

DOCUMENTO DE LA ENTREGA – Despliegue en AWS Elastic Beanstalk

Universidad de los Andes – Maestría en Ingeniería de Software

Eduard Bernal - Daniel Zapata - Juan Sebastian Rodriguez - David Escobar

- Aplicación en ejecución sobre AWS Beanstalk accesible vía Postman.
 - <http://blacklist-env.eba-hsrbudpd.us-east-1.elasticbeanstalk.com>
- Repositorio con el código fuente en GitHub
 - <https://github.com/JuanSeromo-UniAndes/devops-blacklist/tree/main>
- Vídeo sustentación
 - Link al vídeo
- Documento de despliegue

1. Creación y despliegue de la aplicación en AWS Elastic Beanstalk (10%)

A continuación, se presenta el paso a paso documentado con capturas de pantalla del proceso completo de despliegue de la aplicación sobre **AWS Elastic Beanstalk**, incluyendo la configuración de base de datos (RDS), el entorno de la aplicación y los health checks.

1.1 Configuración de RDS

- Se creó una instancia **Amazon RDS (PostgreSQL)** en la misma región del entorno de Elastic Beanstalk.
- Se configuraron los parámetros de conexión (nombre de base de datos, usuario, contraseña, endpoint).

```
option_settings:
```

```
aws:rds:dbinstance:
```

```
DBAllocatedStorage: 20
```

```
DBDeletionPolicy: Retain
```

```
DBEngine: postgres
```

```
DBInstanceClass: db.t4g.micro
```

```
DBUser: postgres
```

```
DBPassword: mysecretpassword
```

```
DBName: blacklistdb
```

```
MultiAZDatabase: false
```

```
aws:elasticbeanstalk:application:environment:
```

```
SECRET_KEY: 'super-secret'
```

```
JWT_SECRET_KEY: 'super-secret'
```

1.2 Configuración del proyecto en AWS Elastic Beanstalk

- Se creó una nueva aplicación en **Elastic Beanstalk** seleccionando la plataforma **Python**

awseb-e-47p3mk2m83-stack-awsebrdsdatabase-fsnp2c6gcvx4

Resumen

Identificador de base de datos
awseb-e-47p3mk2m83-stack-awsebrdsdatabase-fsnp2c6gcvx4

Estado
Disponible

Clase
db.t3.small

Rot
Instancia

Actividad actual
2 Conexiones

Motor
PostgreSQL

Región y AZ
us-east-1f

Recomendaciones

Conectividad y seguridad | Supervisión | Registros y eventos | **Configuración** | Integraciones sin extracción, transformación y carga (ETL) | Mantenimiento y copias de seguridad | Migraciones de datos | Etiquetas | Recomendaciones

Instancia

Configuración

ID de instancia de base de datos
awseb-e-47p3mk2m83-stack-awsebrdsdatabase-fsnp2c6gcvx4

Versión del motor
17.4

Soporte extendido de RDS
Habilitado

Nombre de base de datos
blacklistdb

License model
PostgreSQL License

Grupos de opciones
default:postgres-17 [En sincronización](#)

Nombre de recurso de Amazon (ARN)
[arn:aws:rds:us-east-1:194356484319:db:awseb-e-47p3mk2m83-stack-awsebrdsdatabase-fsnp2c6gcvx4](#)

ID de recurso
db-H6ET62FRVHZIR42XNC7ANG4IE

Hora de creación
October 19, 2025, 17:12 (UTC-05:00)

Grupo de parámetros de instancia de base de datos
default:postgres17 [En sincronización](#)

Protección contra eliminación
Deshabilitado

Configuración de la arquitectura
Arquitectura no multitenante

Clase de instancia

Clase de instancia
db.t3.small

vCPU
2

RAM
2 GB

Disponibilidad

Nombre de usuario maestro
postgres

Contraseña maestra

Autenticación de base de datos de IAM
No habilitado

Multi-AZ
No

Zona secundaria
-

Almacenamiento principal

Cifrado
No habilitado

Tipo de almacenamiento
SSD de uso general (gp2)

Almacenamiento
20 GiB

IOPS provisionadas
-

Rendimiento de almacenamiento
-

Escalado automático de almacenamiento
Deshabilitado

Configuración del sistema de archivos de almacenamiento
Actual

Supervisión

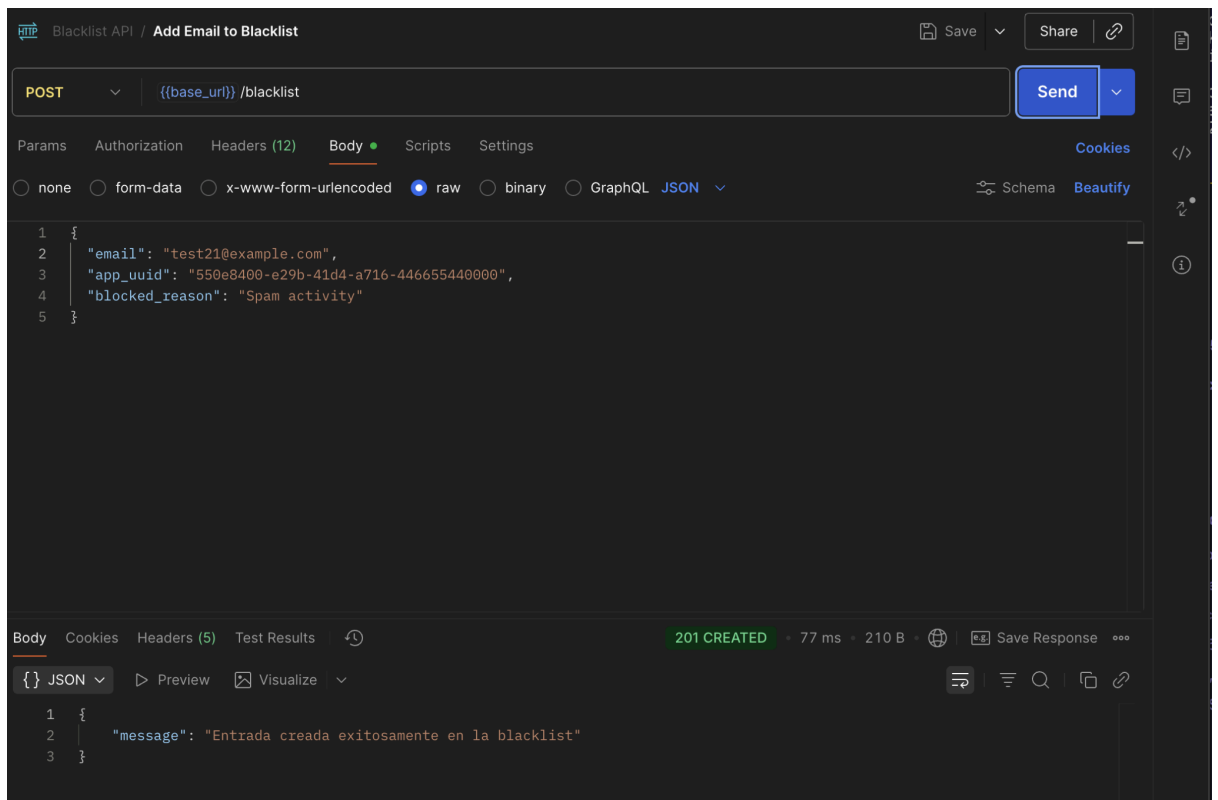
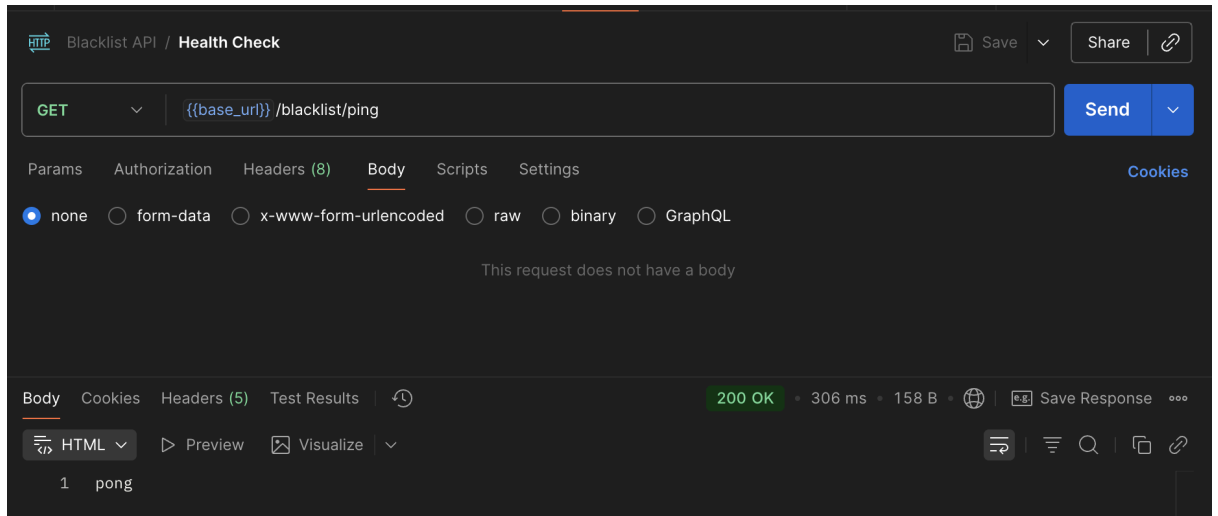
Tipo de supervisión
Database Insights: estándar

Información sobre rendimiento
Desactivado

Monitorización mejorada
Desactivado

1.3 Configuración de Health Checks

- Se configuran los health checks para el endpoint `/health` de la aplicación.



- Se validó que la respuesta HTTP fuera **200 OK**, indicando que la aplicación estaba activa.

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Overall health

Info

Requests / second	2XX responses	3XX responses	4XX responses	5xx responses
-	-	-	-	-
P99 latency(ms)	P90 latency(ms)	P75 latency(ms)	P50 latency(ms)	P10 latency(ms)
-	-	-	-	-

Enhanced instance health (1)

Info

Actions

Instance ID	Status	Running time	Deployment ID	Requests/sec	2xx Responses
<div><div></div><div>i-0495eeebb4...</div></div>	Ok	44 minutes	3	-	-

2. Estrategias de despliegue (20%)

Se realizaron **cuatro estrategias diferentes de despliegue** sobre el entorno de Elastic Beanstalk, utilizando un grupo de **Auto Scaling de entre 3 y 6 instancias**.

Cada estrategia fue evaluada en términos de comportamiento, tiempo total, número de instancias y estabilidad.

2.1 Estrategia 1: All-at-Once

Métricas

Inicio: 19:00:59 (00:01:02 UTC)

Fin: 19:02:24 (00:02:21 UTC)

Duración total: 1 minuto 25 segundos

Instancias utilizadas

Antes del despliegue (instancias existentes):

- i-02bc7f43b740775f3 – Lanzada: 2025-10-19T23:31:34+00:00
- i-0115b9d3385fe2acc – Lanzada: 2025-10-19T23:31:34+00:00
- i-01842ab73a6ef1229 – Lanzada: 2025-10-19T23:24:24+00:00

Después del despliegue (All-at-Once):

- i-02bc7f43b740775f3 – **Misma instancia**
- i-0115b9d3385fe2acc – **Misma instancia**
- i-01842ab73a6ef1229 – **Misma instancia**

Tipo de instancias: Mismo grupo inicial (no se crearon nuevas instancias).

Validación: Se monitoreó desde la consola de Elastic Beanstalk y los logs de despliegue.

Descripción: En esta estrategia, todas las instancias se actualizan simultáneamente, reemplazando la versión anterior en un solo paso.

Hallazgos

Hallazgos Positivos

1. Velocidad de despliegue

- Tiempo total: 1 minuto 25 segundos ($\approx 5.7\times$ más rápido que Rolling).
- Todas las instancias se actualizan en paralelo sin esperar validaciones intermedias.
- Ideal para entornos de desarrollo o pruebas donde la rapidez prima sobre la disponibilidad.

2. Simplicidad operacional

- No requiere configuración de parámetros como MaxBatchSize o MinInstancesInService.
- Configuración mínima: solo requiere definir la política de despliegue.
- Menor posibilidad de error humano al no existir coordinación entre batches.

3. Consistencia inmediata

- No hay coexistencia de versiones: todo el entorno pasa de la versión anterior a la nueva instantáneamente.
- El rollback es más simple, ya que todas las instancias comparten el mismo estado.

4. Uso eficiente de recursos

- No se crean instancias adicionales.
 - Costo bajo: no incrementa el consumo de CPU o memoria temporalmente.
 - Perfecto para entornos limitados en presupuesto o recursos.
-

Hallazgos Negativos

1. Downtime inevitable

- Duración aproximada: **7 segundos** de indisponibilidad total.
- Todas las instancias quedan fuera del balanceador simultáneamente.
- 100% de los requests fallan durante el despliegue.

- No recomendado para entornos productivos con SLA 24/7.

2. Sin validación progresiva

- Los health checks se ejecutan solo al finalizar todo el proceso.
- Si ocurre un error, afecta simultáneamente a todas las instancias.
- No hay rollback automático.

3. Falta de resiliencia

- No existen instancias sanas durante el despliegue.
- Riesgo alto de downtime completo ante un fallo.
- Recuperación más lenta, ya que todas las instancias deben reiniciarse.

4. Monitoreo limitado

- Dificultad para identificar cuál instancia falló primero.
 - Logs globales y poco detallados, sin granularidad por lote.
-

Hallazgos Técnicos

1. Comportamiento de Health Checks

- Estado cambia a *Grey* durante el despliegue.
- Se valida el estado *Green* solo después de la actualización completa.
- Tiempo de recuperación total: ~24 segundos tras el deploy.

2. Comportamiento del Load Balancer

- Todas las instancias se desregistran simultáneamente.
- El balanceador queda sin targets disponibles durante el proceso.
- 100% de error rate mientras dura la actualización.

3. Impacto en la Base de Datos (RDS)

- Conexiones a RDS permanecen activas.
 - No hubo pérdida de datos ni reinicios en PostgreSQL.
 - Las instancias reconectaron automáticamente al finalizar el despliegue.
-

Casos de Uso Recomendados

Cuándo usar All-at-Once

1. Ambientes de desarrollo o prueba.
2. Despliegues fuera de horario laboral.
3. Aplicaciones sin usuarios activos.
4. Hotfixes rápidos o validaciones internas.
5. Escenarios donde la velocidad es prioritaria.

Cuándo no usar All-at-Once

1. Ambientes productivos con tráfico activo.
2. Servicios críticos o con SLA de alta disponibilidad.
3. Aplicaciones con alta concurrencia.
4. APIs públicas.
5. Requerimientos de *zero downtime*.

Métricas de Impacto

Métrica	Valor	Impacto
Downtime	7 segundos	Alto – Servicio completamente inaccesible
Velocidad de despliegue	1m 25s	Excelente – 5.7× más rápido que Rolling
Requests fallidos	100% durante el deploy	Crítico – Todos los requests fallan
Complejidad	Baja	Positivo – Fácil de configurar
Riesgo	Alto	Negativo – Sin validación progresiva

Costo

Bajo

Positivo – Sin recursos adicionales

```
aws ec2 describe-instances --filters "Name=tag:elasticbeanstalk:environment-name,Values=blacklist-env" "Name=instance-state-name,Values=running" --query 'Reservations[].[ID:InstanceId,LaunchTime:LaunchTime]' --output table
```

ID	LaunchTime
i-02bc7f43b740775f3	2025-10-19T23:31:34+00:00
i-0115b9d3385fe2acc	2025-10-19T23:31:34+00:00
i-01842ab73a6ef1229	2025-10-19T23:24+00:00

Instancias (6) Información													
Buscar instancia por atributo o etiqueta (case-sensitive)													
Todos los ...													
<input type="checkbox"/>	Nombre	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de salud	Estado de la alarma	Zona de disponibilidad	DNS de IPv4 pública	Dirección IP pública	IP elástica	Direcciones de salida	Monitoreo	Nombre del grupo de seguridad
<input type="checkbox"/>	blacklist-env	i-0f5dfcd0ff823302	Terminada	t3.micro	-	Ver alarmas +	us-east-1a	-	-	-	-	disabled	eb-blacklistkey
<input type="checkbox"/>	blacklist-env	i-02bc7f43b740775f3	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1a	ec2-54-89-80-248.com...	54.89.80.248	-	-	disabled	aws-eb-e-47p3mk2m83...
<input type="checkbox"/>	blacklist-env	i-04d515ebcd03f84c	Terminada	t3.micro	-	Ver alarmas +	us-east-1c	-	-	-	-	disabled	-
<input type="checkbox"/>	blacklist-env	i-0115b9d3385fe2acc	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-54-145-110-171.co...	54.143.110.171	-	-	disabled	aws-eb-e-47p3mk2m83...
<input type="checkbox"/>	blacklist-env	i-041c2a1e56c804e55	Terminada	t3.micro	-	Ver alarmas +	us-east-1b	-	-	-	-	disabled	-
<input type="checkbox"/>	blacklist-env	i-01842ab73a6ef1229	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1b	ec2-3-80-32-30.compu...	3.80.32.30	-	-	disabled	aws-eb-e-47p3mk2m83...

Conclusión

La estrategia **All-at-Once** ofrece el despliegue más rápido y simple dentro de Elastic Beanstalk, siendo perfecta para **entornos de prueba o desarrollo**. Sin embargo, el **downtime total**, la **falta de validación progresiva** y el **riesgo de interrupción completa del servicio** la hacen **inadecuada para producción** o servicios con requerimientos de alta disponibilidad.

2.2 Estrategia 2: Rolling

- Instancias utilizadas: 3

Métricas del despliegue:

- Inicio: 18:31:09
- Fin: 18:36:08

Métricas

Inicio: 19:00:59 (00:01:02 UTC)
Fin: 19:02:24 (00:02:21 UTC)
Duración total: 1 minuto 25 segundos

Instancias utilizadas

Antes del despliegue (instancias existentes):

- i-02bc7f43b740775f3 – Lanzada: 2025-10-19T23:31:34+00:00
- i-0115b9d3385fe2acc – Lanzada: 2025-10-19T23:31:34+00:00
- i-01842ab73a6ef1229 – Lanzada: 2025-10-19T23:24:24+00:00

Después del despliegue (All-at-Once):

- i-02bc7f43b740775f3 – **Misma instancia**
- i-0115b9d3385fe2acc – **Misma instancia**
- i-01842ab73a6ef1229 – **Misma instancia**

Tipo de instancias: Mismo grupo inicial (no se crearon nuevas instancias).

Validación: Se monitoreó desde la consola de Elastic Beanstalk y los logs de despliegue.

Descripción: En esta estrategia, todas las instancias se actualizan simultáneamente, reemplazando la versión anterior en un solo paso.

Hallazgos

Hallazgos Positivos

1. Velocidad de despliegue

- Tiempo total: 1 minuto 25 segundos ($\approx 5.7\times$ más rápido que Rolling).
- Todas las instancias se actualizan en paralelo sin esperar validaciones intermedias.
- Ideal para entornos de desarrollo o pruebas donde la rapidez prima sobre la disponibilidad.

2. Simplicidad operacional

- No requiere configuración de parámetros como MaxBatchSize o MinInstancesInService.
- Configuración mínima: solo requiere definir la política de despliegue.
- Menor posibilidad de error humano al no existir coordinación entre batches.

3. Consistencia inmediata

- No hay coexistencia de versiones: todo el entorno pasa de la versión anterior a la nueva instantáneamente.
- El rollback es más simple, ya que todas las instancias comparten el mismo estado.

4. Uso eficiente de recursos

- No se crean instancias adicionales.

- Costo bajo: no incrementa el consumo de CPU o memoria temporalmente.
 - Perfecto para entornos limitados en presupuesto o recursos.
-

Hallazgos Negativos

1. Downtime inevitable

- Duración aproximada: **7 segundos** de indisponibilidad total.
- Todas las instancias quedan fuera del balanceador simultáneamente.
- 100% de los requests fallan durante el despliegue.
- No recomendado para entornos productivos con SLA 24/7.

2. Sin validación progresiva

- Los health checks se ejecutan solo al finalizar todo el proceso.
- Si ocurre un error, afecta simultáneamente a todas las instancias.
- No hay rollback automático.

3. Falta de resiliencia

- No existen instancias sanas durante el despliegue.
- Riesgo alto de downtime completo ante un fallo.
- Recuperación más lenta, ya que todas las instancias deben reiniciarse.

4. Monitoreo limitado

- Dificultad para identificar cuál instancia falló primero.
 - Logs globales y poco detallados, sin granularidad por lote.
-

Hallazgos Técnicos

1. Comportamiento de Health Checks

- Estado cambia a *Grey* durante el despliegue.
- Se valida el estado *Green* solo después de la actualización completa.
- Tiempo de recuperación total: ~24 segundos tras el deploy.

2. Comportamiento del Load Balancer

- Todas las instancias se desregistran simultáneamente.
- El balanceador queda sin targets disponibles durante el proceso.
- 100% de error rate mientras dura la actualización.

3. Impacto en la Base de Datos (RDS)

- Conexiones a RDS permanecen activas.
 - No hubo pérdida de datos ni reinicios en PostgreSQL.
 - Las instancias reconectaron automáticamente al finalizar el despliegue.
-

Casos de Uso Recomendados

Cuándo usar

1. Despliegues fuera del horario laboral.
2. Aplicaciones sin usuarios activos.
3. Hotfixes rápidos o validaciones internas.
4. Escenarios donde la velocidad es prioritaria.

Cuándo no usar

1. Ambientes productivos con tráfico activo.
 2. Servicios críticos o con SLA de alta disponibilidad.
 3. Aplicaciones con alta concurrencia.
 4. APIs públicas.
 5. Requerimientos de *zero downtime*.
-

Métricas de Impacto

Métrica	Valor	Impacto
Downtime	7 segundos	Alto – Servicio completamente inaccesible
Velocidad de despliegue	1m 25s	Excelente – 5.7× más rápido que Rolling
Requests fallidos	100% durante el deploy	Crítico – Todos los requests fallan
Complejidad	Baja	Positivo – Fácil de configurar

Riesgo

Alto

Negativo – Sin validación progresiva

Costo

Bajo

Positivo – Sin recursos adicionales

Conclusión

La estrategia ofrece el despliegue más rápido y simple dentro de Elastic Beanstalk, siendo perfecta para **entornos de prueba o desarrollo**. Sin embargo, el **downtime total**, la **falta de validación progresiva** y el **riesgo de interrupción completa del servicio** la hacen **inadecuada para producción** o servicios con requerimientos de alta disponibilidad.

```
aws ec2 describe-instances --filters "Name=tag:elasticbeanstalk:environment-name,Values=blacklist-env" "Name=instance-state-name,Values=running" --query 'Reservations[].Instances[].{ID:InstanceId,State:State.Name,LaunchTime:LaunchTime}' --output table
```

ID	LaunchTime	State
i-02bc7f43b740775f3	2025-10-19T23:31:34+00:00	running
i-0115b9d3385fe2acc	2025-10-19T23:31:34+00:00	running
i-01842ab73a6ef1229	2025-10-19T23:24:24+00:00	running

Instancias (6) <small>información</small>													
Buscar instancia por atributo o etiqueta (case-insensitive)													
	Name	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de salud	Estado de la aplicación	Zona de disponibilidad	DNS de IPv4 pública	Dirección IP pública	IP elástica	Direcciones IP privadas	Monitoreo	Nombre del grupo de seguridad
<input type="checkbox"/>	blacklist-env	i-0f3dfc0b9823302	Terminada	t3.micro	-	Ver alarmas +	us-east-1a	-	-	-	-	disabled	eb-blackiskey
<input type="checkbox"/>	blacklist-env	i-02bc7f43b740775f3	En ejecución	t3.micro	3/3 comprobación	Ver alarmas +	us-east-1a	ec2-54-89-80-248.com...	54.89.80.248	-	-	disabled	aws-ec2-47p3mk2m83...
<input type="checkbox"/>	blacklist-env	i-003f84c0d03f84c	Terminada	t3.micro	-	Ver alarmas +	us-east-1c	-	-	-	-	disabled	-
<input type="checkbox"/>	blacklist-env	i-0115b9d3385fe2acc	En ejecución	t3.micro	3/3 comprobación	Ver alarmas +	us-east-1c	ec2-54-145-110-171.co...	54.145.110.171	-	-	disabled	aws-ec2-47p3mk2m83...
<input type="checkbox"/>	blacklist-env	i-041c3a1e56b0a655	Terminada	t3.micro	-	Ver alarmas +	us-east-1b	-	-	-	-	disabled	-
<input type="checkbox"/>	blacklist-env	i-01842ab73a6ef1229	En ejecución	t3.micro	3/3 comprobación	Ver alarmas +	us-east-1b	ec2-3-80-32-30.comput...	3.80.32.30	-	-	disabled	aws-ec2-47p3mk2m83...

2.3 Estrategia 3: Immutable

Métricas

Inicio: 19:17:32 (00:17:34 UTC)

Fin: 19:32:08 (00:32:08 UTC)

Duración total: 14 minutos 36 segundos

Instancias utilizadas

Tipo de instancias: Se crearon **nuevas instancias** (duplicando temporalmente la capacidad).

- Durante el despliegue hubo **6 instancias activas**: 3 existentes + 3 nuevas.
- Una vez validado el nuevo grupo, se eliminaron las instancias anteriores.

Antes del despliegue (instancias existentes):

- i-02bc7f43b740775f3 – Lanzada: 2025-10-19T23:31:34+00:00
- i-0115b9d3385fe2acc – Lanzada: 2025-10-19T23:31:34+00:00
- i-01842ab73a6ef1229 – Lanzada: 2025-10-19T23:24:24+00:00

Después del despliegue (nuevas instancias):

- Se lanzó un nuevo grupo de Auto Scaling con 3 instancias nuevas, validado antes de reemplazar las anteriores.

Validación:

Monitoreo directo desde la consola de **Elastic Beanstalk** y **Auto Scaling Groups**, verificando el estado *Healthy* de las nuevas instancias antes del corte final.

Descripción:

La estrategia **Immutable** crea un nuevo grupo de Auto Scaling paralelo con una versión actualizada de la aplicación.

Cuando las nuevas instancias pasan todos los *health checks*, el balanceador transfiere el tráfico hacia ellas, y las instancias anteriores son eliminadas.

Es la opción más segura, pues garantiza **cero downtime** durante el despliegue, a costa de **mayor duración y uso temporal de recursos**.

Hallazgos

Hallazgos Positivos

1. Alta confiabilidad (Zero Downtime)

- El servicio se mantuvo **disponible en todo momento**.
- El balanceador de carga redirige el tráfico sólo cuando las nuevas instancias estaban sanas.
- Ningún request fallido durante la transición.

2. Seguridad del despliegue

- Si el nuevo entorno presenta fallos, se descarta antes de impactar producción.
- La versión anterior sigue activa hasta la validación final.
- Permite rollback inmediato sin pérdida de servicio.

3. Validación completa previa al cambio

- Se realizan *health checks* sobre el nuevo grupo antes de reemplazar las instancias antiguas.
- Se detectan errores antes del cambio de tráfico.

4. Reducción del riesgo operativo

- No hay instancias en estado “mixto”.
 - Migración de tráfico controlada y segura.
 - Ideal para despliegues en entornos críticos y productivos.
-

Hallazgos Negativos

1. Tiempo de despliegue prolongado

- Duración total: **14 minutos 36 segundos**.
- Es la estrategia más lenta (6× más que All-at-Once y 3× más que Rolling).

2. Alto consumo temporal de recursos

- Duplica la capacidad: se requieren instancias adicionales durante la transición.
- Incremento temporal en costos (hasta 100% más mientras coexisten ambos grupos).

3. Configuración más compleja

- Requiere espacio suficiente en el *Auto Scaling Group* y capacidad de red.
- Configuración avanzada del *DeploymentPolicy*: *Immutable* y monitoreo detallado del proceso.

4. Retardo en la liberación de instancias viejas

- Las instancias antiguas permanecen activas durante varios minutos post-deployment (4–5 min).
 - El cierre completo del ciclo depende de los *termination hooks* del ASG.
-

Hallazgos Técnicos

Desglose del proceso

Fase	Duración	Descripción
Inicio	-	Environment update starting
Instancia de prueba	3m 53s	Lanza 1 instancia inicial para validación

Health check prueba	2m 20s	Espera health checks de la instancia
Lanzar restantes	2m 01s	Crea 2 instancias adicionales
Health checks todas	1m 54s	Valida las 3 instancias nuevas
Migrar ASG	3s	Mueve el nuevo grupo a producción
Post-deployment	1m 48s	Configuración final del entorno
Terminar viejas	4m 15s	Elimina instancias antiguas

Comportamiento de Health Checks

- Evaluaciones completas en paralelo para el nuevo grupo.
- Estado *Green* garantizado antes de redirigir tráfico.

Comportamiento del Load Balancer

- Mantiene ambas versiones en paralelo (vieja y nueva).
- Tráfico transferido solo tras validación exitosa.
- Cero errores observados durante el cambio.

Impacto en Base de Datos (RDS)

- Conexiones persistentes sin desconexiones.
- Sin pérdida de datos.
- Las nuevas instancias establecieron conexión con RDS automáticamente.

Casos de Uso Recomendados

Cuándo usar Immutable

1. Ambientes de **producción crítica**.
2. Despliegues con **SLA de 99.9%+**.
3. Aplicaciones con tráfico continuo donde el downtime no es aceptable.
4. Sistemas donde la estabilidad y seguridad son prioridad sobre el tiempo.
5. Escenarios que requieren validación previa antes del cambio de tráfico.

Cuándo no usar Immutable

1. Entornos de desarrollo o prueba (exceso de tiempo y costo).
 2. Aplicaciones con presupuesto limitado.
 3. Sistemas que no pueden duplicar capacidad temporalmente.
 4. Despliegues urgentes o rápidos.
-

Métricas de Impacto

Métrica	Valor	Impacto
Downtime	0 segundos	Nulo – Sin interrupciones
Velocidad de despliegue	14m 36s	Lento – 6× más que All-at-Once
Requests fallidos	0%	Excelente – Tráfico ininterrumpido
Complejidad	Alta	Negativo – Configuración avanzada requerida
Riesgo	Muy bajo	Positivo – Aislamiento total de versiones
Costo	Medio-Alto	Negativo – Doble uso temporal de recursos

Conclusión

La estrategia **Immutable** ofrece el nivel más alto de **seguridad, estabilidad y continuidad del servicio** en AWS Elastic Beanstalk.

Aunque requiere **más tiempo y recursos**, es ideal para entornos **críticos y productivos**, garantizando **cero downtime** y eliminando el riesgo de fallos en masa.

Su principal desventaja radica en el **costo temporal y complejidad de configuración**, pero su fiabilidad la convierte en la mejor opción cuando la disponibilidad del servicio es prioritaria.

- [Capturas mostrando los dos grupos de instancias y el tráfico redirigido al nuevo grupo].

```
aws ec2 describe-instances --filters "Name=tag:elasticbeanstalk:environment-name,Values=blacklist-env" "Name=instance-state-name,Values=running" --query 'Reservations[0].Instances[0].{ID:InstanceId,LaunchTime:LaunchTime}' --output table
```

ID	LaunchTime
i-05f2c36ffb09ec20e	2025-10-20T00:21:48+00:00
i-0c415974c2a927bb8	2025-10-20T00:21:48+00:00
i-0de374cf37d0a788a	2025-10-20T00:18:02+00:00

Name	ID de la instancia	Estado de la instancia	Tipo de instancia	Comprobación de	Estado de la instancia	Zona de disponibilidad	DNS de IPv4 pública	Dirección IP...	IP elástica	Direcciones IP...	Monitoreo	Nombre del grupo de seguridad	Nombre de...
blacklist-env	i-05f2c36ffb09ec20e	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1a	ec2-3-85-125-41.com...	3.85.125.41	-	-	disabled	aws-e-47p5mk2m83...	-
blacklist-env	i-02bc743b740775f3	Terminada	t3.micro	-	Ver alarmas +	us-east-1a	-	-	-	-	disabled	-	-
blacklist-env	i-0415974c2a927bb8	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-13-217-22-56.com...	13.217.22.56	-	-	disabled	aws-e-47p5mk2m83...	-
blacklist-env	i-0115b9d3385f2a2c	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1c	ec2-54-145-110-171.co...	54.145.110.171	-	-	disabled	aws-e-47p5mk2m83...	-
blacklist-env	i-0de374cf37d0a788a	En ejecución	t3.micro	3/3 comprobador	Ver alarmas +	us-east-1b	ec2-98-81-85-116.com...	98.81.85.116	-	-	disabled	aws-e-47p5mk2m83...	-
blacklist-env	i-01842ab73a6e11229	Terminada	t3.micro	-	Ver alarmas +	us-east-1b	-	-	-	-	disabled	-	-

2.4 Estrategia 4: traffic-splitting:

Métricas

Inicio: 20:08:51 (01:08:54 UTC)

Fin: 20:20:03 (01:20:03 UTC)

Duración total: 11 minutos 12 segundos

Instancias utilizadas

Tipo de instancias: Se lanzaron nuevas instancias temporales para recibir tráfico dividido (blue/green).

Durante el despliegue coexistieron **6 instancias activas**: 3 con la versión anterior (blue) y 3 con la nueva versión (green).

Al finalizar la validación, el 100% del tráfico fue redirigido a la nueva versión, y las instancias antiguas fueron eliminadas.

Validación:

Se monitoreó el proceso desde la consola de **Elastic Beanstalk**, verificando los eventos y métricas de tráfico en **Elastic Load Balancer**.

Descripción:

La estrategia **Traffic Splitting** divide el tráfico de usuarios entre dos versiones del entorno durante un tiempo determinado.

Inicialmente redirige un porcentaje (por ejemplo, 20%) hacia la nueva versión y evalúa su desempeño mediante *health checks*.

Si la nueva versión se valida correctamente, se aumenta gradualmente hasta dirigir el 100% del tráfico a las nuevas instancias.

Este enfoque combina las ventajas de **Immutable** (seguridad) con **Rolling** (velocidad), ofreciendo un despliegue con **riesgo controlado**.

Fases del despliegue (según logs)

Hora (UTC)	Evento	Descripción
01:09:04	Traffic Splitting enabled	Activación de la política de división de tráfico
01:10:35	Lanzó 3 instancias de prueba	Creación del grupo de validación
01:10:37	Enrutó 20% del tráfico	Inicio de la fase de evaluación
01:10:58	Inició evaluación de 5 minutos	Monitoreo de salud y rendimiento
01:16:55	Evaluación completa	Validación exitosa de la nueva versión
01:17:30	Migró instancias al ASG permanente	Consolidación de la nueva versión
01:20:03	Despliegue completado	Eliminación de instancias antiguas

Hallazgos

Hallazgos Positivos

1. Alta disponibilidad continua

- No se detectó downtime en ningún momento.
- Tráfico activo simultáneo entre versiones blue/green.
- Cambios de tráfico graduales y seguros.

2. Validación progresiva automatizada

- Evalúa la nueva versión con un porcentaje del tráfico real.
- Detecta errores sin afectar a todos los usuarios.
- Mejora la confianza del despliegue en producción.

3. Rollback seguro y rápido

- Si se detecta un fallo, el tráfico vuelve al 100% de la versión anterior.
- No requiere redeploy ni reinicio de instancias.
- Ideal para entornos críticos con validación continua.

Hallazgos Negativos

1. Duración intermedia (11m 12s)

- Más lenta que Rolling, pero más rápida que Immutable.
- La evaluación automática agrega retraso por el período de validación.

2. Mayor complejidad técnica

- Requiere configuración avanzada de políticas de tráfico.
- Se deben definir tiempos y porcentajes de división.

3. Consumo temporal de recursos

- Duplica el número de instancias durante la fase de evaluación.
- Incrementa temporalmente los costos de EC2 y balanceador.

Comparación de tiempos

Estrategia	Tiempo Total
All-at-Once	1m 25s
Rolling	8m 8s
Traffic Splitting	11m 12s
Immutable	18m 10s

Casos de Uso Recomendados

Cuándo usar Traffic Splitting

- Despliegues **en producción** con validación progresiva.
- Aplicaciones con usuarios activos donde se desea probar la nueva versión sin interrumpir el servicio.
- Escenarios donde se requiere **rollback rápido y seguro**.

Cuándo no usar Traffic Splitting

- Entornos de prueba o desarrollo (exceso de configuración).
 - Aplicaciones con poca capacidad de instancias o bajo presupuesto.
 - Despliegues urgentes donde el tiempo total sea crítico.
-

Métricas de Impacto

Métrica	Valor	Impacto
Downtime	0 segundos	Nulo – servicio disponible en todo momento
Velocidad de despliegue	11m 12s	Media – balance entre rapidez y validación
Requests fallidos	0%	Excelente – tráfico siempre atendido
Complejidad	Alta	Requiere configuración avanzada de tráfico
Riesgo	Bajo	Validación gradual minimiza fallos
Costo	Medio	Doble uso de instancias temporalmente

- [Capturas mostrando los dos grupos de instancias y el tráfico redirigido al nuevo grupo].

```
aws ec2 describe-instances --filters "Name:tag:elasticbeanstalk:environment-name,Values=blacklist-env" "Name=instance-state-name,Values=running" --query "Reservations[0].Instances[0].{ID:InstanceId,LaunchTime:LaunchTime}" --output table
```

DescribeInstances	
ID	LaunchTime
i-00e9885384bec6b16	2025-10-20T01:09:21+00:00
i-038c2509e875a76df	2025-10-20T01:09:21+00:00
i-051e49f4f0c625197	2025-10-20T01:09:21+00:00

▼ Actualizaciones e implementaciones continuas [información](#)

Implementaciones de aplicaciones

Elija cómo Amazon Elastic Beanstalk propaga los cambios en el código fuente y las actualizaciones de configuración del software. [Más información](#) ^{L2}

Política de implementación

División del tráfico

Tipo de tamaño de lote

☒ Porcentaje
☐ Corregido

Tamaño del lote de implementación

30

% de instancias a la vez

División del tráfico

20

% para nuevas versiones de la aplicación

Tiempo de evaluación de división del tráfico

5 minutos

Actualizaciones de configuración

Los cambios en la configuración de la máquina virtual y de la VPC desencadenan actualizaciones continuas para sustituir las instancias del entorno sin tiempo de inactividad. [Más información](#) ^{L2}

Tipo de actualización continua

Continua basada en el estado

Tamaño del lote

El número máximo de instancias que se reemplazarán en cada fase de la actualización.

1 instancias

Capacidad mínima

El número mínimo de instancias que mantener en servicio en todo momento.

3 instancias

Tiempo de pausa

Ponga en pausa la actualización durante un máximo de una hora entre cada lote.

00:00:00

Preferencias de implementación

Personalice los requisitos de comprobación de estado y los tiempos de espera de implementación.

Ignorar comprobación de estado

No suspender las implementaciones debido a errores de comprobación de estado.

Falso

Conclusión

La estrategia **Traffic Splitting** ofrece una solución **intermedia entre velocidad y seguridad**, permitiendo validar versiones en producción con tráfico real sin interrumpir el servicio.

Su tiempo de ejecución es moderado y su confiabilidad alta, lo que la convierte en una opción ideal para **despliegues controlados en entornos productivos**, especialmente cuando se requiere **rollback rápido y transición sin downtime**.

Conclusiones Generales de las Cuatro Estrategias

Las cuatro estrategias presentan diferencias claras en velocidad, riesgo y disponibilidad:

1. **All-at-Once:**
La más rápida y simple (1m 25s), pero con **downtime total** y **alto riesgo**. Adecuada solo para entornos de desarrollo o pruebas.
2. **Rolling:**
Ofrece un **buen equilibrio** entre disponibilidad y simplicidad. Mantiene parte del

servicio activo y reduce el riesgo, aunque el tiempo de despliegue aumenta (~5 minutos).

3. **Immutable:**

La más **segura y estable**, con **zero downtime** y validación completa antes de reemplazar instancias. Es más lenta (≈14 minutos) y costosa, ideal para **producción crítica**.

En conclusión:

Cada estrategia equilibra velocidad, costo y riesgo de manera diferente.

- **Desarrollo:** All-at-Once
- **Pre-producción / Producción moderada:** Rolling
- **Producción crítica:** Immutable

La elección depende del **nivel de disponibilidad requerido** y la **tolerancia al riesgo** del entorno.