

ALTA DISPONIBILIDAD Y ESCALABILIDAD

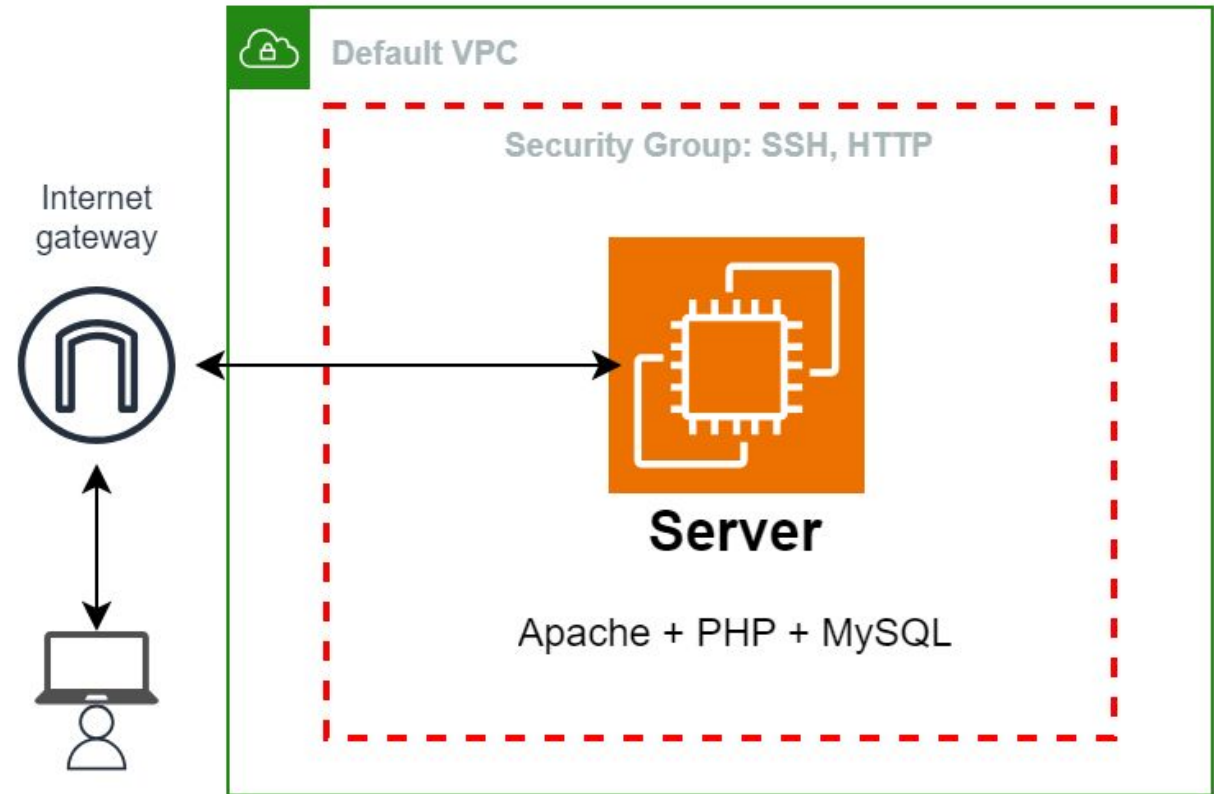
TEMA 7

Contenido

- Arquitectura Monolítica
- Arquitectura Desacoplada
- Arquitectura Tres Capas
 - Balanceo de carga
 - Auto escalado dinámico

Arquitectura monolítica

- En una misma máquina:
 - Apache + PHP + MySQL
- Ventajas
 - Fácil y rápida de construir.
 - Bajo coste
 - Útil para pruebas de concepto
- Inconvenientes
 - Poco flexible
 - Sólo admite escalado vertical
 - En escenarios exigentes requiere hardware potente



Crear Servidor Monolítico (1)

- Crea una EC2 con nombre `Server_Monolitico`
 - AMI: **UBUNTU**
 - Tipo: **t2.micro**
 - Par de claves: **vockey**
 - VPC default
 - IP pública
 - Grupo seguridad: SSH y HTTP

- Instalar Apache y PHP

```
sudo apt update -y
```

```
sudo apt install apache2 php php-mysql -y
```

- Instalar MySQL en local

```
sudo apt install mysql-server -y
```

Crear Servidor Monolítico (2)

- Apache no se ejecuta con permisos de root, por lo que necesitamos crear un usuario para la base de datos
- Entrar en base de datos como *root* y crear nuevo usuario *tutorial_user*

```
sudo mysql -u root
```

- `CREATE DATABASE sample;`
- `CREATE USER 'tutorial_user'@'localhost' IDENTIFIED BY 'tutorial';`
- `GRANT ALL ON sample.* TO 'tutorial_user'@'localhost';`
- `exit`

- Entrar de nuevo como *tutorial_user* y password *tutorial*

```
mysql -u tutorial_user -p
```

- `SHOW databases;`

Crear Servidor Monolítico (3)

- Activar Apache y habilitar que se inicie siempre al arrancar

```
sudo systemctl start apache2
```

```
sudo systemctl enable apache2
```

- Crear archivos

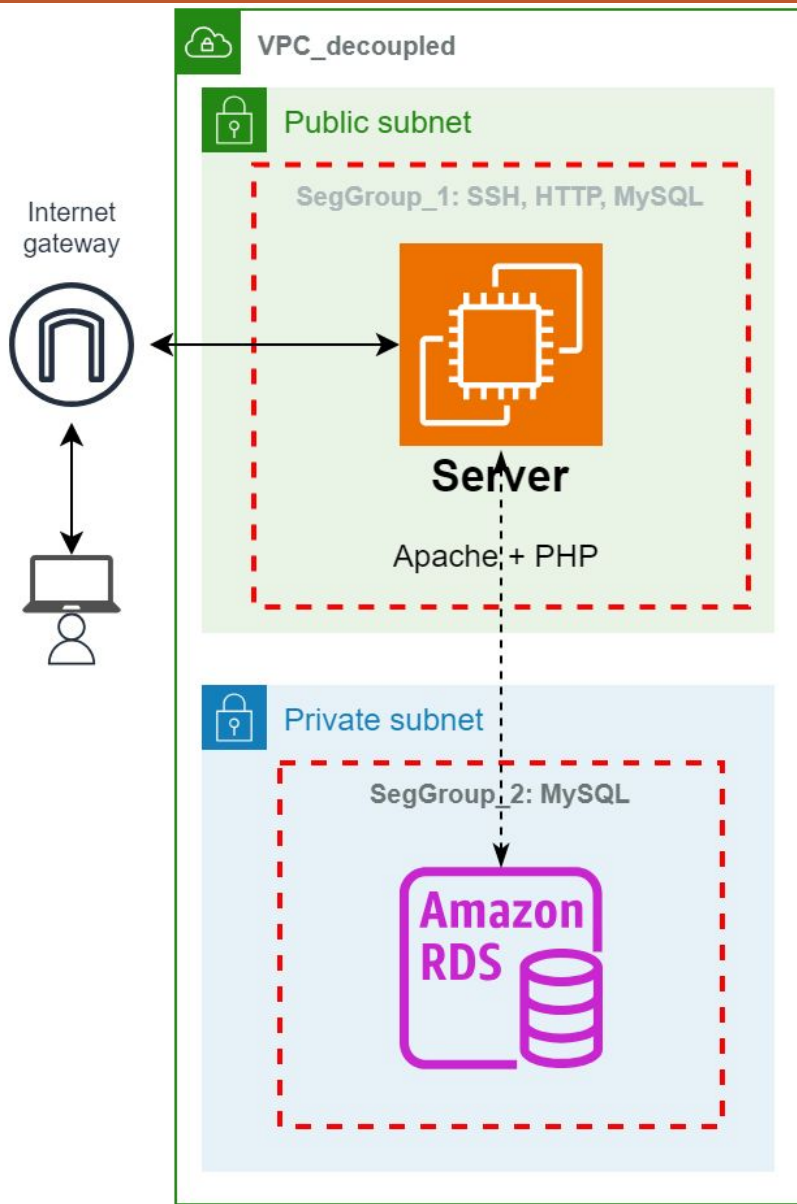
```
/var/www/html/index.php
```

```
/var/www/inc/dbinfo.inc
```

- En *dbinfo.inc*, cambiar nombre del servidor de BD por localhost

- `define('DB_SERVER', 'localhost');`

- Probar



Arquitectura desacoplada

- Dos capas:
 - Servidor Apache + PHP
 - Servidor RDS con MySQL
- Ventajas
 - Más flexible y fácil de escalar
 - Administración de la DB más sencilla
- Inconvenientes
 - Mayor coste y complejidad

Amazon RDS

- Amazon Relational Database Service
- Servicio de bases de datos administrado:
 - Escalado, tolerancia a errores y disponibilidad están incluidas en el servicio
- Compatible con:
 - MySQL, Amazon Aurora, Microsoft SQL Server, PostgreSQL, MariaDB y Oracle
- Coste similar a instancia EC2
 - **Pero no se apaga** cuando cerramos el laboratorio

Niveles de administración de SGBD

Base de datos en las instalaciones

- Optimización de aplicaciones
- Escalado
- Alta disponibilidad
- Respaldos de bases de datos
- Parches de software de base de datos
- Instalación del software de la base de datos
- Parches del sistema operativo
- Instalación del sistema operativo
- Mantenimiento del servidor
- Alimentación, climatización y red

AWS
proporciona

Base de datos en EC2

- Optimización de aplicaciones
- Escalado
- Alta disponibilidad
- Respaldos de bases de datos
- Parches de software de base de datos
- Instalación del software de la base de datos
- Parches del sistema operativo
- Instalación del sistema operativo
- Mantenimiento del servidor
- Alimentación, climatización y red

AWS
proporciona

Base de datos en Amazon RDS

- Optimización de aplicaciones
- Escalado
- Alta disponibilidad
- Respaldos de bases de datos
- Parches de software de base de datos
- Instalación del software de la base de datos
- Parches del sistema operativo
- Instalación del sistema operativo
- Mantenimiento del servidor
- Alimentación, climatización y red

1. Crear Servidor

- Crea VPC:
 - Nombre: **VPC_alta_disp**
 - Contenido: **2 redes públicas y 2 privadas en 2 AZ, sin NAT ni S3 Gateway**

Vista previa

VPC [Mostrar detalles](#)

Su red virtual de AWS

VPC_alta_disp-vpc

Subredes (4)

Subredes dentro de esta VPC

us-east-1a

VPC_alta_disp-subnet-public1-us-

VPC_alta_disp-subnet-private1-us-

us-east-1b

VPC_alta_disp-subnet-public2-us-

VPC_alta_disp-subnet-private2-us-

Tablas de enrutamiento (3)

Dirigir el tráfico de red a los recursos

VPC_alta_disp-rtb-public

VPC_alta_disp-rtb-private1-us-east-1a

VPC_alta_disp-rtb-private2-us-east-1b

Conexiones de red (1)

Conexiones a otras redes

VPC_alta_disp-igw



1. Crear Servidor

- Crea VPC:
 - Nombre: **VPC_alta_disp**
 - Contenido: **2 redes públicas y 2 privadas en 2 AZ, sin NAT ni S3 Gateway**
- Crea una EC2 con nombre **Server_Desacoplado**
 - AMI: **UBUNTU**
 - Tipo: **t2.micro**
 - Par de claves: **vockey**
 - **VPC_alta_disp, subred public 1**
 - IP pública
 - Grupo seguridad: **SSH, HTTP y MySQL/Aurora**

▼ Configuraciones de red [Información](#)

VPC - required [Información](#)

vpc-08416b08f1156b872 (VPC_alta_disp-vpc)
10.0.0.0/16

Subred [Información](#)

subnet-01dadb9f5803a3242 VPC_alta_disp-subnet-public1-us-east-1a
VPC: vpc-08416b08f1156b872 Propietario: 051049719530
Zona de disponibilidad: us-east-1a Direcciones IP disponibles: 4091 CIDR: 10.0.0.0/20

Asignar automáticamente la IP pública [Información](#)

Habilitar

Firewall (grupos de seguridad) [Información](#)
Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agrega reglas para permitir que el tráfico específico llegue a la instancia.

☒ Crear grupo de seguridad ☐ Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad - obligatorio

ssh_y_http

1. Crear Servidor

- Crea VPC:
 - Nombre: **VPC_alta_disp**
 - Contenido: **2 redes públicas y 2 privadas en 2 AZ, sin NAT ni S3 Gateway**
- Crea una EC2 con nombre **Server_Desacoplado**
 - AMI: **UBUNTU**
 - Tipo: **t2.micro**
 - Par de claves: **vockey**
 - **VPC_alta_disp, subred public 1a**
 - IP pública
 - Grupo seguridad: SSH, HTTP y MySQL/Aurora

- Instalar Apache y PHP

```
sudo apt update -y
```

```
sudo apt install apache2 php php-mysql -y
```

2. Crear base de datos en RDS

- Busca RDS y pulsa **Crear base de datos**
- Selecciona **Creación estándar, MySQL y Capa gratuita**
- Usuario maestro: **tutorial_user**
- Contraseña: **tutorial**
- Conectividad:
 - **No se conecte a un recurso informático EC2**
 - **VPC_alta_disp**
 - Acceso público: **NO**
 - Crear nuevo grupo de seguridad: **database_SG**
- Configuración adicional → Nombre de BD inicial: **sample**

2. Crear base de datos en RDS

- Busca RDS y pulsa **Crear base de datos**
- Selecciona **Creación sencilla, MySQL y Capa gratuita**
- Usuario maestro: **tutorial_user**
- Contraseña: **password123**
- Conexión: **MySQL**
 - No se permite el acceso público: **NO**
 - VPC: **VPC_alta_disp**
 - Acceso público: **NO**
 - Crear nuevo grupo de seguridad: **database_SG**
- Configuración adicional → Nombre de BD inicial: **sample**

**ESPERA UNOS MINUTOS A QUE
SE CREE LA BASE DE DATOS**

MIENTRAS...

3. Conectar Servidor y BD (1)

- En el nuevo grupo de seguridad database_SG
 - Cambia el origen de las peticiones MYSQL a ***Anywhere***
- Activar Apache y habilitar que se inicie siempre al arrancar

```
sudo systemctl start apache2
```

```
sudo systemctl enable apache2
```
- Crear archivos

```
/var/www/html/index.php
```

```
/var/www/inc/dbinfo.inc
```

3. Conectar Servidor y BD (2)

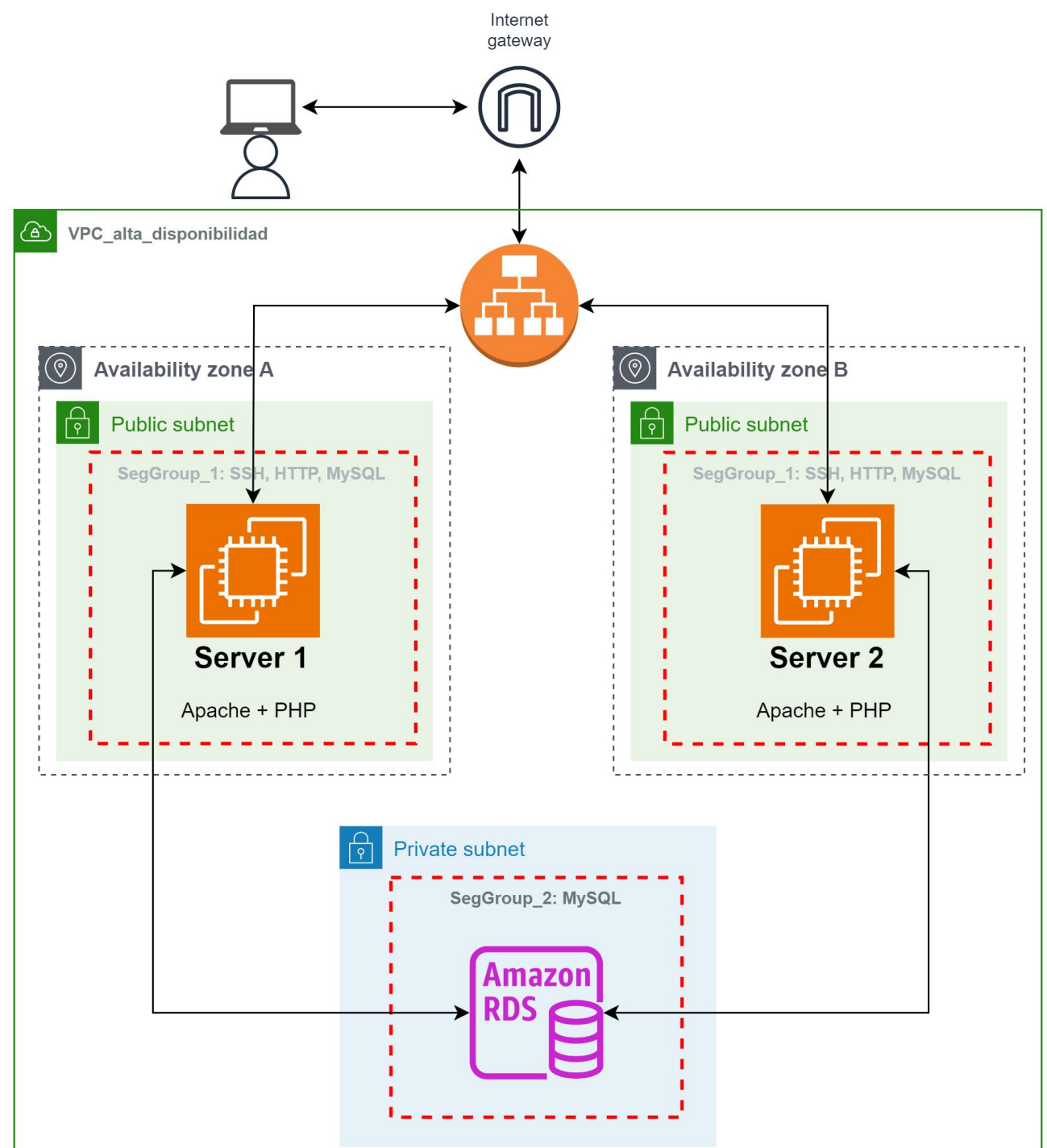
- Cuando la DB esté disponible
- Copiar nombre del punto de enlace
- En *dbinfo.inc*, cambiar nombre del servidor de BD por punto de enlace RDS
 - `define('DB_SERVER',
 'rds_endpoint_name');`
- Probar



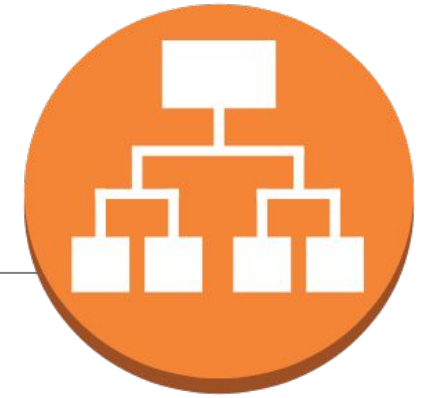
Conectividad y seguridad	
Punto de enlace y puerto	Redes
Punto de enlace database-1.cdymuo60gotn.us-east-1.rds.amazonaws.com	Zona de disponibilidad us-east-1b
Puerto 3306	VPC VPC_alta_disp-vpc (vpc-08416b08f1156b872)
	Grupo de subredes default-vpc-08416b08f115

Alta disponibilidad

El tráfico se distribuye entre varias máquinas que usan la misma base de datos



Balanceador de carga



Elastic Load Balancer (ELB) distribuye el tráfico entrante entre varios destinos en una o varias (AZ).

- Nivel de aplicación: HTTP o HTTPS
- Nivel de red: IP, puerto y protocolo de transporte (TCP o UDP)

Funcionamiento

- Se vincula con un grupo de máquinas
- Comprobación de estado de máquinas asociadas
- Por defecto, round robin entre máquinas disponibles

Coste similar a instancia EC2

- **Pero no se apaga** cuando cerramos el laboratorio

1. Crear AMI de server desacop

Primero crearemos una instancia idéntica a la que ya tenemos

Para ello crearemos una imagen (AMI):

- Selecciona la instancia → Acciones → Imagen y plantillas → Crear imagen...
 - Nombre: Imagen_desacop
- Esperar a que se complete (unos minutos)

The screenshot shows the AWS Management Console interface for the 'Imágenes de Amazon Machine Image (AMI)' section. The left sidebar contains navigation links: Panel de EC2, Vista global de EC2, Eventos, Console-to-Code, Instancias, Tipos de instancia, Plantillas de lanzamiento, Solicitudes de spot, Savings Plans, Instancias reservadas, Alojamiento dedicado, Reservas de capacidad, Imágenes, and Catálogo de AMI. The 'Imágenes' section is selected, and the 'AMI' link is highlighted. The main content area displays the details for the AMI 'Imagen_desacop' with ID 'ami-0f5914ffc4e811648'. The status is 'Disponible' (Available). The details table shows the following information:

ID de la AMI: ami-0f5914ffc4e811648			
Detalles			
ID de AMI	Tipo de imagen	Detalles de la plataforma	Tipo de almacenamiento
ami-0f5914ffc4e811648	machine	Linux/UNIX	EBS
Nombre de AMI	ID de cuenta del propietario	Arquitectura	Operación
Imagen_desacop	051049719530	x86_64	RunInstances
Nombre del dispositivo raíz	Estado	Origen	Tipo de instancia
/dev/sda1	Disponible	051049719530/Imagen_desacop	hvm

2. Lanzar otro servidor desde AMI

- Selecciona Lanzar instancia
 - Nombre: Server_desacoplado_2
 - Mis AMI: **Imagen_desacop**
 - Tipo: t2.micro
 - Claves: vockey
 - Configuración de red:
 - VPC_alta_disp
 - Subred: Public 1b
 - Activar IP pública
 - Grupo seguridad: usar existente SSH, HTTP y MySQL/Aurora

Diferenciar ambos servidores

Conectamos con `server_2` y cambiamos la página para saber que es el segundo servidor:

- `/var/www/html/index.php` *Línea 4*
 - Cambiar `<h1>Welcome page</h1>`
por
`<h1>Welcome page Server2</h1>`

Probar

Welcome page

NAME

ADDRESS

Welcome page Server2

NAME

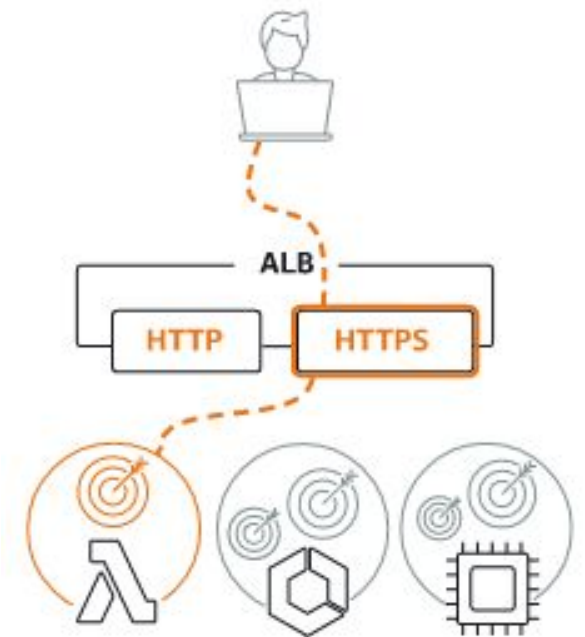
ADDRESS

3. Crear Balanceador de Carga

Crear un Balanceador de carga de aplicaciones

- Nombre: Load-balancer-1
- Expuesto a internet
- VPC_alta_disp
- Selecciona todas las zonas de disponibilidad
- Grupo de seguridad:
 - Crear un nuevo que solo acepte entradas HTTP desde anywhere y asignar
 - Eliminar el resto de grupos
- Agentes de escucha → Acción predeterminada
 - Crear grupo de destino → Instancias y nombre
 - Comprobaciones de estado → Avanzado: bajar umbrales a mínimos para que el balanceador reaccione más rápido

Balanceador de carga de aplicaciones [Info](#)

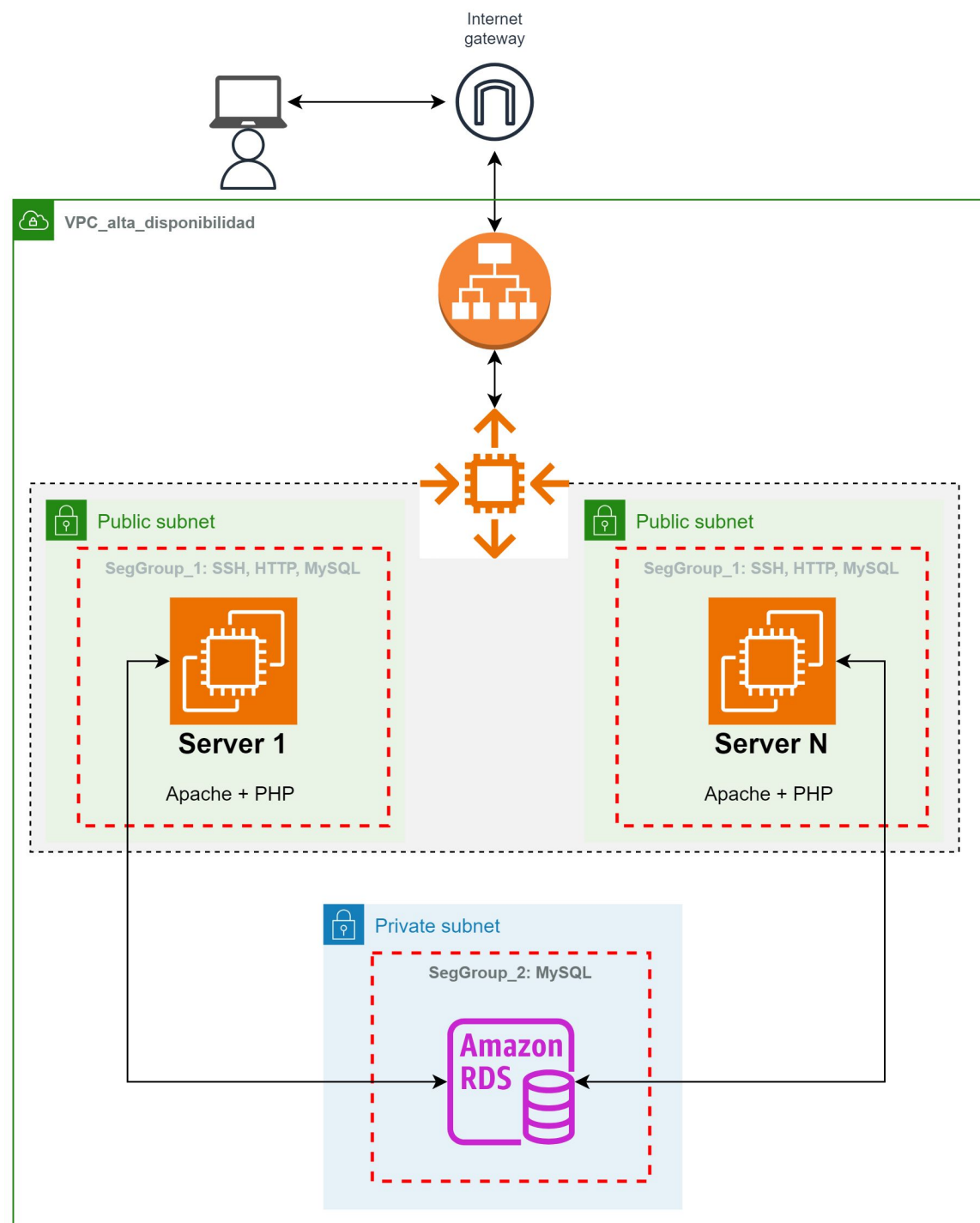


4. Crear grupo de destino y probar

- Crear grupo de destino
 - Seleccionar Instancias
 - Dar nombre
 - Comprobaciones de estado → Avanzado
 - Bajar umbrales a mínimos para que el balanceador reaccione más rápido
 - Seleccionar las instancias Server_desacoplado 1 y 2
 - Incluir como pendientes
- Volver a creación de Balanceador y asignar el grupo
- Una vez Activo, probar usando su Nombre DNS.
 - Probar a apagar un server

Alta Disponibilidad y Escalabilidad automática

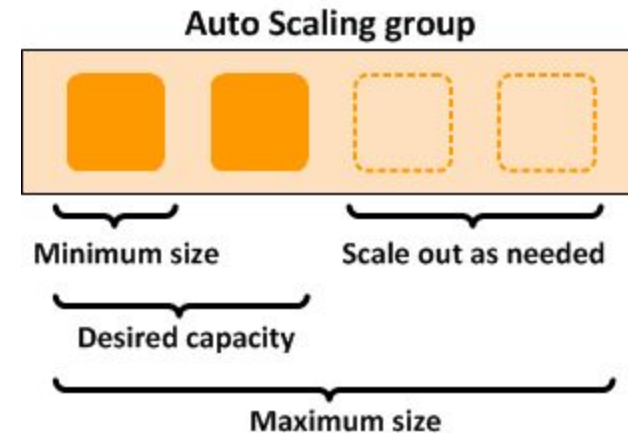
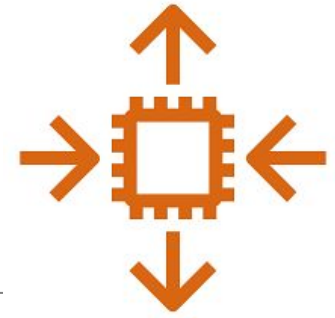
El tráfico se distribuye entre un número de máquinas que se ajusta a la demanda



Auto Scaling groups

Crea un grupo de instancias cuyo número crece o decrece para adecuarse a la carga de la aplicación.

- Se define un mínimo de instancias y un máximo
- Las instancias se crean en base a una plantilla de lanzamiento
- El escalado puede hacerse
 - Manualmente
 - Programando hora y fecha en la que aumentar o reducir
 - Basado en la demanda
 - Predictivo: usando patrones diarios y semanales
- No tiene coste adicional, sólo se paga por los recursos que se empleen (instancias EC2, volúmenes EBS)



1. Crear plantilla de lanzamiento

- Selecciona Server_desacoplado_1 → Acciones → Imagen y plantillas → Crear plantilla...
 - Nombre: Plantilla_server_desacoplado
 - Proporcionar orientación para AutoScaling
 - Mis AMI: Imagen_desacop
 - Tipo: t2.micro
 - Claves: vockey
 - Subred: No incluir en la plantilla de lanzamiento
 - Grupo de seguridad: existente (SSH, HTTP, MYSQL)
 - Configuración avanzada de red → Interfaz de red 1 → Asignar automáticamente la IP pública → Habilitar
 - Etiquetas → Cambiar Name: Server Autoscaling
 - Detalles avanzados → Comportamiento de cierre e hibernación: No incluir

0. Eliminar el Load Balancer anterior

Balanceadores de carga → Seleccionar → Acciones → Eliminar

Crear Grupo Auto Scaling (1)

- Selecciona Grupos de Auto Scaling-> Crear grupo de Auto Scaling
 - Nombre: ASG-server-desac
 - Plantilla lanzamiento: Plantilla_server_desacoplado
 - VPC: VPC_alta_disp
 - Subred: seleccionar todas las públicas (public-1a y public-1b)
 - Asociar a un nuevo balanceador de carga
 - Application Load Balancer
 - Nombre: LB-Autoscaling
 - Internet-facing
 - Agentes de escucha y direccionamiento → Direccionamiento predeterminado→Grupo destino nuevo
 - Comprobaciones de estado: Activar las comprobaciones de estado de Elastic Load Balancing
 - Configuración adicional: Habilitar la recopilación de métricas de grupo en CloudWatch

Crear Grupo Auto Scaling (2)

- Configurar escalamiento y tamaño de grupo
 - Capacidad deseada: 1
 - Capacidad mínima: 1
 - Capacidad máxima: 4
 - Política de escalado de seguimiento de destino
 - Utilización promedio de la CPU
 - Valor de destino: 50
 - Preparación de la instancia: 100 segundos
 - Etiquetas→ Name - Instancia AutoScaling
 - Marcar Etiquetar Instancias nuevas
 - Pulsar Crear grupo de Autoscaling

Probando

Ir a Instancias y comprobar que se ha creado una nueva

Usar IP pública para probar la página

Ir a Balanceadores de carga

- Comprueba que su grupo de seguridad permite tráfico HTTP de entrada desde Anywhere. Si no, añade la regla
- Esperar a que esté disponible: copiar nombre DNS y probar la página

Prueba de estrés

- Instala stress en la primera instancia
 - `sudo apt install stress`
 - `stress --cpu 50 -t 500 &`
 - Lanza 50 procesos que realizan cálculos durante 500 segundos
 - `top`
 - Comprueba que la CPU está al 100%
- Tras unos minutos, comprueba que el grupo autoescalado crea nuevas instancias
 - Registro de actividad de Grupo AutoScaling
- Unos 10-15 minutos después de acabado el test de estrés, se eliminarán las instancias no utilizadas