastic Beanstalk

Administración integral de aplicaciones web.

Empiece

Implemente fácilmente su aplicación web en minutos de inicio.

Create Application

Cómo funciona

Solo tiene que cargar el código y Elastic Beanstalk administrará automáticamente la implementación, desde el aprovisionamiento de capacidad, el balanceo de carga y el escalado automático hasta el monitoreo del estado de la aplicación web con actualizaciones continuas de seguridad y parches totalmente administrados. Más información

Precios

No se aplican cargos adicionales por Elastic Beanstalk. Pague por los recursos de AWS que creamos para almacenar y ejecutar su aplicación web, como los buckets de Amazon S3 y las instancias de Amazon EC2.

Introducción

Configurar App en Elastic Beanstalk

Ejemplo
en vivo



- Elegimos el nombre de la aplicación, en este caso serveraws-ch y luego la plataforma que se corresponde al lenguaje utilizado, que es NodeJs.

The screenshot shows the 'Create a web application' wizard in the AWS Elastic Beanstalk console. In the 'Information about the application' section, the 'Name of the application' input field is highlighted with a red box and contains the value 'serveraws-ch'. Below this, there's a section for 'Application labels' with a note that up to 50 labels can be applied. A large grey arrow points from the left side of the first screenshot to the right side of the second screenshot.

Screenshot 1 (Left): Create a web application - Step 1 of 3

- Entornos**
- Aplicaciones**
- Historial de cambios**

Crear una aplicación web

Cree una nueva aplicación y un nuevo entorno con una aplicación de ejemplo o su propio código. Al crear un entorno, permite que AWS Elastic Beanstalk administre los recursos de AWS y los permisos en su nombre. [Más información](#)

Información de la aplicación

Nombre de la aplicación
serveraws-ch

Aplicar hasta 50 etiquetas. Puede utilizar las etiquetas para agrupar y filtrar sus recursos. Una etiqueta es un par clave-valor. La clave debe ser única en el recurso y distingue entre mayúsculas y minúsculas. [Más información](#)

Etiquetas de la aplicación

Clave Valor [Quitar etiqueta](#)

[Comentarios](#) [Español](#)

© 2008 - 2021, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. [Política de privacidad](#) [Términos de uso](#) [Preferencias de cookies](#)

Screenshot 2 (Right): Create a web application - Step 2 of 3

- Entornos**
- Aplicaciones**
- Historial de cambios**

Plataforma

Plataforma
-- Elegir una plataforma --
.NET Core on Linux
.NET on Windows Server
Docker
GlassFish
Go
Java
Node.js
PHP
Python
Ruby

[Cargar el código](#)
Cargar un conjunto de fuentes del equipo o copiar uno de Amazon S3.

[Comentarios](#) [Español](#)

© 2008 - 2021, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. [Política de privacidad](#) [Términos de uso](#) [Preferencias de cookies](#)

Configurar App en Elastic Beanstalk

Ejemplo
en vivo



- En código de aplicación seleccionamos aplicación de prueba y luego vamos a configurar más opciones y elegimos la configuración personalizada.

The screenshot shows the AWS Elastic Beanstalk configuration interface. It consists of two side-by-side panels connected by a large right-pointing arrow.

Left Panel (Initial Configuration):

- Header:** AWS Services, Buscar servicios, características, productos del Marketplace y documentos [Alt+S], backendNodeAWS, Ohio, Soporte.
- Left Sidebar:** Servicios, Entornos, Aplicaciones, Historial de cambios.
- Current View:** Elastic Beanstalk > Introducción.
- Section:** Configurar Serverch-env.
- Sub-section:** Elementos preestablecidos.
- Content:** Comience a partir de un conjunto predefinido que coincida con su caso de uso o elija Configuración personalizada para eliminar los valores recomendados y usar los valores predeterminados del servicio.
- Options:** Modelos de configuración preestablecidos:
 - Instancia única (*compatible con la capa gratuita*)
 - Instancia única (mediante instancia de spot)
 - Alta disponibilidad
 - Alta disponibilidad (con instancias de spot y a petición)
- Selected Option:** Configuración personalizada (highlighted with a red border).
- Plataforma:** Node.js 14 running on 64bit Amazon Linux 2/5.3.2.

Right Panel (Advanced Configuration):

- Header:** AWS Services, Buscar servicios, características, productos del Marketplace y documentos [Alt+S], backendNodeAWS, Ohio, Soporte.
- Left Sidebar:** Servicios, Entornos, Aplicaciones, Historial de cambios.
- Current View:** Elastic Beanstalk > Introducción.
- Section:** Plataforma.
- Content:** Plataforma: Node.js, Ramificación de la plataforma: Node.js 14 running on 64bit Amazon Linux 2, Versión de la plataforma: 5.3.2 (Recommended).
- Section:** Código de la aplicación.
- Content:** Aplicación de muestra (selected radio button), Cargar el código (radio button).
- Buttons:** Cancelar, Configurar más opciones (highlighted with a red border), Crear una aplicación.



Configurar App en Elastic Beanstalk

- Luego, modificamos la capacidad, configurando el tipo de entorno como carga balanceada. En el balanceador de carga configuramos las instancias con un mínimo de 2 y un máximo de 4.

The image shows two screenshots of the AWS Elastic Beanstalk console. The left screenshot displays the application configuration page for 'backendNodeAWS' in the 'Ohio' region. It shows the platform as 'Node.js 14 running on 64bit Amazon Linux 2/5.3.2'. The 'Capacidad' (Capacity) section is highlighted with a red box, showing settings for 'Instancia individual' (Instance Type: t2.micro, Image ID: ami-01e24c3b80b2f5882) and 'Carga balanceada' (Balanced Load). An arrow points from this section to the right screenshot. The right screenshot shows the 'Modificar la capacidad' (Change Capacity) dialog box. In the 'Grupo de Auto Scaling' (Auto Scaling Group) section, the 'Instancia' (Instances) settings are highlighted with a red box, showing 'Min.' set to 2 and 'Máx.' set to 4. Below this, there are options for 'Composición de la flota' (fleet composition) and 'Instancias a petición' (Request Instances) or 'Combinar opciones de compra e instancias' (Combine purchase and instance options).

Configurar App en Elastic Beanstalk

Ejemplo
en vivo



- Vamos hasta abajo de la página y clickeamos el botón de crear una aplicación con estos cambios que hicimos en la configuración.

The screenshot shows the AWS Elastic Beanstalk configuration interface with two pages of settings:

Left Page (Visible Content):

- Software:** AWS X-Ray: deshabilitada, Rotar registros: deshabilitada (por defecto), Transmisión de registros: deshabilitado (por defecto), Propiedades del entorno: 0.
- Instancias:** IMDSv1: deshabilitado, Tipo de volumen raíz: predeterminado de contenedor, Tamaño del volumen raíz (GB): predeterminado de contenedor, IOPS del volumen raíz: predeterminado de contenedor, Grupos de seguridad: ninguno.
- Capacidad:** Tipo de entorno: cargar balance, Auto Scaling, Zonas de disponibilidad: Cualquiera, Composición de la flota: Instancias a petición.
- Balanceador de carga:** Tipo de balanceador de carga: application, Compartido: false, Agentes de escucha: 1, Procesos: 1, Reglas: 0.
- Actualizaciones e implementaciones continuas:** Política de implementación: Todo a la vez, Actualizaciones continuas: deshabilitada, Comprobación de estado: habilitado.
- Seguridad:** Rol de servicio: arn:aws:iam::791724667592:role/aws-elasticbeanstalk-service-role, Perfil de instancia de la máquina virtual: --, Perfil de instancia de la máquina virtual: aws-elasticbeanstalk-ec2-role.

Right Page (Visible Content):

- Monitorización:** Sistema de informes de estado: Mejorado, Omitir aplicación 4xx: deshabilitado, Omitir balancinador de carga 4xx: deshabilitado, Transmisión de registros de eventos de estado: deshabilitado.
- Actualizaciones administradas:** Actualizaciones administradas: habilitado, Período de actualización semanal: Sat:05:00 UTC.
- Notificaciones:** Dirección de correo electrónico: --.
- Red:** Este entorno no forma parte de una VPC.
- Base de datos:** Motor: --, Clase de instancia: --, Almacenamiento (GB): --, Multi-AZ: --.
- Etiquetas:** Etiquetas: ninguno.

Bottom Navigation: Comentarios, Español, © 2008 - 2021, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Política de privacidad, Términos de uso, Preferencias de cookies.

Bottom Right Buttons: Cancelar, Anterior, **Crear una aplicación** (highlighted with a red box).



Crear App en Elastic Beanstalk

- Elastic Beanstalk suele tardar unos minutos en crear el entorno y cada uno los recursos que administra.

The screenshot shows the AWS Elastic Beanstalk console with the following details:

- Servicios:** Servicios ▾
- Elastic Beanstalk:** Entornos
- Entornos:** BackendNodeAWS, Ohio, Soporte ▾
- Entorno:** Serverch-env
- Proceso:** Creando Serverch-env. Este proceso tardará unos minutos. ...
- Log de creación:**

 - 1:53pm Created CloudWatch alarm named: awseb-e-zr7mpc4str-stack-AWSEBCloudwatchAlarmLow-AAQIMY3JAYQG
 - 1:53pm Created CloudWatch alarm named: awseb-e-zr7mpc4str-stack-AWSEBCloudwatchAlarmHigh-1KCMRLNZ3YEN3
 - 1:53pm Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-2:791724667592:scalingPolicy:30b96ae-371b-4874-8958-832d5701f870:autoScalingGroupName/awseb-e-zr7mpc4str-stack-AWSEBAutoScalingGroup-1TW6HJUTJJYBT:policyName/awseb-e-zr7mpc4str-stack-AWSEBAutoScalingScaleUpPolicy-E1N4KFk3F265
 - 1:53pm Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-2:791724667592:scalingPolicy:ae384d33-4f7e-4a72-ad7-e5d3fa256d19:autoScalingGroupName/awseb-e-zr7mpc4str-stack-AWSEBAutoScalingGroup-1TW6HJUTJJYBT:policyName/awseb-e-zr7mpc4str-stack-AWSEBAutoScalingScaleDownPolicy-QPFRWC3WQGZE
 - 1:53pm Waiting for EC2 instances to launch. This may take a few minutes.
 - 1:53pm Created Auto Scaling group named: awseb-e-zr7mpc4str-stack-AWSEBAutoScalingGroup-1TW6HJUTJJYBT
 - 1:52pm Created security group named: awseb-e-zr7mpc4str-stack-AWSEBSecurityGroup-1GHV1HNLWU229
 - 1:51pm Created security group named: sg-03317617228e4d638



Elastic Branstalk

cuenta con una serie de recursos:



- **Instancia EC2**
- **Grupo de seguridad de la instancia**
- **Balanceador de carga**
- **Grupo de seguridad del balanceador de carga**
- **Grupo de Auto Scaling**
- **Bucket de Amazon S3**
- **Alarmas de Amazon CloudWatch**
- **Scaling aumenta o reduce los recursos**
- **Pila de AWS CloudFormation**
- **Nombre de dominio**

¡Veámoslos en detalle!



Recursos del entorno de Elastic Beanstalk



- ***Instancia EC2:***

Máquina virtual de Amazon Elastic Compute Cloud (Amazon EC2) configurada para ejecutar aplicaciones web en la plataforma que elija. Cada plataforma ejecuta un conjunto específico de software, archivos de configuración y scripts compatibles con una determinada versión de lenguaje, marco de trabajo y contenedor web (o una combinación de estos). La mayoría de las plataformas utilizan Apache o nginx como un proxy inverso que se sitúa delante de la aplicación web, reenvía las solicitudes a esta, administra los recursos estáticos y genera registros de acceso y errores.



Recursos del entorno de Elastic Beanstalk



- **Grupo de seguridad de la instancia:**

Grupo de seguridad de Amazon EC2 configurado para permitir el tráfico entrante en el puerto 80. Este recurso permite que el tráfico HTTP procedente del balanceador de carga llegue a la instancia EC2 en la que se ejecuta la aplicación web. De forma predeterminada, el tráfico no está permitido en otros puertos.

- **Balanceador de carga:**

Balanceador de carga de Elastic Load Balancing que está configurado para distribuir solicitudes a las instancias que se ejecutan en la aplicación. También permiten que las instancias no estén expuestas directamente a Internet.

- **Grupo de seguridad del balanceador de carga:**

Grupo de seguridad como el de la instancia.



Recursos del entorno de Elastic Beanstalk



- ***Grupo de Auto Scaling:***

Está configurado para reemplazar una instancia si termina o deja de estar disponible.

- ***Bucket de Amazon S3:***

Ubicación de almacenamiento para el código fuente, los registros y otros artefactos que se crean al utilizar Elastic Beanstalk.

- ***Alarmas de Amazon CloudWatch:***

Dos alarmas de CloudWatch que monitorean la carga recibida por las instancias y que se activan si la carga es demasiado alta o demasiado baja. Cuando se activa una alarma, en respuesta, el grupo de Auto Scaling aumenta o reduce los recursos.



Recursos del entorno de Elastic Beanstalk

Ejemplo
en vivo



- **Plataforma de AWS CloudFormation:**

Elastic Beanstalk utiliza AWS CloudFormation para lanzar los recursos del entorno y propagar los cambios de configuración.

- **Nombre de dominio:**

Nombre de dominio que dirige el tráfico a la aplicación web con el formato subdominio.region.elasticbeanstalk.com.



Configuración de nuestra App

- Una vez finalizada la instalación, haciendo click en el link seleccionado ya podemos ir a nuestra aplicación, donde veremos una ventana como la mostrada en la imagen.

The screenshot shows the AWS Elastic Beanstalk console for the 'Serverch-env' environment. The URL in the address bar is <https://us-east-2.console.aws.amazon.com/elasticbeanstalk/home/?region=us-east-2#/environment/dashboard?application=serverch-env&environmentName=Serverch-env>. The main area displays the application 'Sample Application' running on 'node.js'. A red box highlights the URL 'Serverch-env.aba-bywb6ppq.us-east-2.elasticbeanstalk.com' in the browser's address bar. On the left sidebar, under the 'Serverch-env' section, there is a link labeled 'Ir al entorno' with a small icon.

The screenshot shows a web browser displaying the deployed application at the URL 'Serverch-env.aba-bywb6ppq.us-east-2.elasticbeanstalk.com'. The page has a green header with the text 'Congratulations' and a message: 'Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud'. Below this, it says 'This environment is launched with Elastic Beanstalk Node.js Platform'. To the right, there is a 'What's Next?' section with a list of links related to AWS Elastic Beanstalk.

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy an Express Application to AWS Elastic Beanstalk
- Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Deploy a Geodjango Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

CONFIGURACIÓN DE NUESTRA APP



Configuración de nuestra App

- Vamos a Servicios > IAM > Roles, seleccionamos el rol asociado a instancias EC2, y le asociamos las siguientes políticas para usar base de datos y servicios de mensajería:
 - AmazonDynamoDBFullAccess
 - AmazonSNSFullAccess

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, the 'Roles' section is selected, displaying a list of roles. One role, 'aws-elasticbeanstalk-ec2-role', is highlighted with a red box. On the right, the 'Permissions' tab of the role's configuration page is shown. Under the 'Asociar políticas' (Associate Policies) section, two policies are listed and highlighted with red boxes: 'AmazonDynamoDBFullAccess' and 'AmazonSNSFullAccess'. The rest of the page includes tabs for 'Relaciones de confianza' (Trust Relationships), 'Etiquetas' (Tags), 'Access Advisor', and 'Revocar las sesiones' (Revoke Sessions). The top navigation bar shows the session name 'backendNodeAWS'.

Configuración de nuestra App - DynamoDB

Ejemplo
en vivo



- Luego, ingresamos a Servicios > DynamoDB > Tablas. Ahí creamos las tablas que vamos a necesitar en nuestra base de datos para la aplicación. Elegimos el nombre (en este caso: 'product-inventory'), la clave primaria ('productId') que figura como *partition key*, y clickeamos el botón de crear.

The image consists of three side-by-side screenshots of the AWS DynamoDB console. The first screenshot shows the 'Tables' page with a red box around the 'crear tabla' (Create Table) button. The second screenshot shows the 'Create a DynamoDB table' wizard, step 1: Set the table name, where 'Nombre' is set to 'registro-usuarios' and 'Clave principal' is set to 'email'. The third screenshot shows the continuation of the wizard, step 2: Configure the table, with 'Usar la configuración predeterminada' checked. Both the second and third screenshots have a red box around the 'Create' button at the bottom right.

CODER HOUSE

Configuración de nuestra App - DynamoDB

Ejemplo
en vivo



- Una vez creada la tabla, ya está lista para crear elementos en ella.

The screenshot shows two side-by-side views of the AWS DynamoDB console.

Left Panel: Shows the 'Crear tabla' (Create Table) wizard for a new table named 'registro-usuarios'. It includes sections for 'Información general' (General Information), 'Elementos' (Items), 'Métricas' (Metrics), 'Alarms' (Alarms), 'Capacidad' (Capacity), 'Índices' (Indices), and 'Tablas globales' (Global Tables). A progress bar indicates 'Se está creando la tabla' (Creating the table).

Right Panel: Shows the 'Crear elemento' (Create Item) wizard for the 'registro-usuarios' table. It includes sections for 'Información general' (General Information), 'Elementos' (Items), 'Métricas' (Metrics), 'Alarms' (Alarms), 'Capacidad' (Capacity), 'Índices' (Indices), and 'Tablas globales' (Global Tables). A progress bar indicates 'Examen: [Tabla] tabla_usuarios: email' (Examining: [Table] tabla_usuarios: email) and 'Mostrando 0 de 0 elementos' (Showing 0 of 0 elements). A search bar at the bottom has 'email' typed into it.

Configuración de nuestra App - SNS

Ejemplo
en vivo

- Luego vamos Servicios -> SNS -> Tópicos y creamos un tópico nuevo: notificaciones

The screenshot shows the AWS SNS service configuration interface. On the left, there's a sidebar with navigation links: Dashboard, Topics (which is selected and highlighted in blue), Subscriptions, Mobile (with Push notifications, Text messaging (SMS), and Origination numbers), and Marketplace. The main content area has a search bar at the top with the text 'sns'. Below it, a 'Services' section lists 'Simple Notification Service' (selected and highlighted with a red box) and other options like Features, Documentation, and Knowledge Articles. To the right, a 'Create topic' dialog box is open, prompting for a 'Topic name'. The input field contains the value 'notificaciones' (highlighted with a red box). At the bottom of the dialog is a large orange 'Next step' button (also highlighted with a red box).

Configuración de nuestra App - SNS

Ejemplo
en vivo

- Elegimos la configuración estándar y aceptamos al final de la página.

The screenshot shows the 'Create topic' page in the Amazon SNS console. At the top, there is a section titled 'Delivery retry policy (HTTP/S) - optional' with a note: 'The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section.' Below this, the 'Details' section is visible, containing fields for 'Name' (set to 'notificaciones') and 'Type' (set to 'Info'). Under 'Type', it says 'Topic type cannot be modified after topic is created'. There are two options: 'FIFO (first-in, first-out)' and 'Standard'. The 'Standard' option is selected and highlighted with a red box. It lists the following features:

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

At the bottom right of the page, there are 'Cancel' and 'Create topic' buttons, with the 'Create topic' button also highlighted with a red box.



Configuración de nuestra App - SNS

- Debería aparecer una confirmación de la creación del tópico, con un ARN asociado. Debemos copiar ese código para usarlo más adelante:

The screenshot shows a green success message box at the top: "Topic notificaciones created successfully. You can create subscriptions and send messages to them from this topic." Below this, the navigation path is "Amazon SNS > Topics > notificaciones". The main title is "notificaciones". A "Details" section displays the following information:

Name	Display name
notificaciones	-
ARN	Topic owner
arn:aws:sns:us-east-1:786671515187:notificaciones	786671515187
Type	
Standard	

The ARN value is highlighted with a red rectangle.

Configuración de nuestra App - SNS

Ejemplo
en vivo



- Luego vamos a suscripciones, y creamos una nueva suscripción. Esto nos permitirá recibir notificaciones cada vez que se publique un nuevo evento en el tópico creado:

The screenshot shows the Amazon SNS console. On the left, there's a sidebar with the following navigation:

- Amazon SNS
- Dashboard
- Topics
- Subscriptions** (highlighted with a red box)
- ▼ Mobile
 - Push notifications
 - Text messaging (SMS)
 - Origination numbers

The main content area is titled "Subscriptions (1)". It includes a toolbar with buttons for Edit, Delete, Request confirmation, Confirm subscription, and a prominent orange "Create subscription" button. A search bar is also present. Below the toolbar is a table with columns: ID, Endpoint, Status, Protocol, and Topic. The table currently has no data.

Configuración de nuestra App - SNS

Ejemplo
en vivo



- Seleccionamos el tópico al cual suscribirnos, el medio de suscripción (email en este caso) e ingresamos el destinatario (nuestra dirección de mail).

Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN

 arn:aws:sns:us-east-1:786671515187:notificaciones

Protocol

The type of endpoint to subscribe

Email Email

Endpoint

An email address that can receive notifications from Amazon SNS.

Cancel Create subscription

CODER HOUSE

Configuración de nuestra App - SNS

Ejemplo
en vivo



- La suscripción está creada pero para que esté activa debemos confirmarla mediante un link enviado a la dirección de la suscripción:

Subscription to notificaciones created successfully.
The ARN of the subscription is arn:aws:sns:us-east-1:786671515187:notificaciones:b28391c2-5eef-4fe7-ab9c-7d513737f5b7.

Amazon SNS > Topics > notificaciones > Subscription: b28391c2-5eef-4fe7-ab9c-7d513737f5b7

Subscription: b28391c2-5eef-4fe7-ab9c-7d513737f5b7

Details	
ARN	arn:aws:sns:us-east-1:786671515187:notificaciones:b28391c2-5eef-4fe7-ab9c-7d513737f5b7
Endpoint	[REDACTED]@gmail.com
Topic	notificaciones
Status	Pending confirmation
Protocol	EMAIL

AWS Notification - Subscription Confirmation Inbox ×



AWS Notifications <no-reply@sns.amazonaws.com>
to me ▾

You have chosen to subscribe to the topic:

arn:aws:sns:us-east-1:786671515187:notificaciones

To confirm this subscription, click or visit the link below (If this was in error no action is necessary)

[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation emails, click here.



Confirmar suscripción

- De esta forma, con esta página que se nos abre queda confirmada nuestra suscripción.

 Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-east-1:786671515187:notificaciones:b28391c2-5eef-4fe7-ab9c-7d513737f5b7

If it was not your intention to subscribe, [click here to unsubscribe](#).



BREAK

¡5/10 MINUTOS Y VOLVEMOS!

CREAR NUESTRO PROYECTO



Configurar nuestro proyecto de prueba

Trabajaremos ahora sobre una aplicación de pruebas. Nuestro proyecto será un CRUD bien sencillo. Comenzamos con el package.json en donde incluiremos express y el SDK de AWS:

```
{  
  "name": "ejemplo-aws",  
  "author": "coderhouse",  
  "version": "1.0.0",  
  "description": "",  
  "main": "app.js",  
  "scripts": {  
    "start": "node ."  
  },  
  "keywords": [],  
  "license": "ISC",  
  "dependencies": {  
    "aws-sdk": "^2.874.0",  
    "express": "^4.17.1"  
  }  
}
```



Configurar nuestro proyecto de prueba

Luego, incluimos requerimos, instanciamos y configuramos nuestras dependencias:

```
const express = require('express');
const AWS = require('aws-sdk');

AWS.config.update({
  region: 'us-east-1'
});

const sns = new AWS.SNS()
const SNS_TOPIC_ARN = 'arn:aws:sns:us-east-1:786671515187:notificaciones'

const dynamodb = new AWS.DynamoDB.DocumentClient();
const TABLE_NAME = 'product-inventory';

const app = express();
app.use(express.json());

app.get('/', (req, res) => { res.send('test api') })
```



Configurar nuestro proyecto de prueba

Agregamos los controladores a las rutas de nuestros endpoints CRUD:

```
app.get('/api/productos', async (req, res) => {
  const params = {
    TableName: TABLE_NAME
  }
  try {
    const productos = await scanDynamoRecords(params);
    res.json(productos);
  } catch (error) {
    console.error('Ocurrió un error: ', error);
    res.sendStatus(500);
  }
})
```

```
async function scanDynamoRecords(scanParams) {
  try {
    let dynamoData = await dynamodb.scan(scanParams).promise();
    const items = dynamoData.Items
    while (dynamoData.LastEvaluatedKey) {
      scanParams.ExclusiveStartKey = dynamoData.LastEvaluatedKey;
      dynamoData = await dynamodb.scan(scanParams).promise();
      items.push(...dynamoData.Items);
    }
    return items;
  } catch (error) {
    throw new Error(error);
  }
}
```



Configurar nuestro proyecto de prueba

```
app.post('/api/productos', (req, res) => {
  const params = {
    TableName: TABLE_NAME,
    Item: req.body
  }
  dynamodb.put(params).promise()
    .then(() => {
      console.log('se guardó')
      const prod = JSON.stringify(req.body)
      return sns.publish({
        Message: `nuevo producto agregado! ${prod}`,
        Subject: 'nuevo producto',
        TopicArn: SNS_TOPIC_ARN
      }).promise()
    })
})
```

```
.then(data => {
  console.log('se notificó')
  console.log(data)

  const body = {
    Operation: 'SAVE',
    Message: 'SUCCESS',
    Item: req.body
  }
  res.json(body);
})

.catch(error => {
  console.error('Ocurrió un error: ', error);
  res.status(500).end();
})
```



Configurar nuestro proyecto de prueba

```
app.put('/api/productos/:id', (req, res) => {
  const item = {
    ...req.body,
    productId: req.params.id
  }
  const params = {
    TableName: TABLE_NAME,
    Item: item
  }
})
```

```
dynamodb.put(params).promise()
  .then(() => {
    const body = {
      Operation: 'UPDATE',
      Message: 'SUCCESS',
      Item: item
    }
    res.json(body);
  })
  .catch(error => {
    console.error('Ocurrió un error: ', error);
    res.sendStatus(500);
  })
})
```



Configurar nuestro proyecto de prueba

```
app.delete('/api/productos/:id', (req, res) => {
  const params = {
    TableName: TABLE_NAME,
    Key: {
      'productId': req.params.id
    },
    ReturnValues: 'ALL_OLD'
  }
})
```

```
dynamodb.delete(params).promise()
  .then(response => {
    const body = {
      Operation: 'DELETE',
      Message: 'SUCCESS',
      Item: response
    }
    res.json(body);
  })
  .catch(error => {
    console.error('Ocurrió un error: ', error);
    res.sendStatus(500);
  })
})
```



Configurar nuestro proyecto de prueba

Finalmente, agregamos el código para conectar nuestro servidor al puerto 8080 o al que corresponda según las variables de entorno:

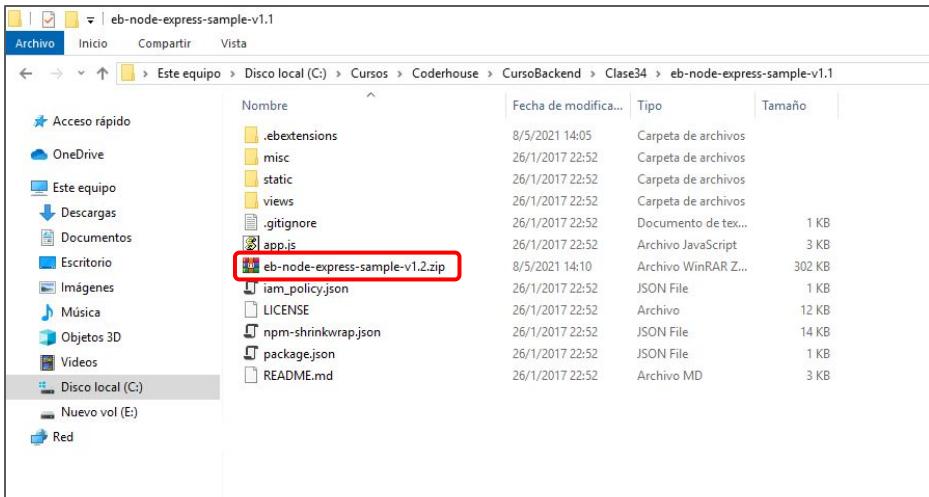
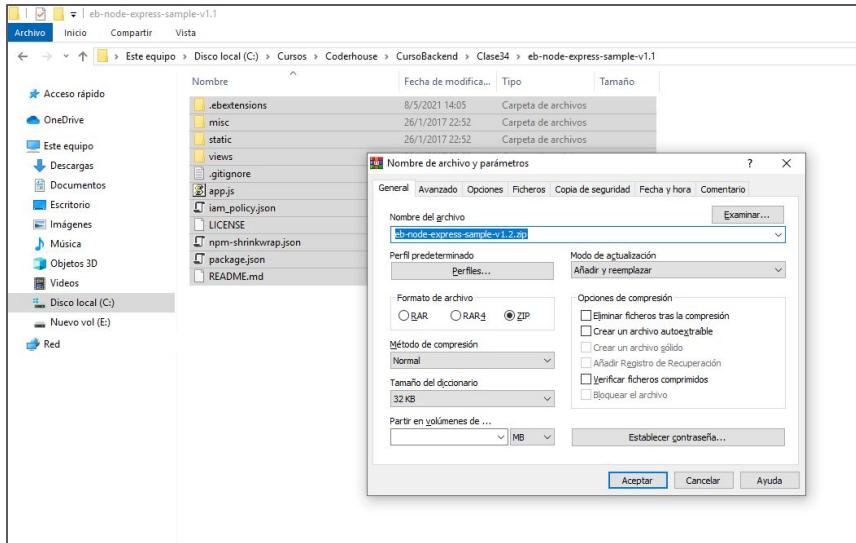
```
const port = process.env.PORT || 8080;
const server = app.listen(port, () => {
  console.log(`Escuchando en puerto ${server.address().port}`);
})
server.on('error', error => { console.log(error) })
```

CARGAR NUESTRO PROYECTO EN LA PLATAFORMA DE ELASTIC BEANSTALK



Configurar nuestro proyecto de prueba

- Creamos un zip con todos los archivos de nuestro proyecto.





Implementar nuestro proyecto

- Volvemos a la página de nuestra aplicación en AWS y vamos al botón de carga e implementar. Clickeamos en elegir archivo para cargar nuestro proyecto.

The screenshot shows the AWS Elastic Beanstalk console for the environment 'Serverch-env'. The main panel displays the application 'Serverch-env' with the URL 'Serverch-env.eba-bywb6ppq.us-east-2.elasticbeanstalk.com'. The 'Estado' section shows a green checkmark icon. The 'Versión en ejecución' section shows 'Sample Application' and a 'Cargar e implementar' button, which is highlighted with a red box. The 'Plataforma' section shows 'node' and 'Node.js 14 running on 64bit Amazon Linux 2/5.3.2'. Below this, the 'Eventos recientes' section is shown with a 'Mostrar todo' button. On the left sidebar, there are sections for 'Entornos', 'Aplicaciones', 'Historial de cambios', 'serverch' (with 'Versiones de la aplicación' and 'Configuraciones guardadas'), and 'Serverch-env' (with 'Ir al entorno', 'Configuración', 'Registros', 'Estado', 'Monitorización', 'Alarms', and 'Actualizaciones'). At the bottom, there are links for 'Comentarios', 'Español', and legal notices: '© 2008 - 2021, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.', 'Política de privacidad', 'Términos de uso', and 'Preferencias de cookies'.



Implementar nuestro proyecto

- Volvemos a la página de nuestra aplicación en AWS y vamos al botón de carga e implementar. Clickeamos en elegir archivo para cargar nuestro proyecto.

The screenshot shows the AWS Elastic Beanstalk console with the 'Cargar e implementar' (Upload and Deploy) dialog box open. The dialog box contains the following elements:

- A message: "Para implementar una versión anterior, vaya a la página de versiones de la aplicación."
- A section titled "Cargar aplicación" with a button labeled "Elegir archivo" (Select file), which is highlighted with a red box.
- A "Etiqueta de versión" (Version label) input field.
- A "Preferencias de implementación" (Deployment preferences) section stating: "La versión de la aplicación se implementará con la política Todo a la vez." (The application version will be deployed with the All at once policy.)
- A note: "Número actual de instancias: 1" (Current number of instances: 1).
- Buttons: "Cancelar" (Cancel) and "Implementar" (Deploy) at the bottom.

The background shows the main Elastic Beanstalk environment configuration page with various tabs like 'Entornos', 'Aplicaciones', and 'Historial de cambios' visible.

Implementar nuestro proyecto



- Elegimos nuestro zip de proyecto y clickeamos en el botón de implementar.

The screenshot shows the AWS Elastic Beanstalk console with a modal dialog titled "Cargar e implementar". In the background, the application "Serverch-envi" is listed under "Aplicaciones". The "Cargar aplicación" section has the file "eb-node-express-sample-v1.1.zip" selected. The "Etiqueta de versión" field contains "Sample Application-1". The "Preferencias de implementación" section indicates "La versión de la aplicación se implementará con la política Todo a la vez." and "Número actual de instancias: 1". A red box highlights the "Implementar" button at the bottom right of the modal. In the bottom left corner, there is a smaller modal dialog with a red box around the "Abrir" button, which is used to select the deployment file.

Nombre de archivo: eb-node-express-sample-v1.1.zip

Todos los archivos (*.*)

Abrir Cancelar

Implementar nuestro proyecto



- Una vez que se actualiza el entorno de Elastic Beanstalk vemos la notificación que se implementó una nueva versión en la instancia.

The screenshot shows the AWS Elastic Beanstalk console for the 'Serverch-env' environment. A prominent message at the top states: 'Elastic Beanstalk está actualizando su entorno. Para cancelar esta operación, seleccione Anular la operación actual en el menú desplegable Acciones. Ver los eventos'. Below this, the 'Estado' section shows a circular arrow icon, indicating the process is ongoing. The 'Versión en ejecución' section shows 'Sample Application'. The 'Plataforma' section indicates 'Node.js 14 running on 64bit Amazon Linux 2/5.3.2'. On the left sidebar, under 'Serverch-env', there are sections for 'Configuración', 'Registros', 'Estado', 'Monitorización', 'Alarms', and 'Actualizaciones'.

The screenshot shows the AWS Elastic Beanstalk console for the 'Serverch-env' environment. It displays deployment logs with the message 'Deploying new version to instance(s)'. Below this, a table lists recent events with details such as 'Environment health has transitioned from Ok to Info. Application update in progress (running for 45 seconds)' and 'Environment update is starting'. The 'Eventos recientes' table includes columns for 'Hora', 'Tipo', and 'Detalles'. The left sidebar is identical to the first screenshot, showing 'Configuración', 'Registros', 'Estado', 'Monitorización', 'Alarms', and 'Actualizaciones'.



Implementar nuestro proyecto

- Y luego, chequeamos la notificación de que se actualizó correctamente el entorno.

The screenshot shows the AWS Elastic Beanstalk console for the environment 'Serverch-env'. The main interface displays the application status: 'Estado' (green checkmark), 'Versión en ejecución' (Sample Application), and 'Plataforma' (Node.js 14 running on 64bit Amazon Linux 2/5.3.2). Below the status, the 'Eventos recientes' (Recent Events) section shows two log entries:

Hora	Tipo	Detalles
08-05-2021 14:22:15 UTC-0300	INFO	Environment update completed successfully.
08-05-2021 14:22:15 UTC-0300	INFO	New application version was deployed to running EC2 instances.

Verificación



- Ya podemos utilizar el link del entorno para probar nuestra api REST usando nuestro cliente HTTP preferido (por ejemplo, postman)

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with navigation links like 'Entornos', 'Aplicaciones', 'Historial de cambios', and sections for 'serverch' and 'Serverch-env'. The 'Serverch-env' section is expanded, showing options like 'Ir al entorno', 'Configuración', 'Registros', 'Estado', 'Monitorización', 'Alarms', and 'Actualizaciones'. The main panel displays the environment details: 'Estado' (green checkmark), 'Versión en ejecución' (Sample Application), 'Plataforma' (node.js), and 'Eventos recientes'. A red box highlights the URL 'Serverch-env.eba-bywb6ppq.us-east-2.elasticbeanstalk.com' in the browser header.

ELIMINAR NUESTRA APP DE LA PLATAFORMA ELASTIC BEANSTALK

Remover el entorno



- Vamos a acciones y elegimos terminar el entorno. En el cartel que nos aparece, ponemos el nombre del entorno y clickeamos en terminar.

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, there's a sidebar with navigation links like 'Entornos', 'Aplicaciones', 'Historial de cambios', and 'serveraws-ch'. Under 'serveraws-ch', 'Serverawsch-env' is selected, with sub-links 'Ir al entorno', 'Configuración', 'Registros', 'Estado', 'Monitorización', 'Alarms', 'Actualizaciones administradas', 'Eventos', and 'Etiquetas'. The main content area shows the 'Serverawsch-env' environment details. It includes sections for 'Estado' (green checkmark), 'Versión en ejecución' (Sample Application-2), and 'Eventos recientes' (listing recent logs). A dropdown menu under 'Acciones' has 'Terminar el entorno' highlighted with a red box. A second window titled 'Confirmar terminación del entorno' is open, asking if you want to permanently terminate the environment. It lists the environment's details: Capa: Servidor web, Plataforma: Node.js 14 running on 64bit Amazon Linux 2/5.3.2, Version: Sample Application-2, and Ultima modificación: 09-05-2021 09:09:29 UTC-0500. Below it, it says 'Si continua con esta acción, ocurrirá lo siguiente:' followed by two bullet points: 'Se lanzará Serverawsch-env.elb.amazonaws.com' and 'Todos los recursos adicionales asociados a su entorno de Elastic Beanstalk se destruirán.' A red arrow points to the 'Introduzca el nombre del entorno para confirmar:' input field, which contains 'Serverawsch-env'. A red box highlights the 'Terminar' button at the bottom right of the confirmation window. The bottom of the screen shows the Coder House logo.

Eliminar la App creada



- Chequeamos que se removió correctamente la instancia del entorno.
- Luego vamos a acciones y elegimos eliminar aplicación.
- Recordamos que vamos a eliminar la aplicación por el tema del cobro de la cuenta de AWS.

Buscar servicios, características, productos del Marketplace y documentos [Alt+S]

backendNodeAWS ▾ Ohio ▾ Soporte ▾

Elastic Beanstalk ▾

Servicios ▾

Serverawsch-env

Serverawsch-env.joycewe.us-east-2.elasticbeanstalk.com (c-htRtkmwzU)

Nombre de la aplicación: serveraws-ch

Estado

Versión en ejecución: Sample Application-2

Cargar e implementar

Plataforma: node

Desconocido

Cambiar

Eventos recientes

Hora	Tipo	Detalles
09-05-2021 09:17:25 UTC-0300	INFO	Removed instance [-0b5d4ee8f2b880e3b] from your environment.
09-05-2021 09:16:25 UTC-0300	INFO	Environment health has transitioned from Ok to Info. Terminate in progress (running for 11 seconds).
09-05-2021 09:16:15	INFO	Waiting for EC2 instances to terminate... This may take a few minutes.

Comentarios Español ▾

© 2008-2021, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Política de privacidad Términos de uso

Buscar servicios, características, productos del Marketplace y documentos [Alt+S]

backendNodeAWS ▾ Ohio ▾ Soporte ▾

Elastic Beanstalk ▾

Servicios ▾

Elastic Beanstalk > Aplicaciones

Todas las aplicaciones

Filtrar resultados que coincidan con los valores de visualización

Nombre de la aplicación	Entornos	Fecha de creación	Última modificación	ARN
serveraws-ch	Serverawsch-env	09-05-2021 08:33:11 UTC-0300	09-05-2021 08:33:11 UTC-0300	arn:aws:elasticbeanstalk:us-east-2:791724667592:application/serveraws-ch

Acciones ▾

Crear un entorno

Eliminar aplicación

Ver las versiones de la aplicación

Ver las configuraciones guardadas

Restaurar entorno terminado

Comentarios Español ▾

© 2008-2021, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados. Política de privacidad Términos de uso Preferencias de cookies

Eliminar la App creada



- Ponemos el nombre de la aplicación y luego eliminar.
- Luego, debemos escribir la palabra delete y clickear en el botón eliminar.

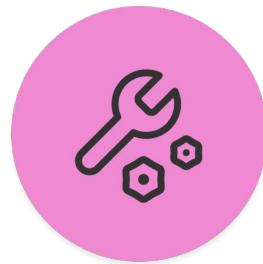
The image shows two screenshots from the AWS console. The left screenshot displays the 'Confirmar eliminación de la aplicación' (Delete application confirmation) dialog for the application 'serveraws-ch'. It asks if you want to permanently delete the application and lists the environments it contains. A red arrow points to the input field where 'serveraws-ch' is typed, and another red arrow points to the 'Eliminar' (Delete) button at the bottom right. The right screenshot shows the 'Eliminar tabla' (Delete table) dialog for the 'registro-usuarios' table in DynamoDB. It asks if you're sure you want to delete the table and provides checkboxes for deleting CloudWatch alarms and creating a backup. A red arrow points to the text input field where 'delete' is typed, and another red arrow points to the 'Eliminar' (Delete) button at the bottom right.

Eliminar la App creada



- Finalmente nos redirige de nuevo a la página inicial de AWS.

The screenshot shows the AWS Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a search bar, and dropdown menus for 'Servicios', 'backendNodeAWS', 'Ohio', and 'Soporte'. The main title 'AWS Elastic Beanstalk' is prominently displayed, followed by the subtitle 'Administración integral de aplicaciones web.' Below this, a description explains that Elastic Beanstalk is a service for deploying and scaling web applications developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker. A large orange 'Create Application' button is visible. To the left, a sidebar lists 'Entornos', 'Aplicaciones', and 'Historial de cambios'. On the right, sections include 'Empezar' (with a sub-note about easily deploying web applications), 'Precios' (noting no additional charges), 'Cómo funciona' (explaining automatic deployment, scaling, and monitoring), 'Beneficios y características' (mentioning ease of use and control over resources), and 'Introducción' (linking to a detailed introduction). At the bottom, there are links for 'Comentarios', 'Español', and various legal notices.



IMPLEMENTAR PROYECTO EN AWS

Tiempo: 10 minutos



Implementar proyecto en AWS

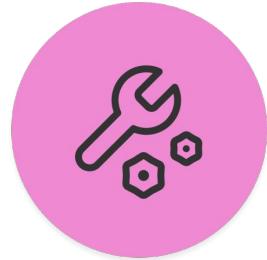
Tiempo: 10 minutos

Crear una cuenta en AWS para trabajar con la plataforma Elastic Beanstalk.

Vamos a implementar la primera parte del proyecto de referencia en AWS:

<https://aws.amazon.com/es/getting-started/hands-on/deploy-nodejs-web-app/>

- Crear un proyecto sobre la plataforma Elastic Beanstalk que contenga un proxy reverso nginx con balanceador de carga y de 2 a 4 instancias de servidor en autoscaling.
- Implementar el servidor con el código de ejemplo.
- Cuando el servidor se ponga online, verificar la página devuelta en su ruta raíz.



CONTINUAR CON LA IMPLEMENTACIÓN

Tiempo: 10 minutos



Continuar con la implementación

Tiempo: 10 minutos

Continuando con lo realizado en el desafío anterior, y utilizando el código de ejemplo suministrado en

<https://aws.amazon.com/es/getting-started/hands-on/deploy-nodejs-web-app/>

- Dar permisos a la aplicación para tener acceso a los servicios de DynamoDB y SNS en la plataforma AWS.
- Crear una tabla en DynamoDB que contenga una clave email tipo string, registrando su nombre dentro del archivo de configuración ubicado en el código de ejemplo.
- Configurar el mail por el cual vamos a recibir notificaciones en el archivo de configuración correspondiente en el código de ejemplo.



Continuar con la implementación

Tiempo: 10 minutos

- Setear el proxy para las static files en la configuración del código de ejemplo.
- Zipear el código de ejemplo y subirlo a la plataforma Elastic Beanstalk.
- Verificar que nos llegue el mail de suscripción. Registrarse al servicio de notificaciones por SNS, cliqueando el link que viene en este email.
- Una vez que la aplicación se ponga online, mediante la página que proveé en su ruta raíz, registrar dos usuarios, comprobando que se guarden en la tabla creada en DynamoDB y lleguen los mails correspondientes avisando por cada nueva registración realizada.

Continuar con la implementación

Desafío
generico



Tiempo: 10 minutos

- Realizar un cambio en el código, puede ser en un archivo de vista, zipeando esta nueva versión y subiéndola a la plataforma. Comprobar que este nuevo cambio esté disponible online y todo funcione correctamente.
- Luego de probar el desempeño del servidor, realizar la finalización del entorno y de la aplicación, sin olvidar borrar la tabla creada en DynamoDB.



DESPLEGAR NUESTRO PROYECTO EN LA NUBE

Retomemos nuestro trabajo para implementar nuestro proyecto en Heroku

IMPLEMENTAR NUESTRO PROYECTO EN HEROKU

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Crear un proyecto en Heroku.com para subir el servidor que venimos realizando, reformando todo lo necesario para su correcto funcionamiento en la nube.

Subir el código a Heroku.com, sin olvidar incluir el archivo .gitignore para evitar subir los node_modules. Comprobar que el proyecto inicie de manera correcta en la nube. Verificar que en su ruta raíz se encuentre la página pública del servidor.

El servidor debe seguir funcionando en forma local.

Realizar un cambio a elección en alguna vista, probar en forma local y subir nuevamente el proyecto a Heroku, verificando que la nueva reforma esté disponible online.

Revisar a través de una consola local, los mensajes enviados por nuestro servidor en Heroku a su propia consola.

*¿*PREGUNTAS?



¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- AWS y su plataforma Elastic Beanstalk.
- Creación de una cuenta y una App en esta plataforma.
- Implementación de nuestro proyecto en la misma.



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDOLAEDUCACIÓN