**TEMA 1**

**Conceptos Básicos de Internet**

* **Internet**: Red global de ordenadores para compartir recursos e intercambiar información, utilizando el protocolo **TCP/IP**.
* **Servicios**: Incluyen **WWW**, **FTP**, **DNS**, **email**, **SSH**, entre otros.

**Introducción**

* **Web**: Sistema organizado por volúmenes de información interconectados mediante **HTML** (HyperText Markup Language), el estándar mundial definido por el **W3C**.
* **Navegadores**: Software para interpretar documentos HTML.

**Arquitecturas de Aplicaciones Web**

1. **Cliente-Servidor**: Divide en dos capas; el cliente solicita recursos al servidor, que responde a través de protocolos como **HTTP/HTTPS**.
   * **Servidores web**: Privativos (IIS, WebLogic) y open source (Apache, NginX).
2. **Tres Capas**: Evolución de la arquitectura cliente-servidor.
   * **Capa de presentación**: Interfaz de usuario.
   * **Capa de negocio**: Lógica del negocio.
   * **Capa de persistencia**: Almacenamiento de datos.
3. **Más de Tres Capas**: Pueden incluir capas adicionales como:
   * **Capa de control**: Manejo de solicitudes y flujo de trabajo.
   * **Capa de servicios**: Intermediaria entre control y datos.
4. **Microservicios**: Estilo arquitectónico que divide la aplicación en servicios pequeños y autónomos, permitiendo despliegue y escalabilidad independientes.
   * **Características**: Descomposición funcional, comunicación a través de APIs, desarrollo autónomo.
   * **Ventajas**: Flexibilidad, escalabilidad, resiliencia.
   * **Desventajas**: Complejidad operativa y de comunicación, consistencia de datos.

**Tipos de Aplicaciones Web**

* **Estáticas**: Páginas HTML sin interacción (Web 1.0).
* **Dinámicas**: Interacción del usuario que genera cambios en la visualización, usando HTML, CSS, y lenguajes de servidor como PHP.
* **Interactivas/Híbridas**: Diálogo entre cliente y servidor, utilizando tecnologías como **AJAX** (Web 2.0 y más allá).

**Lenguajes de Programación Utilizados**

* **Del lado del cliente**: HTML, CSS, JavaScript, XML, JSON.
* **Del lado del servidor**: PHP, Python, Ruby, Node.js, Java (JSP, Servlets), ASP.NET, entre otros.

**Integración con Servidores Web**

* Las peticiones a los servidores deben tener un formato específico (URL).
* **Métodos de Intercambio**:
  + **GET**: Solicitar información.
  + **POST**: Enviar datos al servidor.

**Herramientas de Programación**

* **Navegadores**: Para visualizar aplicaciones web.
* **Editores de Texto**: Para escribir código HTML.
* **Entornos de Programación**: Para editar, compilar y ejecutar programas.
* **Herramientas de Tratamiento de Imágenes**: Para gestionar contenido gráfico.
* **Herramientas para Bases de Datos**: Para la carga y mantenimiento de datos.

**AMAZON AWS**

### ****Amazon Virtual Private Cloud (VPC)****

#### ****Definición:****

* Una **VPC** es una red virtual aislada dentro de AWS donde puedes definir y controlar aspectos como el rango de direcciones IP, la subred, las tablas de rutas y las puertas de enlace.

#### ****Componentes Clave:****

* **CIDR (Classless Inter-Domain Routing):** Se utiliza para definir rangos de direcciones IP. Por ejemplo, 10.0.0.0/16 proporciona un rango de direcciones IP que puede ser dividido en subredes.
* **Subredes:** Segmentos dentro de la VPC. Se pueden clasificar como:
  + **Públicas:** Accesibles desde Internet.
  + **Privadas:** No tienen acceso directo a Internet.
* **Tablas de Rutas:** Controlan el tráfico de red entre subredes y el Internet. Cada subred debe estar asociada con una tabla de rutas.
* **Puerta de Enlace de Internet:** Permite que las instancias dentro de la VPC se comuniquen con Internet. Es esencial para que las subredes públicas puedan recibir tráfico externo.

### ****Grupos de Seguridad****

#### ****Definición:****

* Un **grupo de seguridad** actúa como un cortafuegos virtual que controla el tráfico de las instancias EC2. Especifica qué tráfico está permitido o denegado.

#### ****Características:****

* Permite definir reglas basadas en:
  + **Protocolos:** TCP, UDP, ICMP.
  + **Puertos:** Se puede permitir o denegar tráfico en puertos específicos, como el puerto 80 para HTTP.
  + **Direcciones IP de origen/destino:** Permite especificar qué direcciones pueden acceder a la instancia.

### ****Instancias EC2 (Elastic Compute Cloud)****

#### ****Definición:****

* Las instancias EC2 son máquinas virtuales que puedes usar para ejecutar aplicaciones en la nube de AWS. Puedes elegir entre diferentes sistemas operativos, tipos de instancia y configuraciones de almacenamiento.

#### ****Tipos de Instancias:****

* AWS ofrece varios tipos de instancias, cada una optimizada para diferentes cargas de trabajo:
  + **T2 (General Purpose):** Balancean recursos de computación, memoria y red.
  + **M5 (Balanced):** Proporcionan un buen equilibrio de recursos.
  + **C5 (Compute Optimized):** Optimizadas para aplicaciones con alta demanda de CPU.

#### ****Conexiones:****

* Las instancias pueden conectarse a través de SSH (Secure Shell) para administración remota.
* Las credenciales, como los archivos .pem, son esenciales para acceder a las instancias de manera segura.

### ****IP Elásticas****

#### ****Definición:****

* Una **Elastic IP** es una dirección IP pública que puedes asociar y desasociar a tus instancias EC2 según sea necesario. Garantiza que la dirección IP permanezca constante incluso si la instancia se detiene o reinicia.

#### ****Uso:****

* Las Elastic IP son útiles para aplicaciones que requieren un punto de acceso constante desde Internet. Permiten que las instancias mantengan la misma dirección IP a pesar de ser reiniciadas.

### ****Instalación de Servidores Web****

#### ****Apache y PHP:****

* **Apache:** Es un servidor web de código abierto que permite servir contenido web a través de HTTP.
* **PHP:** Un lenguaje de programación del lado del servidor que se utiliza comúnmente para el desarrollo web y se integra fácilmente con Apache.

AWS CLOUD

#### ****Definición:****

* **AWS Cloud9** es un entorno de desarrollo integrado (IDE) basado en la nube que permite a los desarrolladores escribir, ejecutar y depurar código desde cualquier lugar con conexión a Internet.

#### ****Objetivos de AWS Cloud9:****

* Simplificar el desarrollo de software eliminando la complejidad de la configuración local.
* Mejorar la colaboración entre equipos de desarrollo mediante la posibilidad de compartir el entorno de trabajo.
* Ofrecer escalabilidad y acceso a los recursos de AWS, facilitando el desarrollo de aplicaciones en un entorno dinámico.

### ****Características Clave de AWS Cloud9:****

#### ****1. Acceso Basado en la Nube:****

* Permite a los desarrolladores trabajar desde cualquier lugar sin necesidad de configurar entornos locales.

#### ****2. Colaboración en Tiempo Real:****

* Varios desarrolladores pueden trabajar simultáneamente en el mismo proyecto, viendo y editando el código en tiempo real.

#### ****3. Integración con Servicios de AWS:****

* AWS Cloud9 se integra fácilmente con otros servicios de AWS, como AWS Lambda, AWS EC2, y servicios de almacenamiento, facilitando el desarrollo de aplicaciones que utilizan múltiples servicios en la nube.

#### ****4. Herramientas de Desarrollo Preinstaladas:****

* Viene con herramientas comunes como depuradores, terminales y herramientas de línea de comandos preinstaladas, como Node.js, Python, y Git, entre otros.

### ****Creación de un Entorno de Desarrollo en AWS Cloud9:****

#### ****1. Tipos de Entornos:****

* **Nueva instancia de Amazon EC2:** Se crea un entorno nuevo utilizando una instancia de EC2.
* **Instancia de EC2 existente:** Utiliza una instancia ya creada y accesible.
* **Instancias externas:** Puede conectarse a máquinas externas mediante un punto de enlace SSH.

#### ****2. Configuración de la Instancia EC2:****

* **Tipo de Instancia:** Se elige un tipo de instancia (por ejemplo, t3.medium).
* **Plataforma:** Generalmente se utiliza Amazon Linux 2.
* **Timeout:** Define el tiempo máximo que el entorno puede permanecer inactivo antes de hibernarse.

#### ****3. Configuración de Red:****

* **VPC (Virtual Private Cloud):** Selección de la VPC donde se desplegará la instancia.
* **Subred:** Selección de la subred en la que se ubicará la instancia, asegurando que haya suficiente capacidad.

### ****Desarrollo y Ejecución de Código en AWS Cloud9:****

#### ****1. Edición de Código:****

* Los desarrolladores pueden crear y editar archivos directamente en el IDE utilizando un editor integrado.

#### ****2. Ejecución de Código:****

* AWS Cloud9 permite ejecutar el código de varias maneras, incluyendo el uso de comandos en la terminal integrada o mediante botones en la interfaz del IDE.

#### ****3. Herramientas de Depuración:****

* AWS Cloud9 incluye un depurador que permite establecer puntos de ruptura, observar variables y controlar la ejecución del código.

### ****Integración con Git:****

#### ****1. Control de Versiones:****

* AWS Cloud9 está integrado con Git, permitiendo a los desarrolladores realizar operaciones como commit, push, y pull directamente desde el IDE.

#### ****2. Clonación de Repositorios:****

* Los desarrolladores pueden clonar repositorios desde GitHub u otras plataformas para trabajar en proyectos existentes.

### ****Uso de AWS Cloud9 en Máquinas Externas:****

#### ****1. Entornos Externos:****

* AWS Cloud9 puede configurarse en máquinas Linux externas, permitiendo a los desarrolladores utilizar su propia infraestructura.

#### ****2. Requisitos:****

* Se requiere acceso SSH a la máquina Linux y la instalación de Node.js y otras dependencias necesarias para que el entorno funcione correctamente.

### ****Configuración de AWS CLI:****

* **AWS Command Line Interface (CLI):** Permite interactuar con los servicios de AWS desde la terminal. Es fundamental para gestionar recursos de AWS de manera programática.

### ****Limpieza de Recursos:****

* Es esencial eliminar los entornos creados y cerrar el laboratorio para evitar el consumo innecesario de créditos de AWS Academy.

**AZURE**

**Definición y Propósito:** Las máquinas virtuales (VM) en Azure son recursos informáticos escalables que permiten a los usuarios tener un control completo sobre su entorno. Se utilizan para diversas aplicaciones, como desarrollo y pruebas de software, ejecución de aplicaciones en la nube y expansión de centros de datos, sin la necesidad de hardware físico.

**Ventajas de Usar Máquinas Virtuales:**

1. **Flexibilidad:** Permiten a los usuarios crear entornos virtuales según sus necesidades específicas, con configuraciones personalizadas.
2. **Escalabilidad:** Los usuarios pueden aumentar o disminuir el número de VM según la demanda de sus aplicaciones.
3. **Ahorro de Costos:** Solo se paga por los recursos utilizados, activando o desactivando máquinas virtuales según sea necesario.

**Consideraciones Previas a la Creación:** Antes de crear una máquina virtual, es esencial considerar:

* **Nombres de los recursos:** Identificadores únicos para los recursos en Azure.
* **Ubicación de los recursos:** Regiones geográficas donde se almacenarán.
* **Tamaño de la VM:** Configuración de recursos como CPU y memoria.
* **Número máximo de VMs:** Límite en la creación de instancias según sus necesidades.
* **Sistema operativo:** Selección del sistema operativo que se instalará en la VM.
* **Configuración post-inicio:** Ajustes que se aplicarán una vez que la máquina esté activa.
* **Recursos relacionados:** Otros recursos que la VM puede necesitar.

**Recursos que Apoyan una Máquina Virtual:** Al crear una VM, se generan recursos adicionales que tienen sus propios costos:

* **Red Virtual:** Permite la comunicación entre la VM y otros recursos.
* **Interfaz de Red Virtual (NIC):** Conexión a la red virtual, con un límite en la cantidad según el tamaño de la VM.
* **Direcciones IP:** Para la comunicación, incluyendo IPs privadas y, opcionalmente, públicas.
* **Grupo de Seguridad de Red (NSG):** Controla el tráfico de red hacia y desde la VM.
* **Discos:** Incluye un disco del sistema operativo y, opcionalmente, discos de datos. Es recomendable separar los datos en discos distintos.
* **Licencias:** Costo variable según el sistema operativo y la cantidad de núcleos en la VM.

**Ubicaciones de Recursos:** Los recursos de Azure se pueden crear en diferentes regiones geográficas. La selección de la ubicación afecta dónde se almacenarán los discos virtuales. Las ubicaciones se pueden consultar a través del Azure Portal, Azure PowerShell, la API REST o Azure CLI.

**Opciones de Disponibilidad:** Azure ofrece varias opciones para garantizar la disponibilidad de las máquinas virtuales:

* **Zonas de Disponibilidad:** Zonas separadas físicamente en una región, asegurando una conectividad del 99.99% para al menos una instancia.
* **Escalado de Máquinas Virtuales (Scale Sets):** Permiten crear y administrar grupos de VMs con balanceo de carga, ajustando automáticamente el número de instancias según la demanda.