

Computació en Entorns Al Núvol

Pràctica 2 - Load Balancer & Blue/Green Deployment

Grup 19

- Adrià Muro Gómez (1665191)
- David Morillo Massagué (1666540)

Data: 11/05/2025

Introducció.....	3
Problema a resoldre.....	3
Disseny de l'arquitectura.....	4
Diagrama de l'arquitectura.....	5
Justificació de les decisions.....	6
PROs i CONs del disseny.....	6
PROs:.....	6
CONs:.....	6
Possibles implementacions de millora.....	7
Creació de l'arquitectura.....	7
Virtual Private Cloud (VPC).....	7
Security Group (SG).....	9
Load Balancer (LB).....	9
Target groups (TG).....	10
Templates Blue/Green.....	10
Auto Scaling Groups (ASG).....	11
Comprovació del correcte funcionament.....	12
Traffic Split.....	14
Conclusions.....	18

Introducció

La metodologia tradicional de desplegar aplicacions, normalment presenta riscos importants per a la continuïtat del servei. Els **temps d'inactivitat** poden afectar l'experiència de l'usuari, especialment en entorns amb **alta demanda**. A més, la falta d'una estratègia robusta per gestionar errors o retrocessos fa que els desenvolupadors hagin de reaccionar de manera reactiva a incidents imprevists.

Davant d'aquest problema, les tècniques de desplegament Blue/Green són una solució moderna i eficaç, amb l'objectiu de minimitzar els riscos i els temps de parada, permetent actualitzacions contínues **sense interrupcions**. Aquesta estratègia separa les dues versions d'una aplicació, mantenint una versió activa (Blue) i una versió nova (Green), de manera que es pugui fer la transició entre ambdues amb el mínim impacte. Aquesta pràctica també facilita la detecció d'errors i la possibilitat de retrocedir a la versió anterior (Blue) de forma ràpida si fos necessari (rollback).

En aquesta pràctica, implementarem un desplegament Blue/Green per a una aplicació al núvol, demostrant com aquest enfocament pot millorar la qualitat del servei sense interrupcions. A més, discutirem la necessitat d'adoptar aquesta metodologia davant de les limitacions de les tècniques tradicionals, destacant els avantatges de la **seguretat**, la **flexibilitat** i la **fiabilitat** del desenvolupament d'aplicacions i les operacions al núvol.

Problema a resoldre

Es requereix **alta disponibilitat** per evitar interrupcions del servei durant el desplegament de noves versions. Es farà servir el patró de desplegament **blue/green** per migrar el servidor web de "httpd" a "nginx".

La configuració inclou dos entorns de producció idèntics dins del VPC:

- **Entorn blau:** aplicació actual.
- **Entorn verd:** nova versió de l'aplicació.

Estratègia de distribució del trànsit:

1. Inici amb una distribució 80/20 (blau/verd).
2. Passar a 50/50.
3. Ajustar a 20/80.
4. Finalment, redirigir tot el trànsit cap a l'entorn verd.

Aquesta estratègia permet un **desplegament sense interrupcions** visibles per a l'usuari i possibilita una reversió ràpida si hi ha problemes amb la nova versió.

Disseny de l'arquitectura

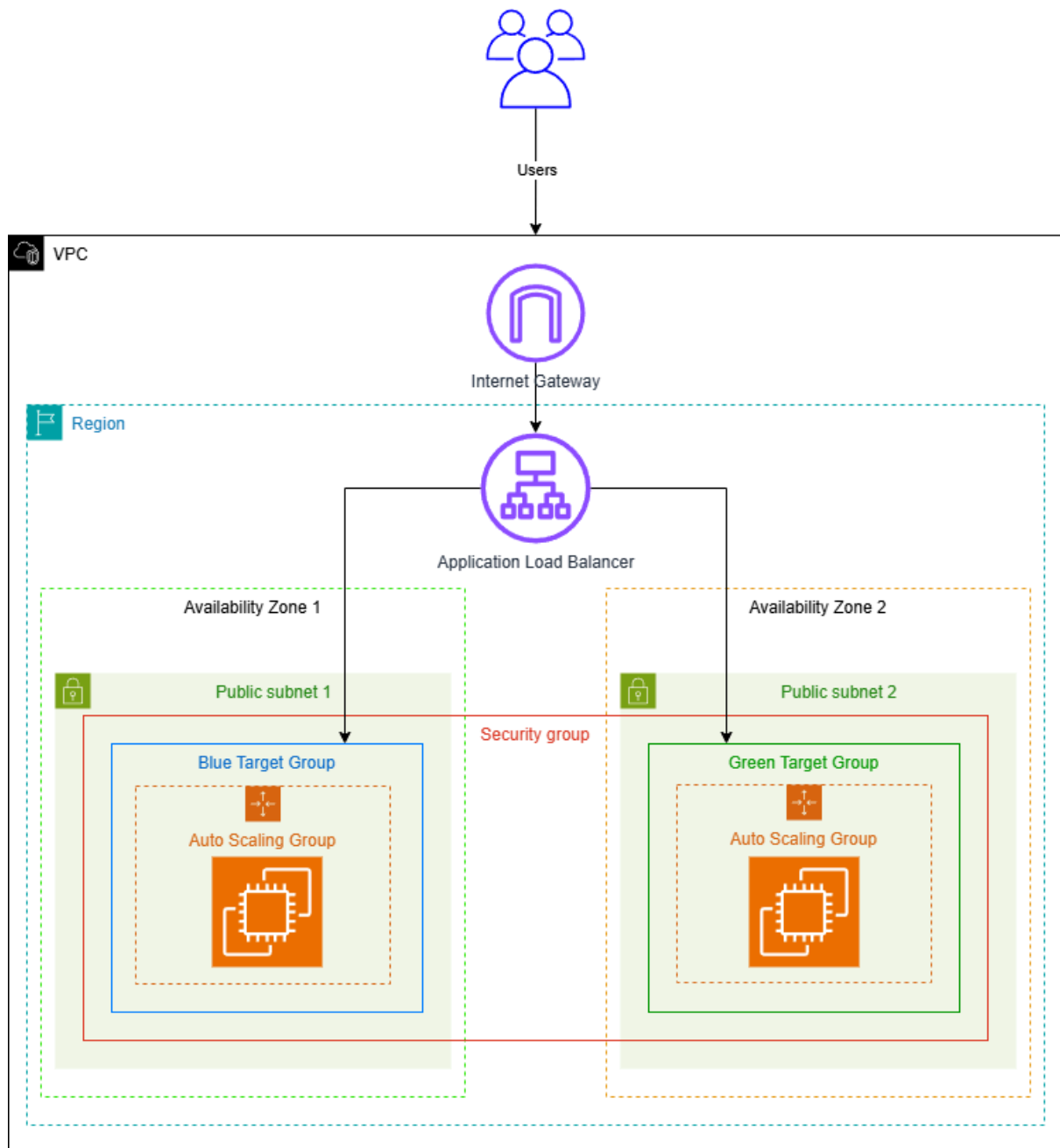
L'arquitectura de l'entorn Blue/Green Deployment a AWS es compon dels següents elements principals:

1. **VPC:** Un núvol privat amb una sola regió i dues Availability Zones (AZs) en el nostre cas. Per al diagrama donat s'han utilitzat només dues AZs, ja que considerem suficients per a demostrar la capacitat del Load Balancer de gestionar el tràfic per tot el VPC, a la vegada que intentem maximitzar la simplicitat del diagrama. Aquesta configuració ja proporciona un cert grau d'alta disponibilitat i tolerància a fallades, encara que si falla una de les AZs, només es podria accedir a una de les versions (green/blue)
2. **Subxarxes:**
 - Dues subxarxes públiques per allotjar les instàncies EC2 (Blue i Green). Per simplicitat no fem ús de xarxes privades en aquesta pràctica, ja que no és un aspecte essencial en el desenvolupament d'aquesta, però coneixem la importància d'aquesta en la vida real, en cas de llançar aquestes aplicacions a producció.
3. **Load Balancer (ALB):** Gestiona el trànsit entre *Blue* i *Green* amb una distribució gradual (80/20, 50/50, 20/80, 0/100). Aquest load balancer rep el tràfic a partir de l'Internet Gateway de la VPC, i dirigeix el tràfic entre els dos Target Groups, explicats a continuació.
4. **Target Groups:** Un per a cada versió (*Blue* i *Green*) per dirigir el trànsit a les instàncies corresponents. En el nostre cas, cada Target Group conté una sola instància EC2, però en entorns reals el més habitual és tenir múltiples instàncies distribuïdes per AZs per escalar l'aplicació horitzontalment.
5. **Auto Scaling Groups:**
 - Un per a la versió *Blue*
 - Un altre per a la versió *Green*

Cada Auto Scaling Group pot escalar automàticament el nombre d'instàncies EC2 segons, per exemple, l'ús de la CPU.
6. **Instàncies EC2 (Blue i Green):**
 - *Blue:* Apache HTTP Server
 - *Green:* Nginx Server

Utilitzem una instància per cada tipus d'aplicació, gestionades pels respectius Auto Scaling Groups, i són el backend de cadascuna de les versions.

Diagrama de l'arquitectura



Justificació de les decisions

- **Elecció del Load Balancer (ALB):** S'ha escollit un Application Load Balancer (ALB) en lloc d'un Network Load Balancer (NLB) perquè ALB ofereix una millor gestió per al trànsit HTTP i HTTPS. ALB permet treballar amb rutes d'URL, que són útils quan es necessita gestionar trànsit específic per a diferents versions de l'aplicació (*Blue* i *Green*). A més, ALB proporciona una millor integració amb les funcions de Auto Scaling i target groups.
- **Ús d'Auto Scaling Groups:** Els Auto Scaling Groups es consideren una millor opció per gestionar l'escalabilitat de les instàncies EC2 perquè poden afegir o eliminar màquines de manera automàtica en resposta a canvis en la demanda de trànsit. Això millora l'eficiència i redueix el cost operatiu, ja que només es mantenen les instàncies necessàries en tot moment.
- **Divisió de trànsit gradual entre Blue i Green:** La divisió inicial del trànsit (80/20) permet provar la nova versió amb una càrrega limitada abans de fer la migració completa a *Green*. Això redueix el risc de problemes greus que podrien sorgir en un desplegament immediat a tota l'audiència.

PROs i CONs del disseny

PROs:

- **Alta disponibilitat:** Amb dues AZs i instàncies duplicades (*Blue/Green*), l'arquitectura està dissenyada per garantir que si una zona falla, l'altre pot seguir operant sense interrupcions.
- **Desplegament sense downtime:** El sistema *Blue/Green* permet actualitzar l'aplicació sense que els usuaris percebin interrupcions, ja que el trànsit es distribueix gradualment entre les dues versions.
- **Escalabilitat automàtica:** Els *Auto Scaling Groups* permeten adaptar-se automàticament a la demanda de trànsit, afegint o eliminant instàncies segons sigui necessari.
- **Rollback ràpid:** Si la versió *Green* té problemes, el trànsit es pot redirigir ràpidament cap a *Blue*, minimitzant l'impacte de possibles errors.

CONs:

- **Costos elevats:** Mantenir dues versions de l'aplicació en funcionament al mateix temps pot ser costós, ja que implica duplicar els recursos de computació, emmagatzematge i altres components.
- **Complexitat de gestió:** La necessitat de gestionar dues versions (*Blue/Green*), *Auto Scaling Groups* i *Target Groups* pot ser complexa, especialment en entorns més grans amb múltiples serveis.
- **Temps de validació de la nova versió:** Encara que el trànsit es distribueix gradualment, cal temps per validar que la versió Green funciona correctament abans de completar la migració.

Possibles implementacions de millora

- **Implementació de Monitoratge Avançat amb Alarms:** Es podria afegir un monitoratge més exhaustiu amb alarms de CloudWatch per controlar paràmetres específics com la latència de la resposta del servidor, el temps de càrrega de pàgines o el nombre d'errors. Això permetria detectar problemes en temps real i prendre mesures abans que afectin als usuaris finals.
- **Reforçar la Seguretat:** Es podria millorar la seguretat de l'arquitectura implementant mecanismes com AWS Shield per protegir l'aplicació davant atacs DDoS o aplicant regles de seguretat més estrictes a les subxarxes privades i públiques. A més, l'ús de certs protocols de seguretat (com SSL/TLS) per a les comunicacions pot millorar la protecció de les dades en trànsit

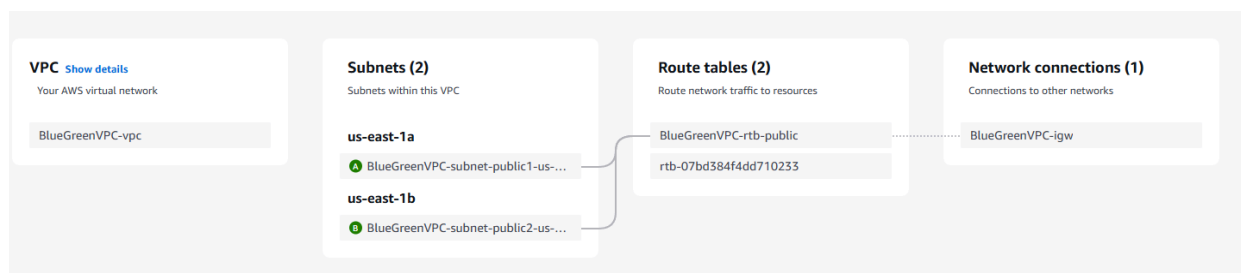
Creació de l'arquitectura

Virtual Private Cloud (VPC)

Per garantir una arquitectura amb alta disponibilitat i preparació per al desplegament Blue/Green, hem creat una VPC, anomenada com a *BlueGreenVPC-vpc*, amb un CIDR block 10.0.0.0/16, que permet gestionar internament una àmplia gamma d'adreces IP privades.

Dins d'aquesta VPC hem configurat:

- **Dues subxarxes públiques** situades en zones de disponibilitat diferents per assegurar redundància:
 - **Subxarxa 1:** *BlueGreenVPC-subnet-public1-us-east-1a* amb CIDR 10.0.0.0/20
 - **Subxarxa 2:** *BlueGreenVPC-subnet-public2-us-east-1b* amb CIDR 10.0.16.0/20
- **Internet Gateway** per permetre connexions entrants/sortints a Internet.
- **Route Table** associada a les dues subxarxes per permetre el trànsit extern via l'Internet Gateway



Security Group (SG)

sg-052d9c0feba9d4ff4 - SG-BlueGreen

Details

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Details

Security group name

SG-BlueGreen

Security group ID

sg-052d9c0feba9d4ff4

Description

Allow SSH, HTTPS, HTTP

VPC ID

vpc-001b84f8eb22f9bfc

Owner

194357582553

Inbound rules count

3 Permission entries

Outbound rules count

1 Permission entry

Hem creat un únic Security Group (SG) anomenat *SG-BlueGreen* amb les següents regles d'entrada per permetre:

- Port 80 (**HTTP**): per accedir a la web des del navegador.
- Port 443 (**HTTPS**): per futures connexions segures (encara que no s'ha implementat en aquesta pràctica).
- Port 22 (**SSH**): restringit a una única IP (79.158.224.171/32) per connexió segura a les instàncies EC2 només desde la IP pròpia.

Inbound rules (3)										Manage tags
Search										
<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source			
<input type="checkbox"/>	-	sgr-0bc62724c0bc6b537	IPv4	HTTP	TCP	80	0.0.0.0/0			
<input type="checkbox"/>	-	sgr-0a5a6a20ba61918ff	IPv4	HTTPS	TCP	443	0.0.0.0/0			
<input type="checkbox"/>	-	sgr-036f5ac70a1ec03ef	IPv4	SSH	TCP	22	79.158.224.171/32			

Aquest SG s'ha associat a totes les instàncies (tant *Blue* com *Green*).

Load Balancer (LB)

Per distribuir el trànsit entre els dos entorns, hem creat un *Application Load Balancer* anomenat *LB-GB*, que opera sobre **HTTP** (port 80) i cobreix les dues zones de disponibilitat. Això assegura redundància i alta disponibilitat.

El DNS públic del Load Balancer és:

- ***LB-GB-1100012921.us-east-1.elb.amazonaws.com***

Aquest LB és el punt d'entrada comú dels usuaris i redirigeix el trànsit cap als target groups (i, per tant, a les instàncies EC2).

Resource map [Info](#)

View, explore, and troubleshoot your load balancer's architecture.

Overview

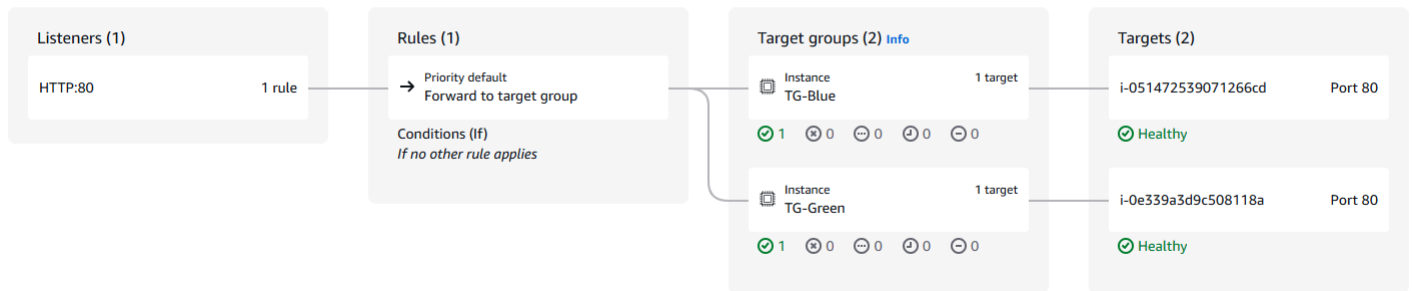
Unhealthy target map



Show resource details

LB-GB

Last fetched seco



Target groups (TG)

Hem creat **dos** Target Groups:

- *TG-Blue*
- *TG-Green*

Els dos són del tipus **Instance** i escolten per HTTP al port 80. Cada target group apunta a un Auto Scaling Group diferent, de manera que es manté clarament separada la infraestructura de l'entorn *blue* i *green*. Aquesta separació és clau per poder fer la gestió del trànsit (i el rollback) amb flexibilitat. A més, els Target Groups **estan assignats al Load balancer** creat anteriorment.

Target groups (2) [Info](#)

Filter target groups

<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer
<input type="checkbox"/>	TG-Blue	arn:aws:elasticloadbalancin...	80	HTTP	Instance	LB-GB
<input type="checkbox"/>	TG-Green	arn:aws:elasticloadbalancin...	80	HTTP	Instance	LB-GB

Templates Blue/Green

Hem creat dues plantilles de llançament (launch templates), una per cada entorn:

- *BlueT*
- *GreenT*

Totes dues fan servir:

- Amazon Linux 2
- T2.micro

- El SG definit anteriorment
- No especifiquen subxarxa (s'assigna a nivell de ASG)

El que diferencia un template d'un altre és el User Data:

- En el cas de Blue **instal·la Apache i mostra un HTML blau**

```
#!/bin/bash
yum install -y httpd
cat << HEREDOC > /var/www/html/index.html
<html><body bgcolor="#5DBCD2"><h1>Blue Environment</h1></body></html>
HEREDOC
chkconfig httpd on
service httpd start
```

- En el cas de Green **instal·la nginx i mostra un HTML verd**

```
#!/bin/bash
amazon-linux-extras enable nginx1
yum clean metadata
yum install -y nginx
mv /usr/share/nginx/html/index.html /usr/share/nginx/html/index.html.orig
cat << 'EOF' > /usr/share/nginx/html/index.html
<html><body bgcolor="#98FB98"><h1>Green Environment</h1></body></html>
EOF
systemctl enable nginx
systemctl start nginx
```

Auto Scaling Groups (ASG)

Hem creat **dos Auto Scaling Groups**:

- ASG-Blue amb plantilla *BlueT* i assignat a *TG-Blue*
- ASG-Green amb plantilla *GreenT* i assignat a *TG-Green*

Cada ASG treballa en una zona diferent:

- Blue → subnet 1 (us-east-1a)
- Green → subnet 2 (us-east-1b)

Les configuracions de capacitat són:

- Desired Capacity: 1
- Min Capacity: 1
- Max Capacity: 2

Afegim una **política d'escalat** basada en la utilització mitjana de CPU al 50%.

Auto Scaling groups (2) <small>Info</small>								
Last updated 11 minutes ago Refresh Launch configurations Launch templates <small>?</small> Actions								
Search your Auto Scaling groups								
<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	ASG-Blue	BlueT Version Default	1	-	1	1	2	us-east-1a
<input type="checkbox"/>	ASG-Green	GreenT Version Default	1	-	1	1	2	us-east-1b

Comprovació del correcte funcionament

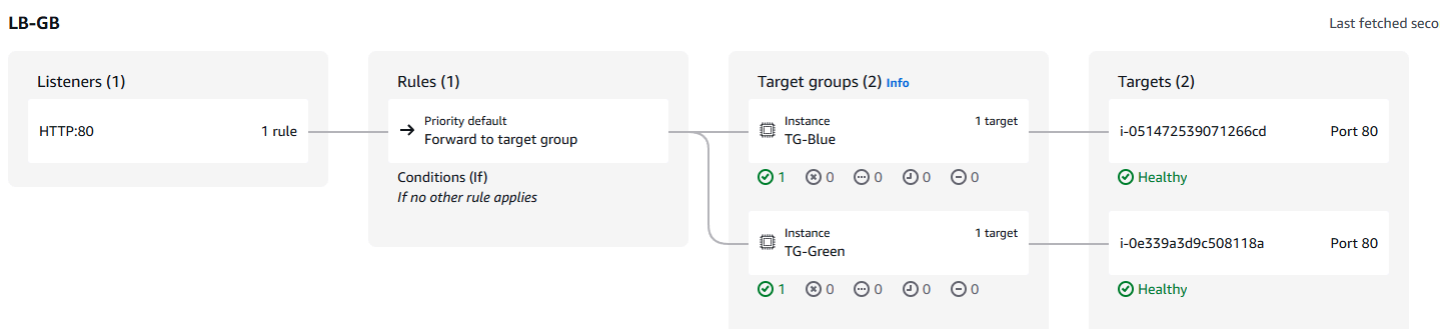
Per verificar que tota l'arquitectura desplegada funciona correctament, s'han realitzat diverses comprovacions finals centrades en la connexió entre components, el funcionament dels entorns *Blue* i *Green* i la distribució de trànsit mitjançant l'Application Load Balancer.

Resource Map

En primer lloc, s'ha revisat el resource map del servei EC2 per comprovar visualment que tots els recursos es troben correctament interconnectats. Això inclou:

- El **Load Balancer** (LB-GB), de tipus Application Load Balancer, desplegat en ambdues subxarxes públiques de la VPC.
- Els dos **Target Groups** (TG-Blue i TG-Green), associats al listener HTTP:80 del Load Balancer.
- Les **instàncies** creades automàticament pels Auto Scaling Groups (ASG-Blue i ASG-Green) i registrades correctament en els seus respectius Target Groups.

Aquest mapa permet visualitzar fàcilment com el trànsit extern pot arribar a les instàncies de cada entorn a través del Load Balancer.



Accés als entorns Blue/Green

Abans de realitzar la comprovació de balanceig de trànsit, s'ha validat que les instàncies creades per cada ASG funcionen correctament de forma individual. Per fer-ho, s'ha accedit directament a la **IP pública** de cada instància per comprovar el contingut servit.

En el cas de l'entorn *Blue*, s'ha pogut accedir a una pàgina HTML bàsica servida amb Apache, amb fons blau i un missatge identificant clarament que es tracta de l'entorn *Blue*.



En el cas de l'entorn *Green*, s'ha accedit correctament a una pàgina servida amb nginx, amb fons verd i missatge personalitzat indicant que és l'entorn *Green*.

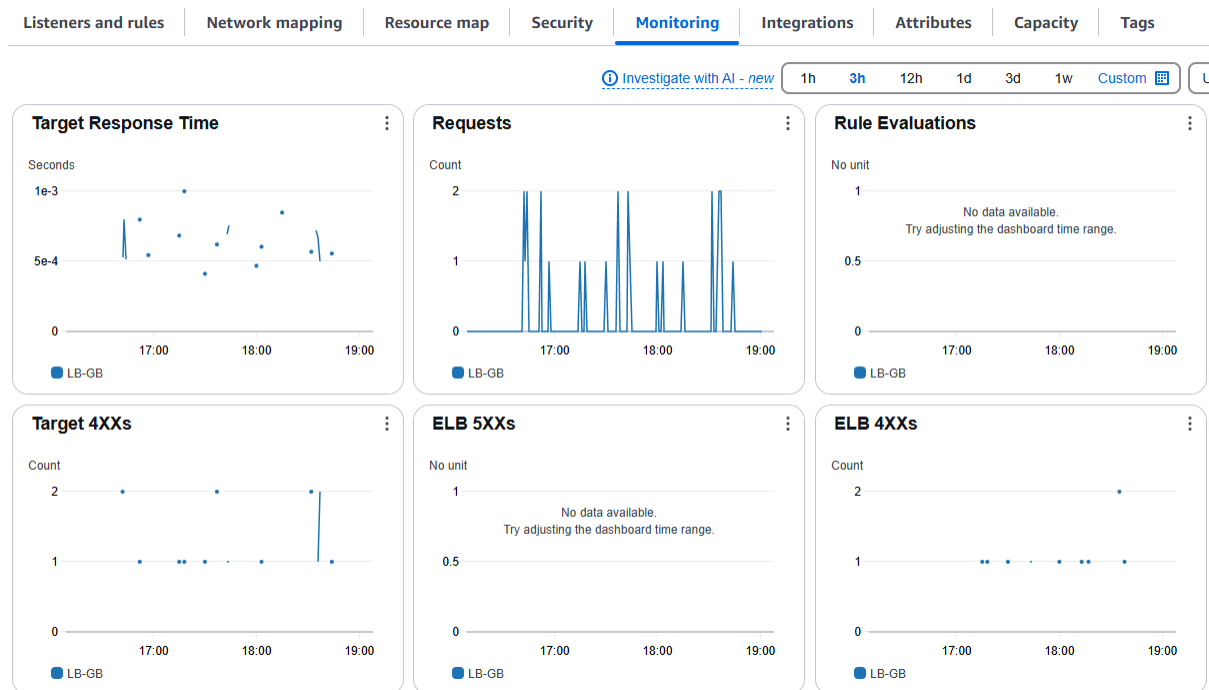


Aquest pas confirma que el user data de cada Launch Template ha funcionat correctament i que cada instància està operativa i preparada per rebre trànsit.

Monitoratge de l'estat dels serveis

Per complementar la comprovació anterior, s'ha consultat el sistema de monitoratge integrat d'AWS per verificar que els serveis associats als dos entorns es troben en bon estat.

Les gràfiques indiquen clarament que els dos serveis (*Blue* i *Green*) estan actius i responen correctament a les peticions de salut (health checks). Això confirma que el Load Balancer considera les instàncies com a **operatives** i que **poden rebre trànsit**.



Traffic Split

L'objectiu central d'aquesta pràctica és implementar un desplegament *Blue/Green* utilitzant un *Application Load Balancer* (ALB) per controlar la transició del trànsit entre dues versions independents del servei. Aquest enfocament permet dur a terme una **migració progressiva, segura i totalment reversible**.

La distribució del trànsit entre els entorns *Blue* i *Green* es controla mitjançant la configuració del listener del *Load Balancer*. Per fer-ho s'ha seleccionat el *LB-GB*, i dins l'apartat *Listeners*, s'ha editat la **regla per defecte del listener** HTTP (port 80).

A través d'aquesta regla, és possible assignar **diferents percentatges** del trànsit a diversos Target Groups (en aquest cas *TG-Blue* i *TG-Green*), permetent així una distribució personalitzada i ajustable en temps real.

Aquest mecanisme és essencial per implementar desplegaments *Blue/Green* progressius, on es pot controlar amb precisió quin percentatge d'usuaris accedeix a cada versió del sistema.

Per validar el comportament, s'ha accedit repetidament a la URL pública del Load Balancer ***LB-GB-1100012921.us-east-1.elb.amazonaws.com*** des d'un navegador web, amb l'objectiu d'observar quin entorn es servia en funció dels percentatges configurats.



S'han fet proves visuals carregant la pàgina múltiples vegades i comparant si es mostrava el fons blau (entorn Blue amb Apache) o el verd clar (entorn Green amb Nginx), segons el cas.

Els resultats observats han estat els següents:

80% Blue – 20% Green: El sistema *Blue* domina clarament la càrrega. No obstant això, en recarregar diverses vegades la pàgina, es pot veure de tant en tant l'entorn *Green*, indicant que el 20% de trànsit s'està desviant correctament.

Details	Listeners and rules	Network mapping	Resource map
Listeners and rules (1) Info			
A listener checks for connection requests on its configured protocol and port. Traffic receiver			
<input type="text" value="Filter listeners"/>			
<input type="checkbox"/>	Protocol:Port	Default action	
<input type="checkbox"/>	HTTP:80	Forward to target group <ul style="list-style-type: none">TG-Blue: 80 (80%)TG-Green: 20 (20%)Target group stickiness: Off	

50% Blue – 50% Green: En aquest cas s'ha detectat un repartiment equitatiu. A l'accedir repetidament al Load Balancer, aproximadament la meitat de les vegades es mostrava la pàgina blava i l'altra meitat la verda. Aquest escenari és ideal per validar el comportament de les dues versions funcionant en paral·lel.

Rules

Attributes

Tags

Listener rules (1) Info

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the

Filter rules

<input type="checkbox"/>	Name tag	Priority ▲	Conditions (If)	Actions (Then)
<input type="checkbox"/>	Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none">TG-Blue 50 (50%)TG-Green 50 (50%)Target group stickiness: Off

20% Blue – 80% Green: El comportament s’inverteix respecte a l’inicial. L’entorn *Green* apareix amb molta més freqüència en les peticions, tal com estava previst, i només puntualment es mostra la versió blava.

Rules

Attributes

Tags

Listener rules (1) Info

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from th

Filter rules

<input type="checkbox"/>	Name tag	Priority ▲	Conditions (If)	Actions (Then)
<input type="checkbox"/>	Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none">TG-Green 80 (80%)TG-Blue 20 (20%)Target group stickiness: Off

0% Blue – 100% Green: El trànsit passa íntegrament a l’entorn nou. Totes les càrregues de pàgina mostren la versió verda, confirmant que el canvi s’ha completat correctament i que només s’està servint des del *TG-Green*.

Listener rules (1) Info

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from

Filter rules

<input type="checkbox"/>	Name tag	Priority ▲	Conditions (If)	Actions (Then)
<input type="checkbox"/>	Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none">TG-Green 100 (100%)TG-Blue 0 (0%)Target group stickiness: Off



Aquest procediment ha permès validar que la configuració del listener i la distribució de trànsit entre Target Groups es comporta tal com s'esperava, reflectint fidelment els percentatges assignats.

Rollback

En un entorn *Blue/Green*, una de les principals funcionalitats que es busca és la possibilitat de **revertir** el desplegament de forma ràpida i segura en cas que el nou entorn (*Green*) presenti errors o no compleixi amb les expectatives funcionals.

En cas que fos necessari revertir el desplegament a l'entorn anterior (*Blue*), el procés seria pràcticament el mateix que el canvi de trànsit cap al nou entorn *Green*, però en **ordre invers**. Així, es podrien modificar els percentatges de trànsit a través del *Load Balancer*, redirigint el trànsit cap al *Target Group Blue* i reduint progressivament o immediatament el trànsit cap al *Target Group Green*.

Aquest procés de rollback es pot realitzar de manera **progressiva**, ajustant els percentatges de trànsit de manera escalonada (per exemple, de 20/80 a 50/50, després a 80/20 i finalment a 100/0), o de manera **immediata**, redirigint tot el trànsit al servei *Blue* sense cap interacció adicional.

Aquest control permet garantir la disponibilitat del servei en cas de fallades, assegurant una transició suau entre versions de l'aplicació.

Conclusions

Hem demostrat que el model Blue/Green Deployment és molt més robust i segur en relació a les tècniques de desplegament tradicionals. Aquesta enfocament ens ha permès preparar un sistema que pot actualitzar versions sense interrupció del servei, evitant els típics problemes de disponibilitat o errors que sovint sorgeixen amb mètodes més antics. L'ús d'un Application Load Balancer per repartir el trànsit entre entorns, juntament amb Auto Scaling Groups i Target Groups separats, ens dona un control molt més precís sobre el comportament del sistema. Tot i que en aquesta pràctica no hem hagut d'executar un rollback real, hem deixat l'arquitectura preparada per fer-ho en cas d'emergència, i que ens permeti tornar a la versió anterior (Blue).

Aquest nivell de flexibilitat i seguretat no només millora l'experiència de l'usuari final, sinó que també facilita el treball dels desenvolupadors un cop implementat, ja que permet fer canvis amb confiança i menys risc. En definitiva, el model Blue/Green mostra com una bona estratègia de desplegament d'aplicacions pot reforçar la fiabilitat i estabilitat de les aplicacions en entorns de núvol, i creiem que és una evolució necessària davant les limitacions de les tècniques tradicionals. De cara al futur, podríem integrar monitoratge avançat (CloudWatch + CloudTrail) i eines de seguretat per portar encara més beneficis, però el que hem vist en aquesta pràctica ja marca una diferència clara i significativa, i ens ha servit per a aprendre la part més important d'aquest tipus de desplegament. Si volguéssim implementar el nostre sistema a nivell professional, deixariem de banda la simplicitat d'aquesta pràctica i utilitzariem més de dos Availability Zones, fariem ús de xarxes privades, i un seguit d'eines d'AWS per a afegir més disponibilitat i seguretat al sistema.