Curso de Introducción a AWS: Fundamentos de Cloud Computing

Enrique Alexis Lopez Araujo

Platzi

Entendamos primero como funciona la web en términos simples. Tenemos un **cliente** con una dirección IP que se conecta a una **red** para hacer una **petición** a un **servidor** con otra dirección IP. Este servidor devuelve una respuesta al cliente.

Si la web fuera un servicio postal, el **cliente** seríamos nosotros con la **petición** o paquete que queremos enviar, la **red** sería el servicio postal en sí y el **servidor** representaría al destinatario al que queremos enviar el paquete.

Diagrama

Descripción generada automáticamente

**¿Cómo está compuesto un servidor?**

Un servidor posee los siguientes componentes:

* **Cómputo/CPU**: Realiza las operaciones que necesitamos.
* **Memoria RAM**: Contiene la información a procesar mediante la CPU. Es como un cerebro
* **Almacenamiento**: Archiva datos, a modo de archivos de texto plano.
* **Bases de datos**: Información almacenada de manera estructurada
* **Redes**: Cables, routers y servidores conectados unos a otros. Servidores DNS

**Terminología de IT (redes)**

En términos generales, un cliente envía un paquete a un **router**, el cual reenvía este paquete al **switch**, y este se encarga de distribuirlo.

* **Router**: dispositivo que reenvía paquetes de datos entre redes informáticas
* **Switch**: dispositivo que toma paquetes y los envía al servidor/cliente correcto en la red

Diagrama

Descripción generada automáticamente

**Diseño tradicional de infraestructura**

Las grandes empresas de IT empezaron comprando servidores y montándolos en sus garajes. Se encontraron con problemas al tratar de expandir su infraestructura, como los costos de mover estos servidores, comprar nuevos, y más…

**Problemas del enfoque de IT tradicional**

A continuación, conocerás algunas dificultades del enfoque de tecnología de la información habitual:

* **Renta**: los costos de rentar espacios para mantener servidores son altos
* **Mantenimiento**: el funcionamiento de los servidores es difícil de asegurar
* **Remplazar y agregar hardware**: pueden existir largos tiempos de espera para conseguir el hardware necesario
* **Escalamiento limitado**: podemos vernos limitados por el espacio donde almacenamos los servidores
* **Monitoreo 24/7**: debemos contratar gente para monitorear los servidores
* **Desastres naturales**: ¿cómo evitamos que se caigan nuestros servicios si ocurre un imprevisto?

**Qué es la computacion en la nube**

3/15

La computación en la nube es la entrega bajo demanda de **recursos de IT** como computación, almacenamiento y otros servicios a través de internet. En pocas palabras, es como si alquiláramos la computadora de otra persona.

Esta tecnología permite acceso instantáneo a los recursos que necesites, así como la adquisición del tipo y tamaño exacto de estos recursos. Algunos servicios que seguramente has usado son Gmail (proveedor de email), Dropbox (proveedor de almacenamiento) y Netflix (proveedor de video bajo demanda).

**Modelos de computación en la nube**

A continuación, conocerás las distintas plataformas en la nube que utilizamos cuando trabajamos en proyectos personales o en nuestra empresa.

**Nube pública**

La nube pública se refiere a los recursos de proveedores que utilizamos a través de internet y algunos ejemplos son Google Cloud Platform (GCP), Azure y AWS.

Además, posee estas ventajas:

* Elimina los gastos de capital comercial (CapEx) y reduce el gasto operativo (OpEx).
* Reduce los precios en economías de escala
* Despliega aplicaciones a nivel global en cuestión de minutos

**Nube privada**

La nube privada es un servicio empleado por una organización que no está abierto al público. Permite un control total de la infraestructura y es útil para aplicaciones con requerimientos específicos de seguridad o comerciales.

**Nube híbrida**

La nube híbrida consiste en mantener nuestra infraestructura y extender sus capacidades mediante la nube pública. Posibilita el control sobre activos sensibles en tu infraestructura privada, aprovechando la flexibilidad y rentabilidad de la nube pública.

**Características de la computación en la nube**

Ahora que conoces los distintos modelos de tecnología en la nube, es importante que hablar sobre sus propiedades de computación.

* Este modelo genera un autoservicio en demanda (con registros en la plataforma ya se pueden proveer recursos)
* Tiene un amplio acceso a la red
* Proporciona un espacio donde los clientes pueden compartir infraestructura y recursos de manera segura

**Problemas resueltos por la nube**

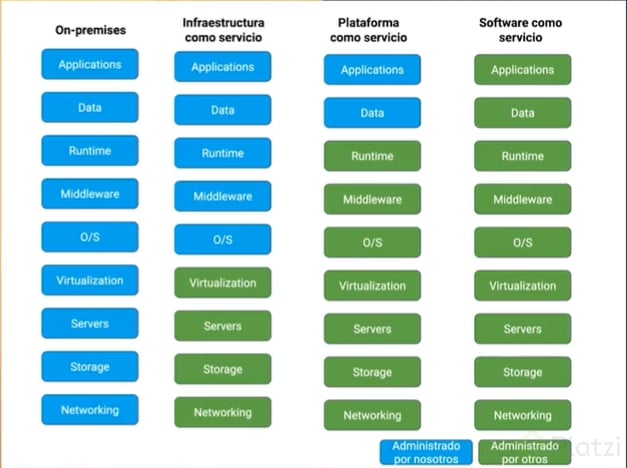
Por último, es crucial que conozcas las cualidades que trae implementar un sistema de computación en la nube.

* La nube aporta flexibilidad (puedes cambiar los tipos de recursos cuando sea necesario)
* Brinda rentabilidad y un servicio medido pues pagas solo por lo que usas
* Trae escalabilidad al agregar capacidad para hardware o equipos que necesitan acomodar cargas grandes
* Ofrece elasticidad al dar capacidad de escalar automáticamente cuando sea necesario
* Tiene alta disponibilidad y tolerancia a fallos
* Proporciona agilidad (puedes desarrollar, probar y ejecutar rápidamente aplicaciones en la nube)



**Los diferentes tipos de cómputo: IaaS vs. PaaS vs. SaaS**

4/15



Ahora que conoces más sobre la tecnología en la nube, es importante introducir sus **distintos tipos de servicio** en la industria para identificar sus diferencias.

Estos modelos varían de acuerdo al tipo de servicio informático que pueda ofrecer, como servidores, almacenamiento, software o bases de datos.

**Infrastructure as a Service (IAAS)**

La infraestructura como servicio (IAAS) proporciona componentes básicos de IT en la nube, es decir, **redes, computación, almacenamiento, etc.** A su vez, provee el máximo nivel de flexibilidad para adaptarlo a tus necesidades.

**Ejemplos:**

* Azure Virtual Machines
* Linode
* Digital ocean
* EC2 AWS

**Platform as a Service (PAAS)**

Los modelos que ofrecen una plataforma como servicio (PAAS) eliminan la necesidad de que administremos la infraestructura y proveen una plataforma para gestionar aplicaciones.

**Ejemplos:**

* Heroku
* Google App Engine
* AWS Elastic Beanstalk

**Software as a Service (SAAS)**

El Software como servicio (SAAS) brinda un producto de software terminado que es ejecutado y administrado por el proveedor del servicio.

**Ejemplos:**

* Amazon Rekognition
* Dropbox
* Zoom
* Gmail

**On -premises**

On-premises se refiere a una forma tradicional de cómputo en la cual nos encargamos de gestionar nuestra propia infraestructura.

**Responsabilidades según el tipo de cómputo**

En la siguiente tabla se muestra qué componentes de IT están administrados según el tipo de cómputo en la nube. “Sí” indica que el componente está administrado por el proveedor de nube, “No” indica que nosotros somos responsables del componente.

| **Componente** | **On-premises** | **IAAS** | **PAAS** | **SAAS** |
| --- | --- | --- | --- | --- |
| Aplicaciones | No | No | No | Sí |
| Data | No | No | No | Sí |
| Runtime | No | No | Sí | Sí |
| Middleware | No | No | Sí | Sí |
| O/S | No | No | Sí | Sí |
| Virtualización | No | Sí | Sí | Sí |
| Servidores | No | Sí | Sí | Sí |
| Almacenamiento | No | Sí | Sí | Sí |
| Redes | No | Sí | Sí | Sí |

**Una zona de disponibilidad es un data center**

* Un **data center** está lleno de **servidores**
* Una **zona de disponibilidad** puede ser de varios data center.
* Cada zona de disponibilidad es uno o mas centro de datos discretos con energía, redes y conectividad redundante, están separados entre sí y están conectados entre sí con un gran ancho de banda, redes de latencia ultrabaja

**Infraestructura**

* AWS Regiones
* AWS Zonas de disponibilidad
* AWS Centro de datos
* Ubicaciones de AWS Edge / puntos de presencia

[Infraestructura global](https://aws.amazon.com/es/about-aws/global-infrastructure/?p=ngi&loc=0)

**Servicios Globales**

* IAM
* Route 53
* Cloudfront
* WAF

**Servicios Regionales**

* EC2
* Beanstalk
* Lambda
* Rekognition

[Servicios regionales de AWS](https://aws.amazon.com/es/about-aws/global-infrastructure/regional-product-services/)

**Diagrama del modelo de responsabilidad compartida**

* **AWS**:
  + hardware y la infraestructura global
    - Regiones (regions)
    - Zonas de disponibilidad (availability zones)
    - Ubicaciones de AWS Edge / puntos de presencia (Edge locations)
  + software:
    - computo (compute)
    - almacenamiento (storage)
    - bases de datos (database)
    - redes (networking)
* **Cliente**:
  + Actualizaciones de S.O.
  + Proteger los datos que se almacenan
  + Aplicaciones
  + Accesos
  + Administración de usuarios y grupos



Diagrama, Escala de tiempo

Descripción generada automáticamente

**Seguridad e identidad:**

**Protección a Datos**

* Amazon Macie: para descubrir y proteger sus datos sensibles
* AWS Key Management Service: almacena y administra claves de cifrado
* AWS CloudHSM: almacenamiento de claves basado en hardware y el cumplimiento normativo
* AWS Certificate Manager, provisiona, administra e implementa certificados de seguridad TSL y TLS
* AWS Secrets Manager: rotar, gestionar y recuperar secretos como contraseñas

**Protección de la infraestructura**

* AWS Shield, para la protección de denegación de servicio
* AWS Web Aplication Firewall, (WAF) filtra el tráfico de sitios web maliciosos
* AWS Firewall Manager, administra las reglas del firewall de forma centralizada

**Detección de amenazas**

* Amazon GuarDuty, detecta amenazas automáticamente
* Amazon Inspector, ayuda a analizar la seguridad de la aplicación
* Amazon config, registra y evalúa configuraciones de nuestros recursos
* Amazon CloudTrail, rastrea la actividad del usuario y el uso de las API

**Gestión de identidades**

* AWS Identity and Access Management, (IAM) administra de forma segura el acceso a una cuenta, servicios y recursos
* AWS Inicio de sesión único: Implemente el acceso de sesión único (single sign on)
* AWS administra la identidad dentro de las aplicaciones, se puede hacer el inicio de sesiones moviles
* AWS Servicio de Directorio, implementa y administra un Active Directory Service
* AWS Organizaciones, para gobernar y administrar de forma centralizada en un mismo lugar

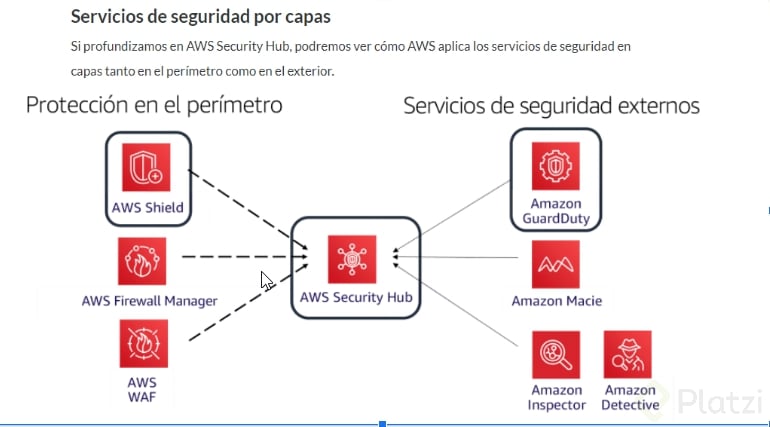




Imagen que contiene Gráfico

Descripción generada automáticamente

**Identity and Access Management (IAM)** es un servicio gratuito que nos ayuda a administrar los accesos a los servicios y recursos de tu cuenta en AWS. A su vez, puedes crear usuarios, grupos y establecer permisos de acceso a los recursos mediante el uso de políticas.

**Usuarios y grupos de usuarios de IAM**

Los usuarios y grupos de usuarios son de los principales componentes de IAM. Al crear tu cuenta de AWS te proporcionan un **usuario Root** que tiene acceso a todos los recursos,

Este usuario puede generar otros perfiles y cada uno con un acceso único a distintos recursos de AWS. Además, Root también puede configurar grupos de usuarios, donde cada miembro tiene y puede compartir permisos de acceso.

**Ejemplos de políticas de IAM**

El acceso a recursos se otorga mediante políticas. Este es un ejemplo de una política que otorga acceso de administrador.

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "\*",

"Resource": "\*"

}

]

}

También está este ejemplo de políticas de acceso a un bucket de S3 (almacenamiento)

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:ListBucket"

],

"Resource": "arn:aws:53 ::: bucket-name"

},

{

"Effect": "Allow",

"Action": [

"s3: GetObject",

"s3: PutObject",

],

"Resource": "arn:aws:53 ::: bucket-name /\*"

}

]

}

**IAM Roles**

Además de todas estas funciones, **IAM de AWS** permite asumir roles y otorgar permisos a otras tecnologías. Por ejemplo, podemos conceder a una máquina virtual el acceso a una base de datos mediante un rol de IAM.

**Secrets Manager**

Secrets Manager es un servicio de AWS que nos ayuda a proteger los datos secretos (contraseñas, claves y tokens) necesarios para acceder a nuestras aplicaciones, servicios y recursos.

También nos permite compartir automáticamente esta información cuando queramos. Además, este servicio evita que tengamos que copiar y pegar los secretos directamente en nuestro código.

****

****

**Servicio de directorio**

Un **directorio** es una base de datos que contiene información de inicio de sesión de todos los usuarios de una red y puede implementar políticas de seguridad.

Dado que Windows es el sistema operativo más usado a nivel mundial, Microsoft lanzó **Active Directory**. Este servicio permite que las empresas gestionen los inicios de sesión de sus empleados.

**AWS Directory Service**

Es una oferta de servicio administrado de AWS que posibilita que sus recursos utilicen **Active Directory** y ofrecen:

* Un directorio activo administrado sin tener que ejecutar servidores manualmente
* La opción de directorio activo simple
* El conector AD que brinda a usuarios inicio de sesión en aplicaciones de AWS con sus credenciales
* Un Servicio distribuido con error automático que funciona si hay fallas en los servidores
* El AWS Directory Service ss compatible con otros servicios de AWS

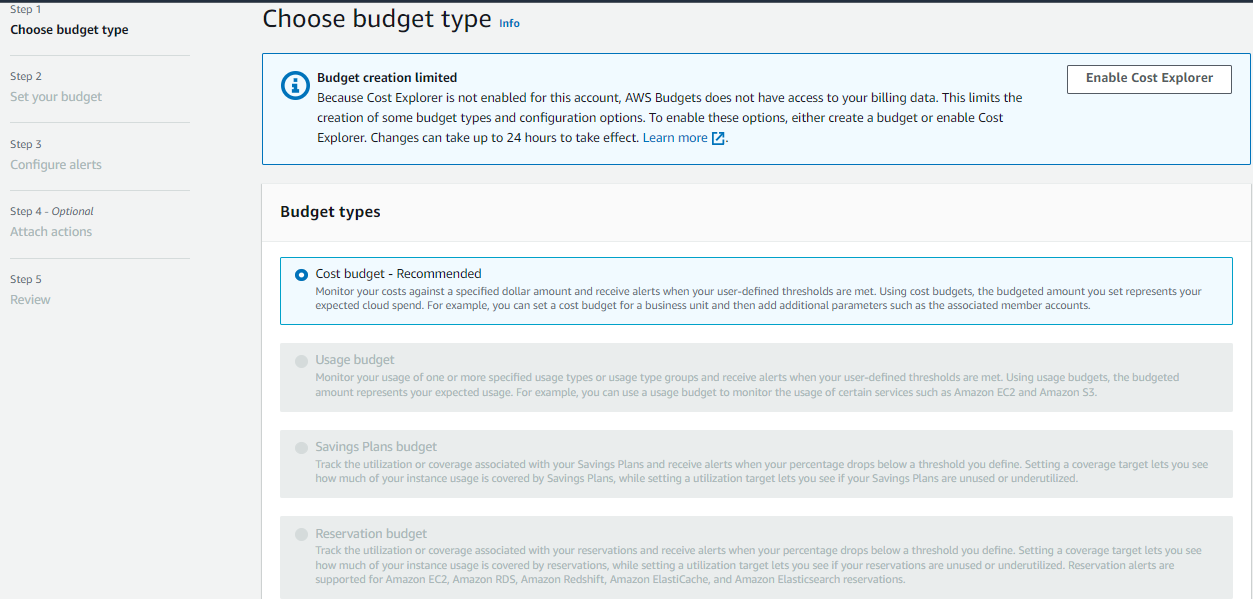
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Laboratorio: crea una alerta de facturación**

Para crear una alerta de facturación ve a tu nombre de usuario en la plataforma de AWS. Haz clic en la opcion **“Billing Dashboard”** > **“Budgets”** > **“Create a budget”**.

Aquí veremos los Budget Types. Marca la opción por defecto **“Cost budget - Recommended”** y haz clic en el botón superior: **“Enable cost Explorer”** para habilitar el seguimiento de gastos.

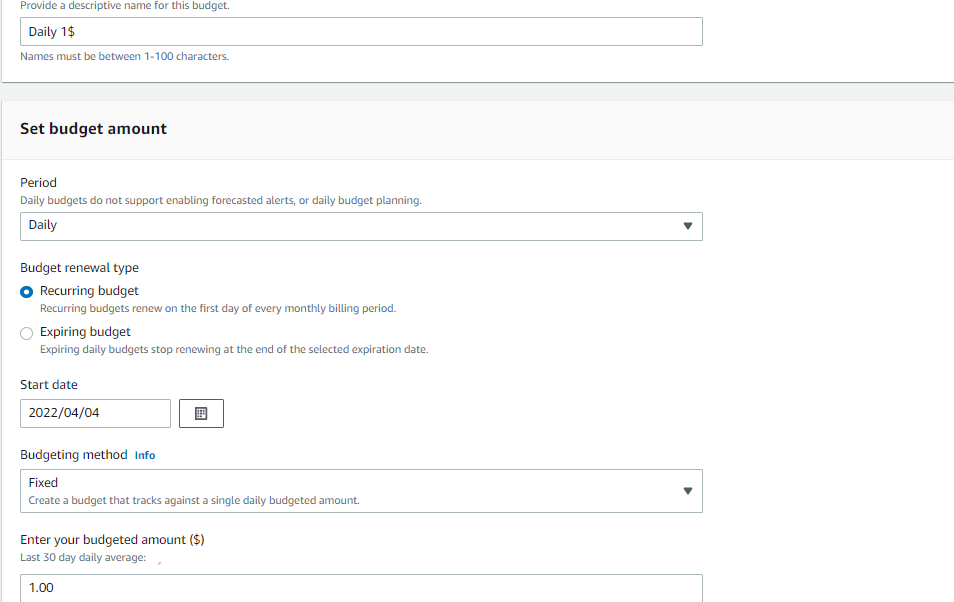


**Sigue los pasos para crear una alerta de facturación**

**Paso 1.**

De nuevo en la página de Budget Types, haz clic en **“Next”**. Verás la página **Set Your Budget.** Aquí oprime la opción **“budget”** y selecciona su frecuencia: **daily, monthly, etc.** Escoge desde cuando quieres empezar a hacer el monitoreo en **“Start Date”.**

En Budgeting method escoge **“Fixed”**. Esta opción implica que se va a monitorear el presupuesto cuando se gasta más de la cantidad indicada. Después completa la casilla **“Enter your budgeted amount”** según el tiempo.



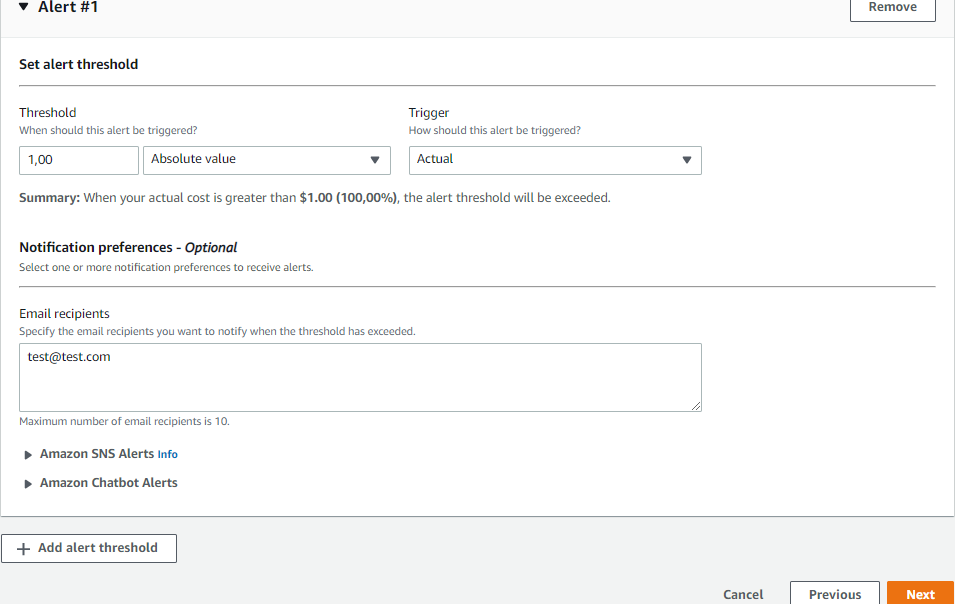
**Paso 2 (opcional)**

Adicionalmente, en la sección **“Budget Scope”** podemos delimitar nuestro presupuesto con algún servicio específico u otras opciones. De momento, seleccionamos **All AWS Services** y hacemos clic en **“Next”**.

**Paso 3**

Oprimimos el botón **“Add an alert threshold”**. En la sección: Alert #1, configuramos cómo se debe ejecutar la alerta, si es por el valor absoluto o algún porcentaje, en ese caso seleccionamos valor absoluto o **“Absolute Value”** y colocamos la cifra que queremos que encienda la alerta.

Luego en Email recipients indicamos el correo electrónico al que llegara la alerta. Finalmente, hacemos clic en **“Next”** y en **“Create budget”.**



**Laboratorio: crea una alerta de facturación**

Proceso de configuracion:

1. click en el nombre de usuario
2. click en opcion: **Billing Dashboard**
3. **Budgets**
4. **Create a Budgets**
5. Nos mostrara los **Budget Types**, dejar amarcada la opcion por defecto **Cost budget -Recommended** y click en el boton superior: **Enable cost Explorer** para habilitar el seguimiento de gastos
6. En la pagina de **budget types**, dar click en next
7. nos mostrara la pagina **Set Your Budget**
8. seleccionar cada cuando se realizara el budget, daily, monthly, etc
9. seleccionar desde cuando se quire empezar a hacer el monitoreo en **Start Date**
10. seleccionar en **Choose how budget**. Fixed significa que va monitorear si nos gastamos mas o igual de la cantidad indicada en el campo **Enter your budgeted amount($)**
11. click en **Next**
12. Click en el boton: **Add an alert threshold**
13. En la seccion **Alert #1,** configuraremos como se debe ejecutar la alerta
14. Indicamos si la alerta se hara por el valor absoluto o algun porcentaje
15. en **Email recipients** indicar el correo electronico al que llegara la alerta
16. click en: **Add alert threshold**
17. click en: **Create Budget**