Para la actividad de presentación sobre Computación en la Nube, cada grupo deberá elegir uno de los temas propuestos en la lista de consignas. A continuación, desarrollarán una presentación enfocada en el tema que seleccionen, profundizando en los aspectos específicos que se indican en la consigna.

O Listo

O En proceso

O Por hacer

**Ejemplos Prácticos de Aplicación en la Nube**

1. **Caso Práctico 1:** Aplicación Web: Describir cómo se implementaría una aplicación web utilizando la Computación en la Nube. Explicar el proceso desde el desarrollo hasta el despliegue en una plataforma como AWS o Azure.
2. **Caso Práctico 2:** Almacenamiento de Archivos: Mostrar cómo funciona el almacenamiento en la nube usando un servicio como Amazon S3 o Google Cloud Storage, incluyendo la seguridad de acceso y gestión de datos
3. **Diagramas de Arquitectura:** Incluir diagramas sencillos que muestren la arquitectura de cada caso y cómo los datos fluyen a través de los servicios de la nube.
4. **Conclusión y Preguntas**

* Resumen de la Presentación: Resumir los conceptos clave de cada sección, como los modelos de servicio, implementación y las plataformas Azure y AWS.
* Opinión del Grupo: Cada grupo debe dar su conclusión personal sobre el impacto y la relevancia futura de la Computación en la Nube.

**Requisitos para la Presentación**

* Duración: Cada presentación debe durar aproximadamente 15 minutos.
* Visuales: Usar diapositivas con gráficos, imágenes y diagramas claros y explicativos.
* Claridad y Explicación: Explicar cada concepto con lenguaje claro y fácil de entender para asegurar que todos comprendan los conceptos.

Caso Práctico 1: para elegir UNO-> AWS o Azure

**1. Introducción a Azure App Service**

**Azure App Service** es una plataforma completamente gestionada para crear, desplegar y escalar aplicaciones web. Soporta múltiples lenguajes y marcos de desarrollo, como .NET, PHP, Node.js, Python y Java, y ofrece características como autoescalado, alta disponibilidad y seguridad.

**Principales beneficios**:

* Despliegue y gestión fáciles.
* Soporta entornos Windows y Linux.
* Integración con otros servicios de Azure (por ejemplo, Azure SQL Database, Azure Storage).

### **2. Configuración de Azure App Service**

El artículo explica los pasos para configurar un **Azure App Service** y alojar tu aplicación web.

#### **Paso 1: Crear una cuenta de Azure**

Si no tienes una cuenta en **Azure**, el artículo sugiere registrarse en una cuenta gratuita, que ofrece $200 en créditos para los primeros 30 días y acceso a servicios gratuitos limitados después de eso.

#### **Paso 2: Crear un Resource Group**

1. Ve al **Portal de Azure**.
2. Crea un **Resource Group** para organizar tus recursos.
   * Este es un contenedor lógico para los recursos de Azure, como bases de datos, aplicaciones web, etc.

#### **Paso 3: Crear un App Service Plan**

El **App Service Plan** determina la ubicación, características y nivel de precios de tu aplicación. Puedes elegir entre:

* **Free o Shared**: Recursos limitados, ideal para desarrollo y pruebas.
* **Basic, Standard o Premium**: Proveen más recursos, recomendados para producción.

#### **Paso 4: Crear la aplicación web**

1. En el **Portal de Azure**, ve a **App Services** y crea una nueva aplicación web.
2. Selecciona el **Resource Group** y el **App Service Plan** que creaste previamente.
3. Asigna un nombre único a tu aplicación, que será parte de la URL de tu app (por ejemplo, miaplicacion.azurewebsites.net).
4. Elige el **stack de ejecución** (por ejemplo, Node.js, .NET, PHP).

### **3. Desplegar tu Aplicación Web**

#### **Paso 1: Preparar la aplicación web**

El artículo detalla cómo preparar tu aplicación para su despliegue. Esto implica asegurarse de que la aplicación esté construida correctamente y sin errores.

#### **Paso 2: Opciones de despliegue**

Azure App Service ofrece varias formas de desplegar tu aplicación web:

* **Azure CLI**: Usa la línea de comandos para enviar tu código a Azure.
* **GitHub Actions o Azure DevOps**: Automatiza el despliegue con pipelines de CI/CD.
* **FTP**: Sube manualmente los archivos vía FTP.
* **Visual Studio Code**: Si usas Visual Studio Code, puedes desplegar directamente desde el editor.

#### **Paso 3: Publicar tu aplicación web**

* El artículo recomienda usar **GitHub** como método de despliegue, donde subes el código de tu aplicación a un repositorio de GitHub y luego conectas ese repositorio con tu **App Service**. Azure soporta **GitHub Actions** para automatizar los despliegues.

### **4. Gestionar tu Aplicación Web**

Una vez que la aplicación esté desplegada, puedes gestionarla a través del **Portal de Azure**.

#### **Escalado:**

* Puedes escalar tu aplicación **hacia arriba** (agregando más recursos) o **hacia afuera** (aumentando el número de instancias) según la demanda. Azure App Service ajusta automáticamente los recursos en función de la carga y el tráfico de la aplicación.

#### **Monitoreo:**

* Usa **Azure Monitor** para realizar un seguimiento del rendimiento y la salud de la aplicación. Puedes ver el uso de la CPU, la memoria, los tiempos de respuesta y más.
* **App Insights**: Utiliza **Azure Application Insights** para obtener diagnósticos en tiempo real y registrar errores.

#### **Configuración de ajustes:**

* Puedes modificar los ajustes de la aplicación (por ejemplo, cadenas de conexión, variables de entorno) directamente en la sección de **Configuración** del **Portal de Azure**.

#### **Copia de seguridad y restauración:**

* Azure App Service permite realizar **copias de seguridad** periódicas de tu aplicación y restaurarlas si es necesario.

### **5. Asegurar tu Aplicación Web**

El artículo destaca cómo asegurar tu aplicación con los siguientes pasos:

* **Certificados SSL**: Activa SSL para asegurar conexiones HTTPS.
* **Autenticación**: Azure App Service soporta múltiples proveedores de autenticación (por ejemplo, Microsoft, Google, Facebook, Twitter) para asegurar tu aplicación.

### **6. Conclusión**

El artículo concluye destacando la facilidad de uso y las potentes características de **Azure App Service**. Ofrece una forma sencilla y eficiente de alojar, escalar y gestionar aplicaciones web sin tener que preocuparse por la infraestructura subyacente. Azure App Service es adecuado tanto para aplicaciones pequeñas como para grandes soluciones empresariales, lo que lo convierte en una opción flexible y confiable para desarrolladores.

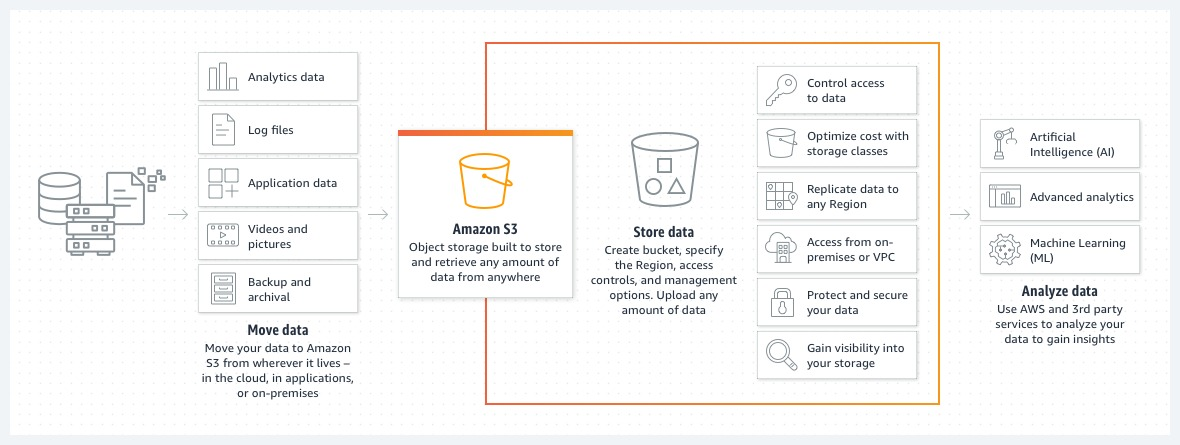
### **Resumen**

En resumen, el artículo proporciona una **guía completa** para configurar y alojar una **aplicación web en Azure App Service**. Los pasos cubiertos incluyen:

1. Crear un **Resource Group**, un **App Service Plan** y una **aplicación web**.
2. Desplegar la aplicación usando **GitHub** u otros métodos.
3. Gestionar y monitorear la salud de la aplicación, escalando según sea necesario.
4. Asegurar la aplicación con SSL y autenticación.

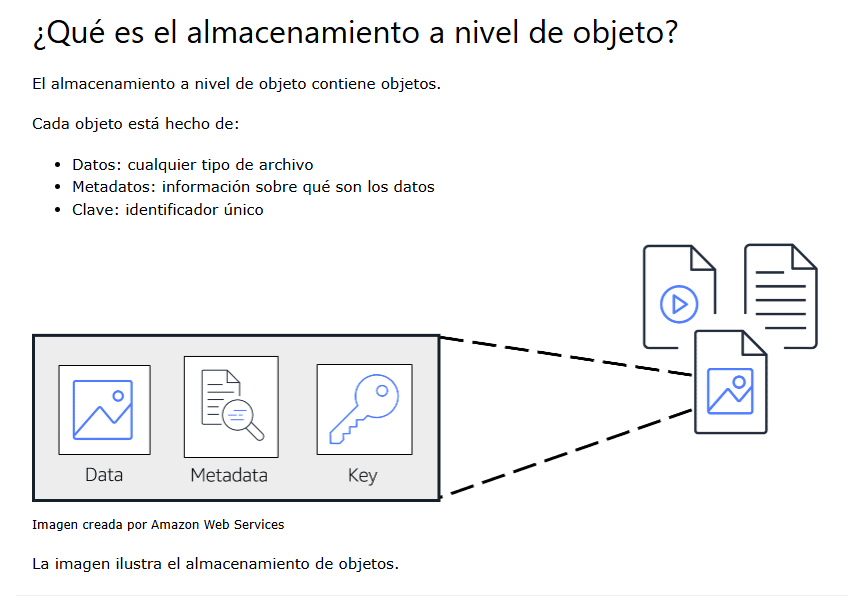
Esta guía paso a paso ayuda a los usuarios a comenzar con el alojamiento de sus aplicaciones en **Azure**, lo que la convierte en un recurso excelente para principiantes.

**Caso Práctico 2**

****

Amazon S3 (Simple Storage Service) es un servicio de almacenamiento de objetos de Amazon Web Services (AWS) que permite almacenar y gestionar grandes volúmenes de datos en la nube de forma segura, escalable y eficiente. Es una solución popular dentro del entorno de AWS, especialmente diseñada para facilitar el almacenamiento de diversos tipos de archivos, desde fotografías y videos hasta logs y backups, siendo una opción ideal para proyectos que requieren alta disponibilidad de datos y rendimiento.

S3 se ha consolidado como una capa de almacenamiento clave para proyectos de big data y data lakes en la nube, ya que permite una integración fluida con herramientas de análisis y procesamiento de datos. Al ser de tipo PaaS (Platform as a Service), los usuarios pueden concentrarse en gestionar y analizar sus datos, mientras AWS se encarga de la infraestructura y seguridad subyacente. Esto convierte a Amazon S3 en una elección confiable para aplicaciones móviles, sitios web estáticos y aplicaciones nativas de la nube, proporcionando la flexibilidad de almacenar cualquier cantidad de datos y acceder a ellos cuando sea necesario.

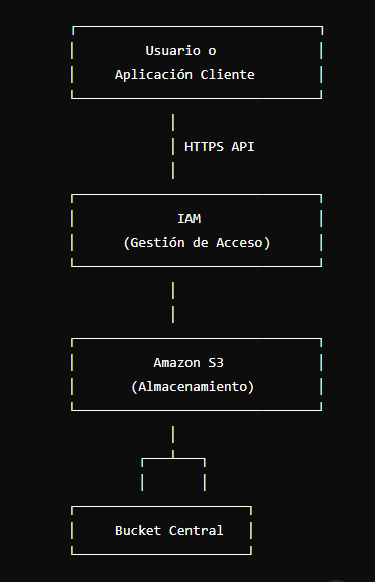


### **1. Funcionamiento Básico de Amazon S3**

**Estructura de almacenamiento**: S3 organiza los datos en "buckets" (contenedores), donde cada bucket puede almacenar una cantidad casi ilimitada de objetos (como archivos de texto, imágenes o videos).

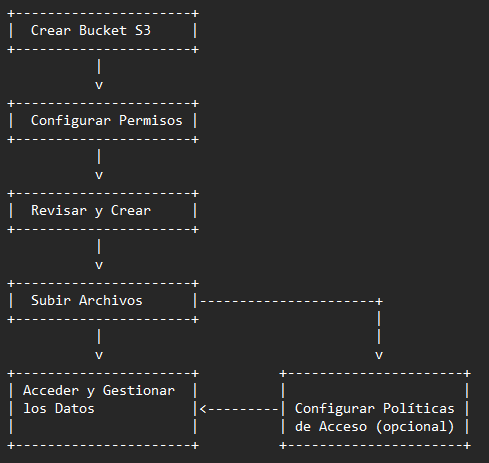
**Objetos**: Cada archivo o unidad de almacenamiento dentro de un bucket se llama "objeto". Los objetos tienen una clave (key) única y están acompañados de metadatos que describen las propiedades del archivo, como su tamaño, fecha de modificación y permisos.

**Acceso mediante API**: S3 permite interactuar con los datos almacenados a través de una API RESTful, lo que facilita integraciones con otras aplicaciones y servicios en la nube.



**Explicación**:

* **Usuario o Aplicación Cliente**: El usuario o aplicación realiza una solicitud para acceder o manipular datos en la nube.
* **HTTPS API**: La solicitud se envía a través de HTTPS, garantizando la encriptación y la protección de los datos en tránsito.
* **IAM (Gestión de Acceso)**: El sistema IAM verifica si el usuario o la aplicación tiene los permisos adecuados para acceder al recurso solicitado. Si la autenticación y autorización son correctas, se otorgan los permisos necesarios.
* **Amazon S3 (Almacenamiento)**: Una vez autenticado, la solicitud se dirige al servicio de almacenamiento en la nube, **Amazon S3**, que gestiona y almacena los datos en la nube.
* **Bucket Central**: Los datos se almacenan en un "bucket" (contenedor) dentro de Amazon S3, el cual centraliza el almacenamiento y organización de la información, facilitando su acceso y gestión.

Proceso de creación de un bucket

1) Crear un bucket

El nombre del bucket debe ser globalmente único, ya que S3 usa este nombre para identificar el bucket en la nube.

Se elige la región donde se almacenarán los datos.

Se puede personalizar la configuración del bucket según las necesidades (por ejemplo, habilitar el versionado de objetos, configurar el cifrado, etc.).

**2) Configurar permisos del bucket**

### En la sección Permissions, se puede establecer políticas de acceso al bucket y controlar quién puede acceder a él (por ejemplo, si se desea que sea público o restringido).

### Se pueden habilitar opciones de Block public access para evitar accesos no deseados.

### También se puedes configurar una policy para establecer reglas específicas de acceso.

### **3) Revisar y crear**

### Revisa todas las configuraciones realizadas y luego de verificar, se crea el bucket.

### **4) Subir archivos al bucket**

### Una vez creado el bucket, ya se encuentra habilitado para cargar archivos manualmente o a través de algún software.

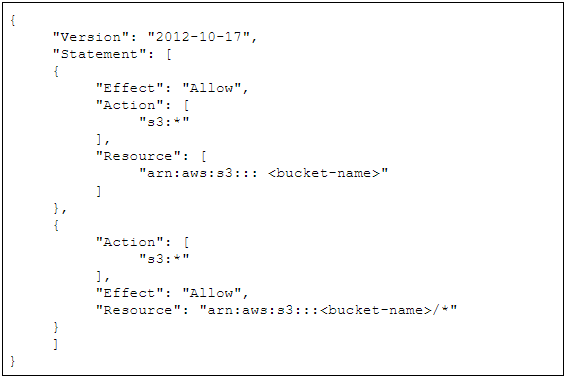
### **5) Configurar políticas de acceso (opcional)**

### Se pueden configurar políticas avanzadas de acceso a través de IAM (Identity and Access Management) o políticas de bucket para controlar cómo los usuarios o aplicaciones pueden interactuar con tu bucket y objetos.

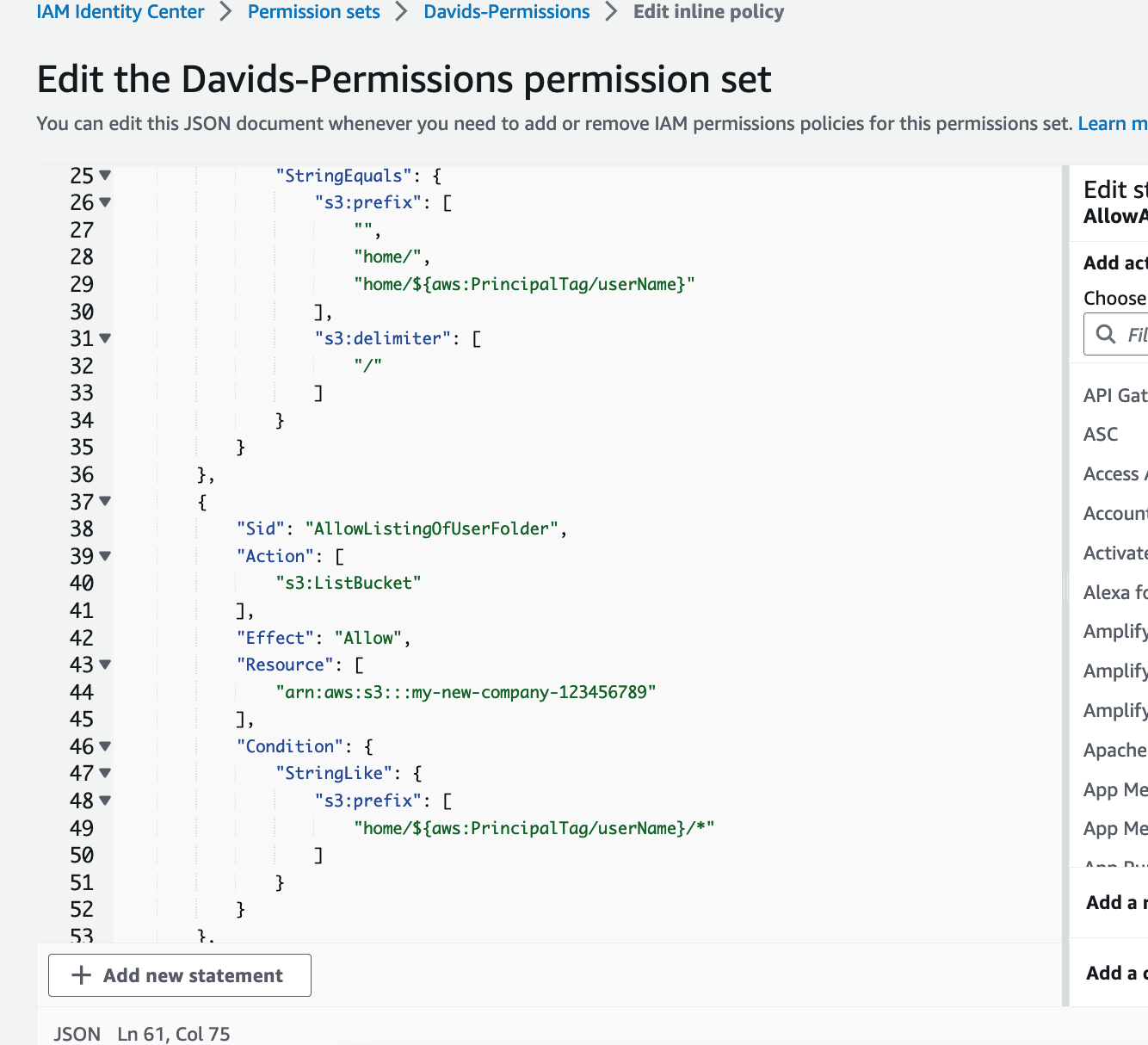
### **6) Acceder y gestionar los datos**

### Se puede acceder a los archivos subidos a través de la consola de S3, o utilizando herramientas como AWS CLI, SDKs o API para integrar S3 en aplicaciones y servicios.

### **2. Seguridad de Acceso en Amazon S3**

Para proteger los datos en Amazon S3, los usuarios solo tienen acceso a los recursos que crean. Para otorgar acceso a otros, se pueden usar características de gestión como AWS Identity and Access Management (IAM), para crear usuarios y gestionar sus permisos, o listas de control de acceso (ACL), para conceder acceso a objetos específicos.

* **Políticas de Bucket y Control de Acceso**: Cada bucket puede tener políticas de acceso para especificar quién puede acceder a los datos. AWS usa políticas basadas en permisos para limitar el acceso a usuarios y servicios específicos.

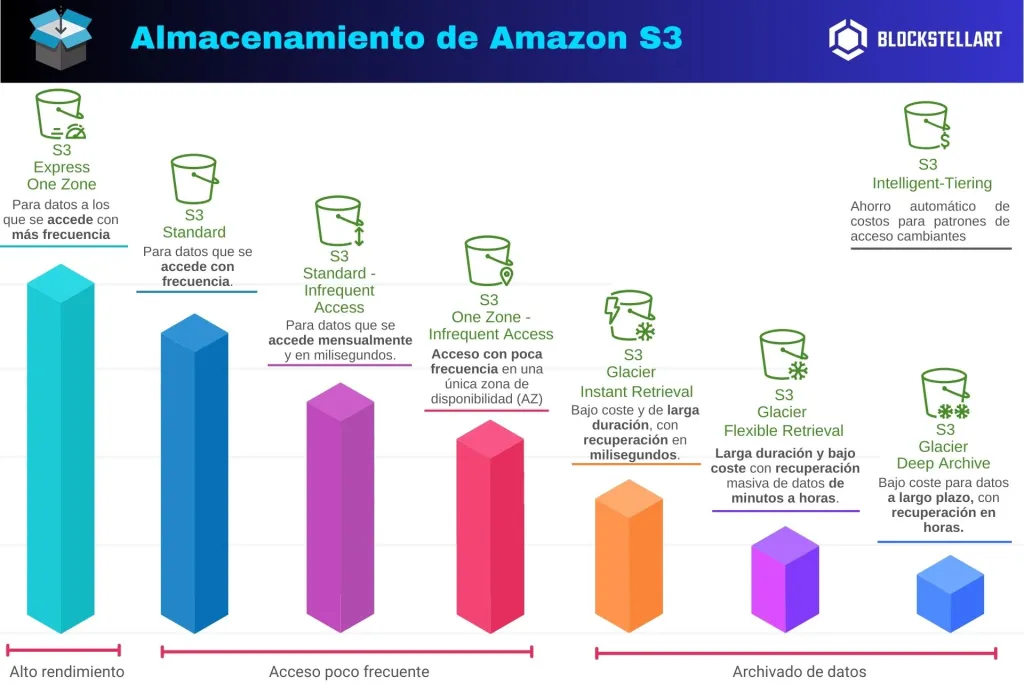


* **IAM (Identity and Access Management)**: IAM permite gestionar roles y permisos a nivel de usuario y grupo, controlando qué acciones específicas pueden realizar dentro de S3 (como leer, escribir o listar objetos en un bucket).
* **Cifrado de Datos**:
  + **Cifrado en reposo**: Amazon S3 permite el cifrado automático de datos almacenados mediante claves administradas por AWS (SSE-S3) o con claves proporcionadas por el cliente (SSE-C o SSE-KMS, este último utilizando AWS Key Management Service).
  + **Cifrado en tránsito**: S3 utiliza SSL/TLS para proteger los datos en tránsito entre el cliente y el servicio.

### **3. Gestión de Datos en Amazon S3**

* **Versionado**: El control de versiones de objetos permite almacenar varias versiones de un objeto en un mismo cubo. Esta función puede proteger los objetos almacenados en Amazon S3 frente a ediciones, sobrescrituras o eliminaciones no deseadas.
* **Lifecycle Policies**: Puedes configurar políticas de ciclo de vida que muevan los objetos automáticamente a diferentes clases de almacenamiento (como S3 Glacier para almacenamiento de bajo costo) o que eliminen objetos después de un tiempo específico.

**Clases de Almacenamiento**: S3 ofrece varias clases de almacenamiento con distintos niveles de costo y acceso, como:



* + S3 Estándar: Almacenamiento de datos en caliente, ideal para datos de uso frecuente (ej. sitios web, aplicaciones en la nube). Tiene altos costos de almacenamiento, pero permite un acceso rápido a los datos.
  + S3 Standard-IA (Acceso Infrecuente): Optimizado para almacenamiento a largo plazo de datos a los que se accede con poca frecuencia. Aunque tiene menores costos de almacenamiento que S3 Estándar, la recuperación de datos tiene un costo adicional.
  + S3 One Zone-IA: Almacenamiento de datos infrecuentes en una sola zona de disponibilidad. Tiene menores costos de almacenamiento y restauración, pero con un menor nivel de redundancia y un tiempo de disponibilidad del 99,5%.
  + S3 Glacier: Para almacenamiento a largo plazo a bajo costo, pero con tiempos de recuperación más largos (desde minutos hasta horas). Es ideal para archivar datos que no necesitan acceso frecuente.
  + S3 Glacier Deep Archive: Similar a Glacier, pero con un tiempo de recuperación aún mayor (12-48 horas) y un costo aún más bajo. Es adecuado para archivado a largo plazo, como backups o datos normativos.
  + S3 Intelligent-Tiering: Una clase que cambia automáticamente entre S3 Estándar y S3 Standard-IA según los patrones de acceso a los datos, optimizando el costo sin comprometer el rendimiento.