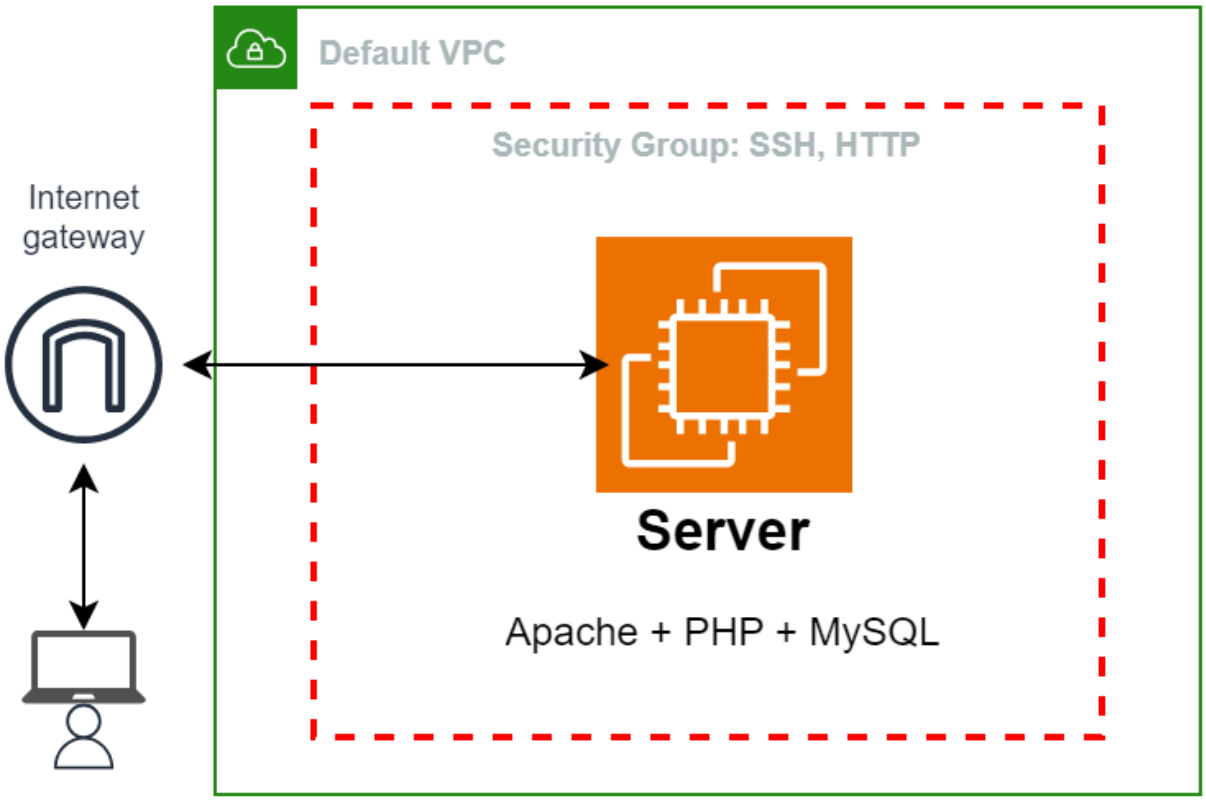
IES Abastos – DAW 7K
Despliegue de Aplicaciones Web
Oscar Pereira

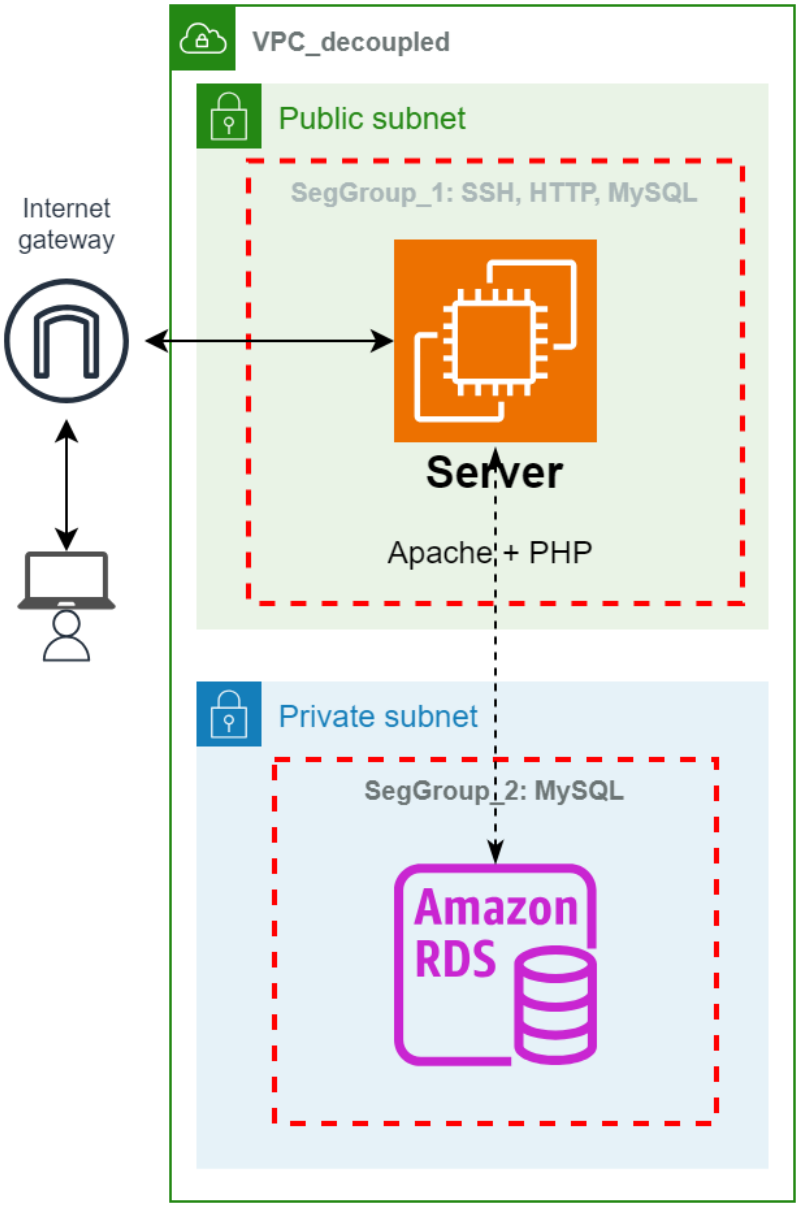
# Apuntes Alta Disponibilidad

## 1. Arquitecturas Principales

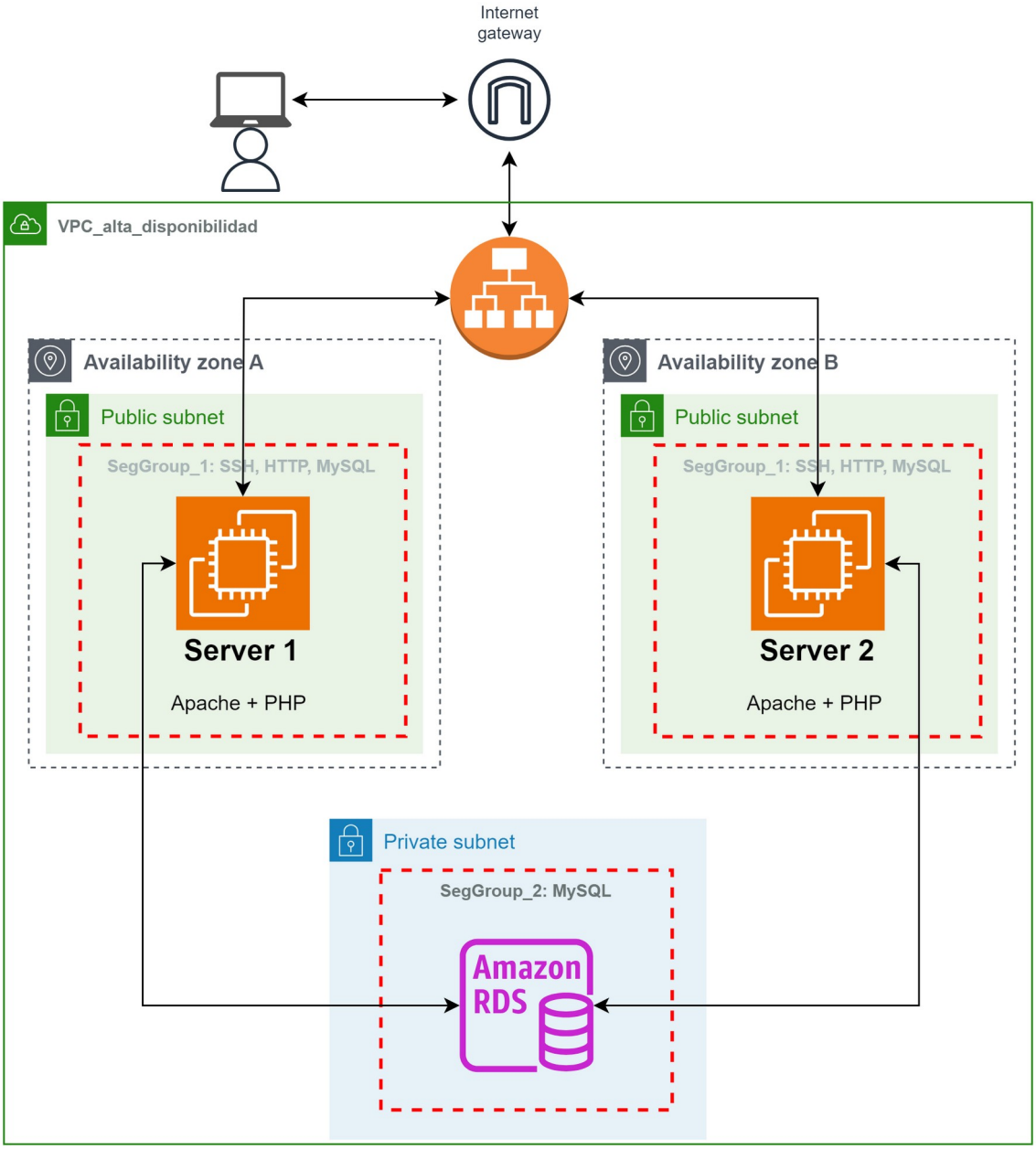
- **Arquitectura Monolítica**:
  - **Definición**: Todos los componentes (Apache, PHP, MySQL) están en una sola máquina.
  - **Ventajas**:
    - Fácil y rápida implementación.
    - Bajo coste inicial.
    - Ideal para pruebas de concepto.
  - **Desventajas**:
    - Escalado vertical (solo mejorando hardware).
    - Poca flexibilidad y mantenimiento complejo en entornos exigentes.



- **Arquitectura Desacoplada**:
    - **Definición**: Separación en dos capas: servidor web (Apache + PHP) y base de datos (RDS).
    - **Ventajas**:
        - Escalado horizontal independiente.
        - Administración simplificada de la base de datos.
    - **Desventajas**:
        - Mayor coste y complejidad de configuración.



- **Arquitectura de Tres Capas**:
    - Balanceo de carga y autoescalado dinámico para distribuir tráfico entre múltiples instancias.



**2. Conceptos Clave**

- **Alta Disponibilidad**:
    - Distribución de tráfico entre múltiples máquinas y zonas de disponibilidad (AZ).
    - Uso de **Balanceadores de Carga (ELB)** para direccionar peticiones.
    - **Amazon RDS**: Base de datos gestionada con replicación, respaldos automáticos y alta disponibilidad.
- **Escalabilidad**:
    - **Vertical**: Aumentar recursos de una instancia (CPU, RAM).
    - **Horizontal**: Añadir más instancias.
    - **Autoescalado**: Ajuste automático del número de instancias según demanda (basado en métricas como uso de CPU).
- **VPC (Virtual Private Cloud)**:
    - Redes públicas (acceso a internet) y privadas (solo comunicación interna).
    - Grupos de seguridad para controlar tráfico (SSH, HTTP, MySQL).

---

**3. Servicios AWS Relevantes**

- **EC2**: Servidores virtuales escalables.
- **RDS**: Base de datos relacional gestionada (alta disponibilidad integrada).
- **Elastic Load Balancer (ELB)**: Distribuye tráfico entre instancias.
- **Auto Scaling Groups**: Grupos de instancias que escalan automáticamente.

---

## Amazon RDS: Resumen para Estudio

---

**Definición**

Amazon Relational Database Service (RDS) es un servicio de bases de datos **relacionales administrado** por AWS. Proporciona:

- **Escalado automático** (vertical/horizontal).
- 
- **Tolerancia a fallos** (réplicas en múltiples AZ).
- **Alta disponibilidad** (backups automáticos, recuperación ante desastres).
- Compatibilidad con motores populares: **MySQL, Aurora, PostgreSQL, SQL Server, MariaDB, Oracle**.

- Coste similar a una instancia EC2, pero **no se apaga al cerrar el laboratorio** (ideal para producción).

---

**Niveles de Administración de SGBD**

| Responsabilidad | En las instalaciones | EC2 | Amazon RDS |
|---|---|---|---|
| **Optimización de aplicaciones** | ✔️ Usuario | ✔️ Usuario | ✔️ Usuario |
| **Escalado** | ✔️ Usuario | ✔️ Usuario | ✔️ AWS |
| **Alta disponibilidad** | ✔️ Usuario | ✔️ Usuario | ✔️ AWS |
| **Respaldos de BD** | ✔️ Usuario | ✔️ Usuario | ✔️ AWS |
| **Parches de software de BD** | ✔️ Usuario | ✔️ Usuario | ✔️ AWS |
| **Instalación de software de BD** | ✔️ Usuario | ✔️ Usuario | ✔️ AWS |
| **Parches del sistema operativo** | ✔️ Usuario | ✔️ AWS | ✔️ AWS |
| **Instalación del sistema operativo** | ✔️ Usuario | ✔️ AWS | ✔️ AWS |
| **Mantenimiento del servidor físico** | ✔️ Usuario | ✔️ AWS | ✔️ AWS |
| **Alimentación/climatización/red** | ✔️ Usuario | ✔️ AWS | ✔️ AWS |

---

**Ventajas de cada Modelo**

1. **En las instalaciones**:
    - Control total sobre hardware y software.
    - Ideal para entornos con requisitos de seguridad o normativos estrictos.
2. **EC2**:
    - AWS gestiona la infraestructura física y el SO.
    - Flexibilidad para personalizar la configuración de la BD.
3. **RDS**:
    - **Máxima simplicidad**: AWS gestiona la BD, el SO y la infraestructura.
    - Ideal para equipos que priorizan el desarrollo sobre la administración de BD.
    - Alta disponibilidad y respaldos automáticos sin esfuerzo.

---

**Desventajas**

- **RDS**: Menor control sobre configuraciones avanzadas de la BD (ej: parámetros específicos del motor).
- **EC2**: Mayor carga administrativa (parches, escalado manual).
- **En las instalaciones**: Coste y complejidad elevados (infraestructura física, personal especializado).

---

## TL;DR

- **RDS**: AWS maneja **todo** excepto la optimización de la aplicación. Ideal para proyectos que necesitan una BD lista para producción sin gestión operativa.
- **EC2**: AWS gestiona el servidor físico y el SO, pero el usuario administra la BD. Ofrece flexibilidad con algo de carga administrativa.
- **En las instalaciones**: Todo es responsabilidad del usuario. Solo recomendado para casos específicos (ej: normativas estrictas).

## TL;DR Práctico

**Pasos Clave para Configurar un Entorno Escalable en AWS**

1. **Crear VPC**:
    - 2 subredes públicas y 2 privadas en 2 AZ (sin NAT/S3 Gateway).
    - Grupos de seguridad con reglas para SSH, HTTP y MySQL.
2. **Servidor Desacoplado**:
    - **EC2**: Instalar Apache + PHP (`sudo apt install apache2 php php-mysql`).
    - **RDS**: Crear BD MySQL con usuario maestro y contraseña. Configurar grupo de seguridad para permitir conexiones desde EC2.
3. **Conectar Servidor y BD**:
    - Modificar `dbinfo.inc` para apuntar al endpoint de RDS.
4. **Balanceador de Carga y Autoescalado**:
    - Crear AMI de la instancia EC2 configurada.
    - Lanzar instancias desde la AMI en múltiples AZ.
    - Configurar **Application Load Balancer** (HTTP) y grupo de destino.
    - **Auto Scaling Group**: Definir políticas basadas en uso de CPU (mín. 1, máx. 4 instancias).
5. **Pruebas**:
    - Usar el DNS del balanceador para verificar distribución de tráfico.
    - Simular carga con `stress --cpu 50 -t 500` para activar autoescalado.