AWS GAME DAY 2023

NIVEL INICIAL

DIVERTITE, JUGÁ Y APRENDÉ HACIENDO

Sponsor

Haciendo foco en:

- EC2
- VPC
- AutoScaling
- ELB/ALB

- CloudFront
- Elasticache
- S3
- CloudWatch
- ECS/Fargate

Recursos

YouTube

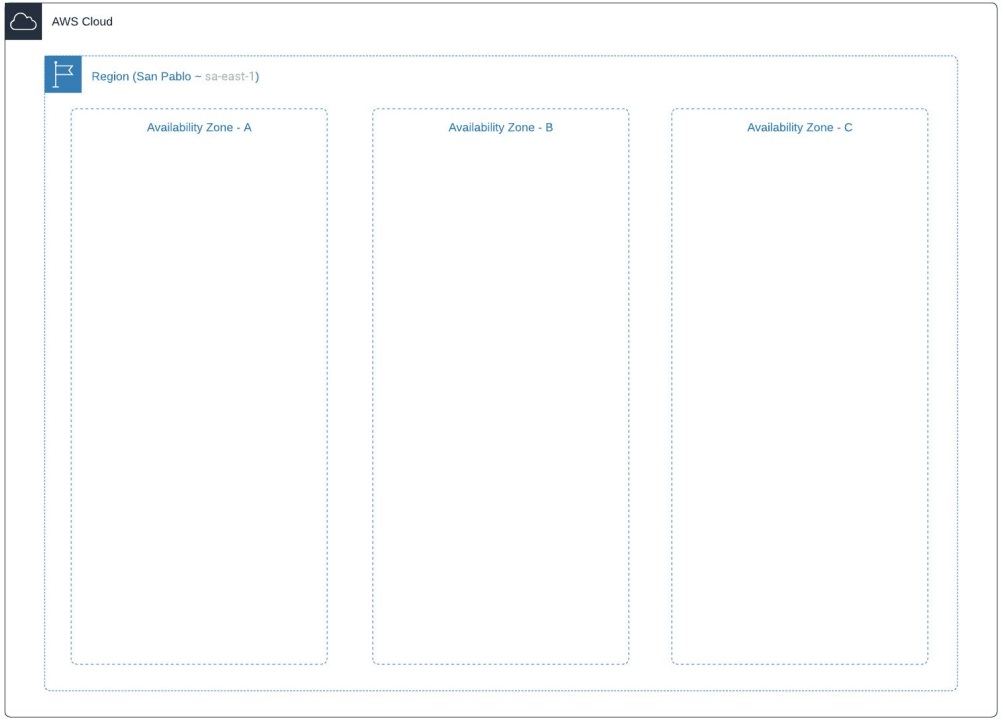
- Sesión 1 "IAM"
- Sesión 2 "VPC"
- Sesión 3 "EC2"
- Sesión 4 "AWS Elastic Load Balancers y Auto Scaling"
- Sesión 5 "Storage (EBS, EFS, S3) + Monitoring/Log"
- Sesión 6 "Cloudfront+Route 53"

- Sesión 7 "Databases"
- Sesión 8 "Serverless"
- Sesión 9 "Security & Compliance"
- Sesión 10 "Billing & Support"
- Sesión 11 "TIPS"

VPC from Scratch

Unicorn.Rentals

1. Building a Custom VPC from Scratch in AWS



Networking

Internet GtW ~vs.~ Nat Gateway
Nat Instance ~vs.~ Nat Gateway
Nat Instance ~vs.~ Bastion
Nat Instance ~ Disabling source/destination checks

Route Table Public ~> Path to ~ Internet Gateway
Route Table Private ~> Path to ~ Nat Gateway

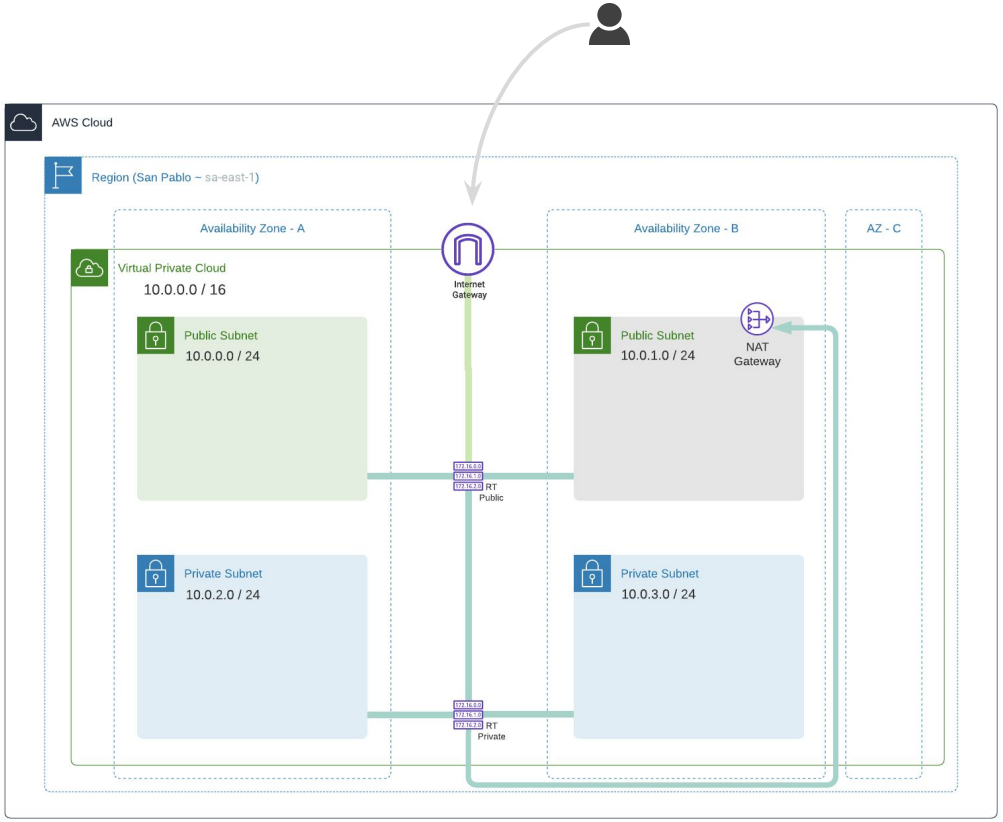
Security Group ~Stateful ~ (allow)
NACL ~ Stateless ~ (allow & deny)

Security Group, NACL Default - Allow All
Security Group, NACL Custom - Deny All

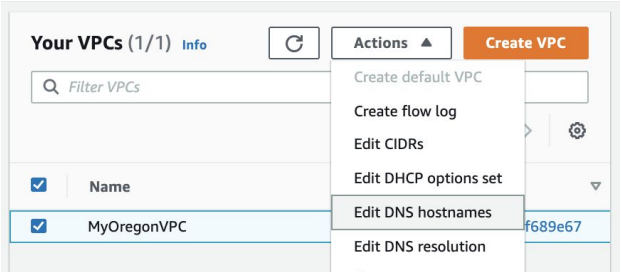
ICMP, Ephemeral Ports

1. Building a Custom VPC from Scratch in AWS

Redes y Ruteo



RT-Public	10.0.0.0 / 16 →	local	RT-Private	10.0.0.0 / 16 →	local
	0.0.0.0 / 0 →	igw-		0.0.0.0 / 0 →	ngw-



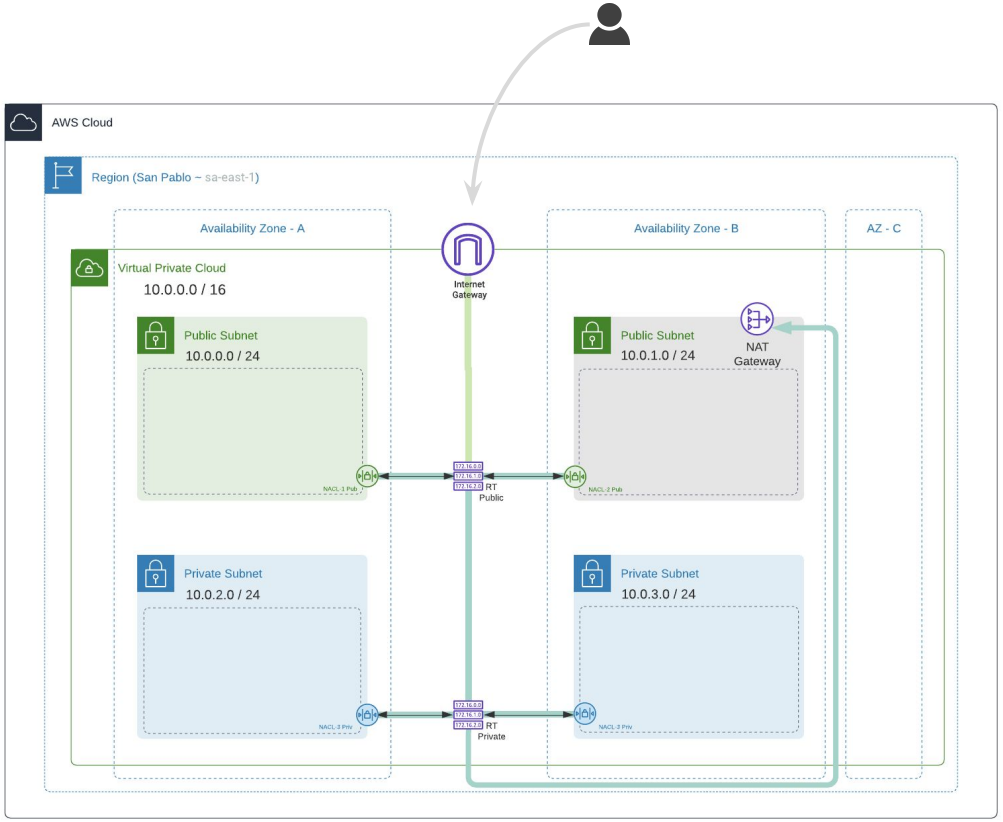
Auto-assign IPv4 Info

☑ Enable auto-assign public IPv4 address

RT-Public	0.0.0.0 / 0 →	igw-	
RT-Public	10.0.0.0 / 16 →	local	
RT-Private	0.0.0.0 / 0 →	nat-	
RT-Private	10.0.0.0 / 16 →	local	

1. Building a Custom VPC from Scratch in AWS

Seguridad - “Física”



RT-Public

10.0.0.0 / 16	→	local
0.0.0.0 / 0	→	igw-

RT-Private

10.0.0.0 / 16	→	local
0.0.0.0 / 0	→	ngw-

Caso de USO: Internet - Web Server - BD

NACL-Public-A — Inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
200	SSH (22)	TCP (6)	22	181.164.85.20/32	Allow
300	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Outbound rules

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

NACL-Private — Inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	MySQL/Aurora (3306)	TCP (6)	3306	10.0.0.0/24	Allow
200	SSH (22)	TCP (6)	22	10.0.0.243/32	Allow
300	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

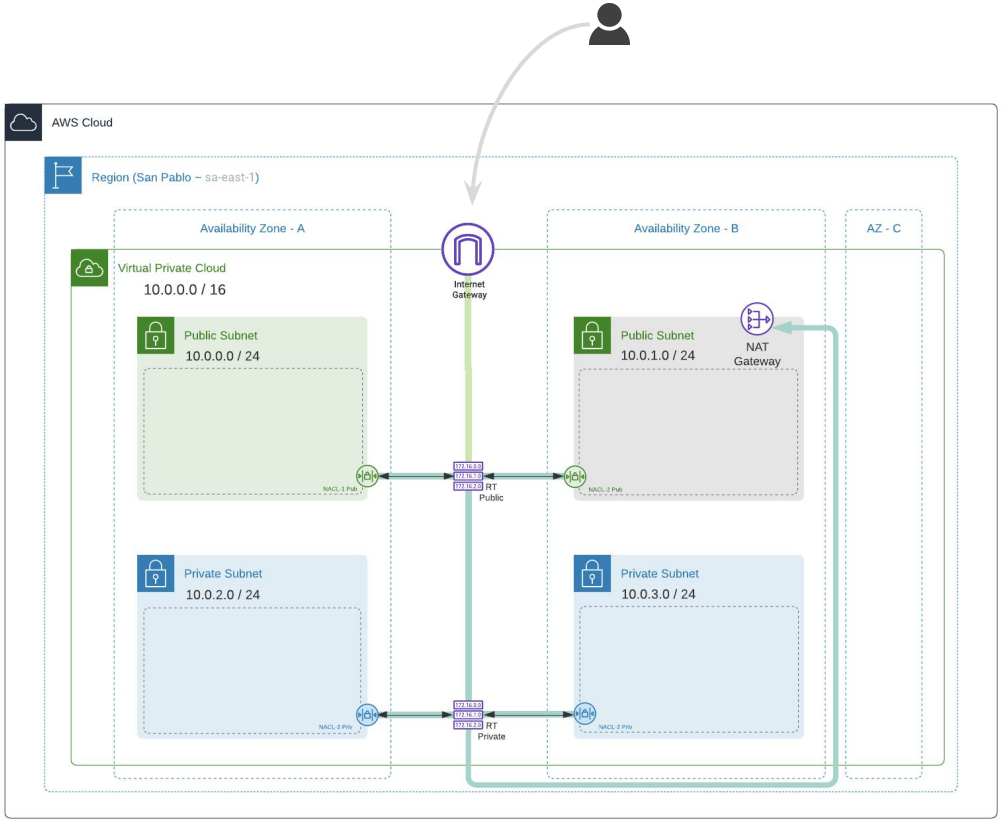
Outbound rules

100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

ephemeral ports

1. Building a Custom VPC from Scratch in AWS

Seguridad - “Física”



RT-Public
10.0.0.0 / 16 → local
0.0.0.0 / 0 → igw-

RT-Private
10.0.0.0 / 16 → local
0.0.0.0 / 0 → ngw-

Caso de USO: Update SW. Privado por Nat Gwy

NACL-Public-B — Inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	HTTP (80)	TCP (6)	80	10.0.2.0/24	Allow
200	HTTPS (443)	TCP (6)	443	10.0.2.0/24	Allow
300	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Outbound rules

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
200	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
300	Custom TCP	TCP (6)	1024 - 65535	10.0.2.0/24	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

NACL-Private — Inbound rules

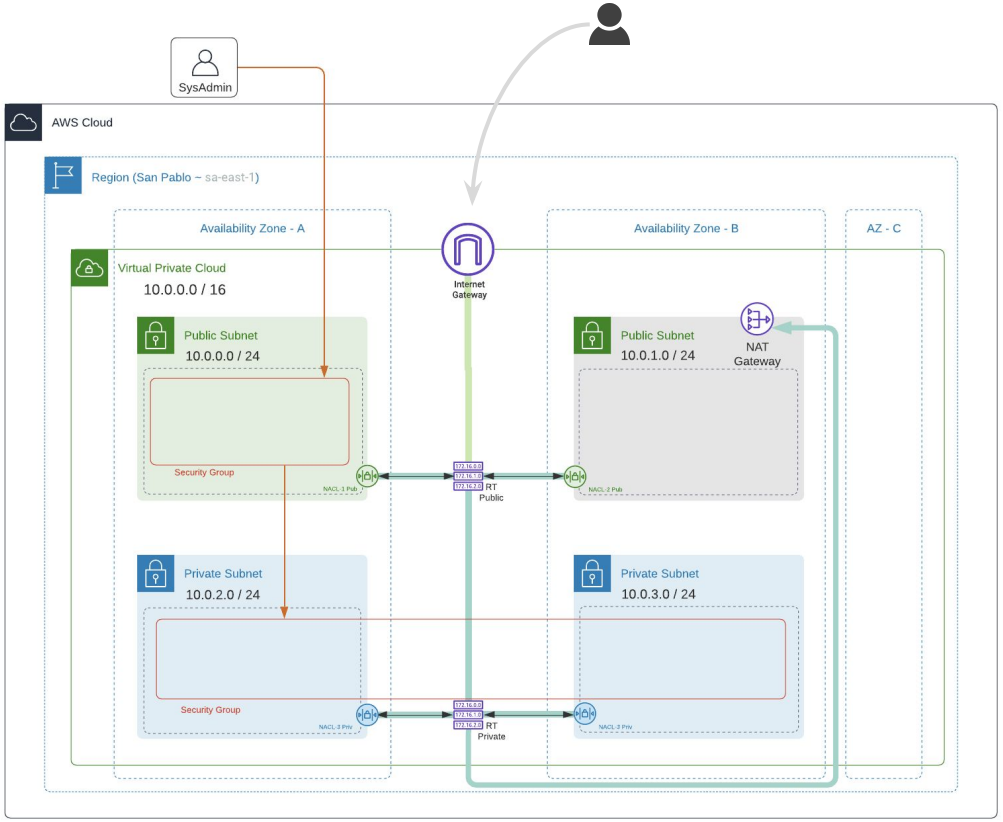
Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	MySQL/Aurora (3306)	TCP (6)	3306	10.0.0.0/24	Allow
200	SSH (22)	TCP (6)	22	10.0.0.243/32	Allow
300	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Outbound rules

100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

1. Building a Custom VPC from Scratch in AWS

Seguridad - “Lógica”



RT-Public

10.0.0.0 / 16	→	local
0.0.0.0 / 0	→	igw-

RT-Private

10.0.0.0 / 16	→	local
0.0.0.0 / 0	→	ngw-

WebDMZ-SG

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
SSH	TCP	22	181.164.85.20/32	-
Custom TCP	TCP	1024 - 65535	0.0.0.0/0	User-Data Setup (yum update, git clone)

Outbound rules

Type	Protocol	Port range	Destination
All TCP	TCP	0 - 65535	0.0.0.0/0

MyDB-SG

Inbound rules

Type	Protocol	Port range	Source
SSH	TCP	22	sg- (WebDMZ-SG)
MYSQL/Aurora	TCP	3306	sg- (WebDMZ-SG)

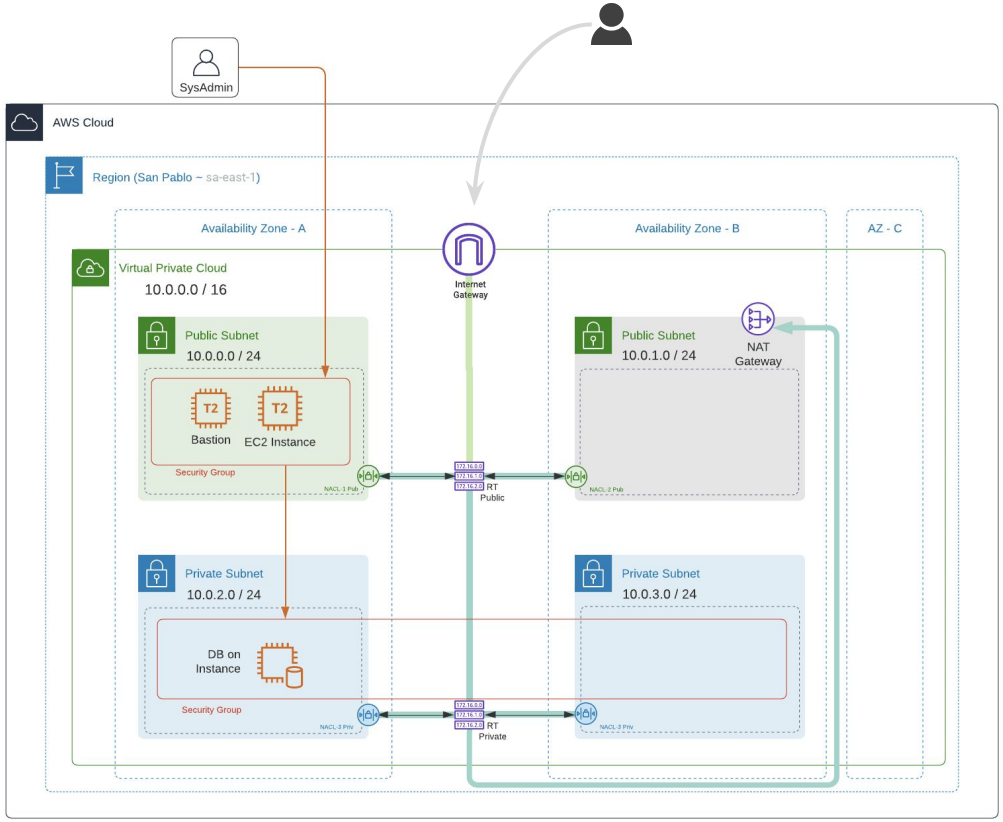
Outbound rules

Type	Protocol	Port range	Destination
All traffic	All	All	0.0.0.0/0

3306 | desde SG-1
22 | desde SG-1

1. Building a Custom VPC from Scratch in AWS

Despliegue



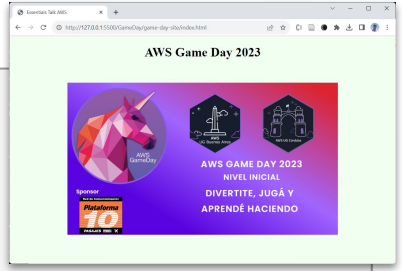
RT-Public

10.0.0.0 / 16	→	local
0.0.0.0 / 0	→	igw-

RT-Private

10.0.0.0 / 16	→	local
0.0.0.0 / 0	→	ngw-

EC2 Web




```
#!/bin/bash
yum update -y
yum install httpd -y
yum install git -y
yum install php -y
service httpd start
chkconfig httpd on

#------------------------
echo "Hello Static Site!"  > /var/www/html/index.html

#------------------------
echo "<?php echo '<p>Hello PHP World</p>'; ?>" > /var/www/html/index.php

#------------------------
# Static web Site

git clone https://github.com/Pabloin/AWS-Essentials.git

cp -r AWS-Essentials/GameDay/game-day-site/* /var/www/html/
```

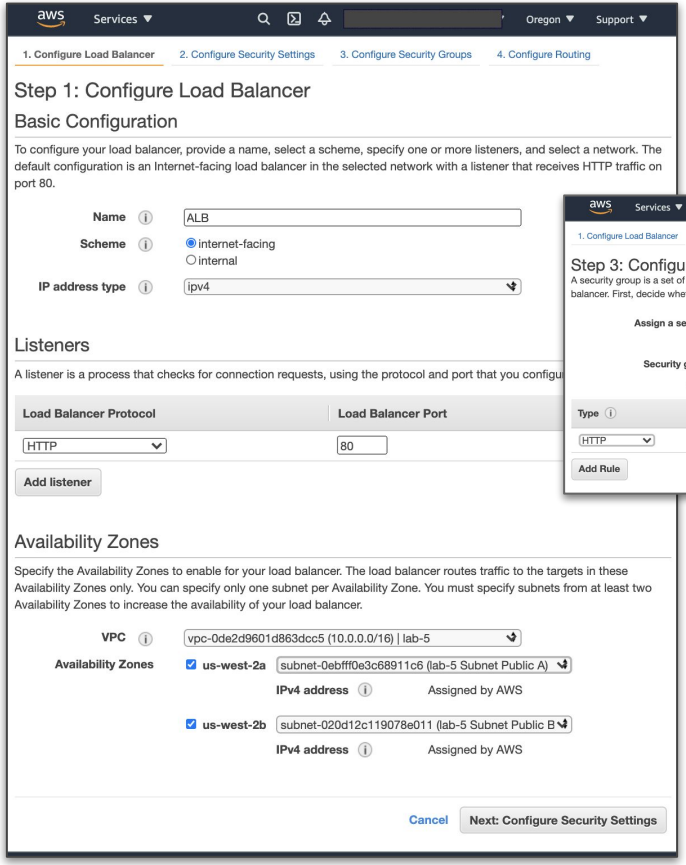
user_data_gameday.sh

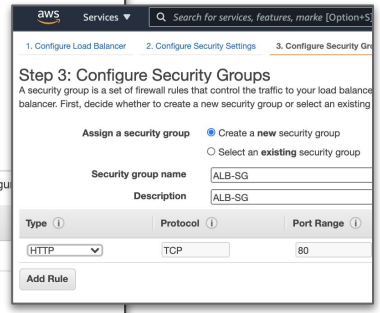
Unicorn.Rentals

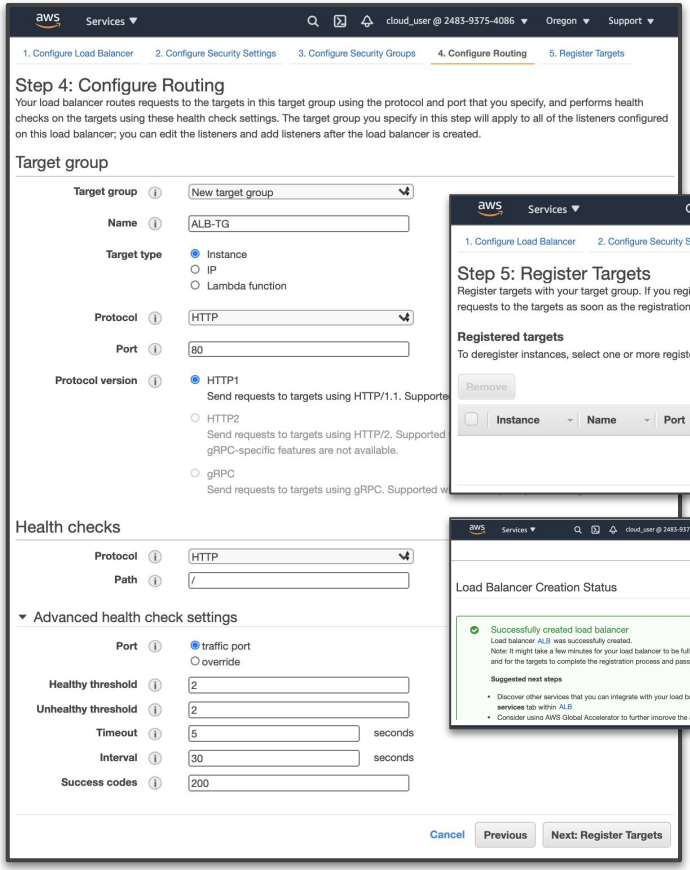
ALB & ASG

Application Load Balancers
Auto Scaling Group

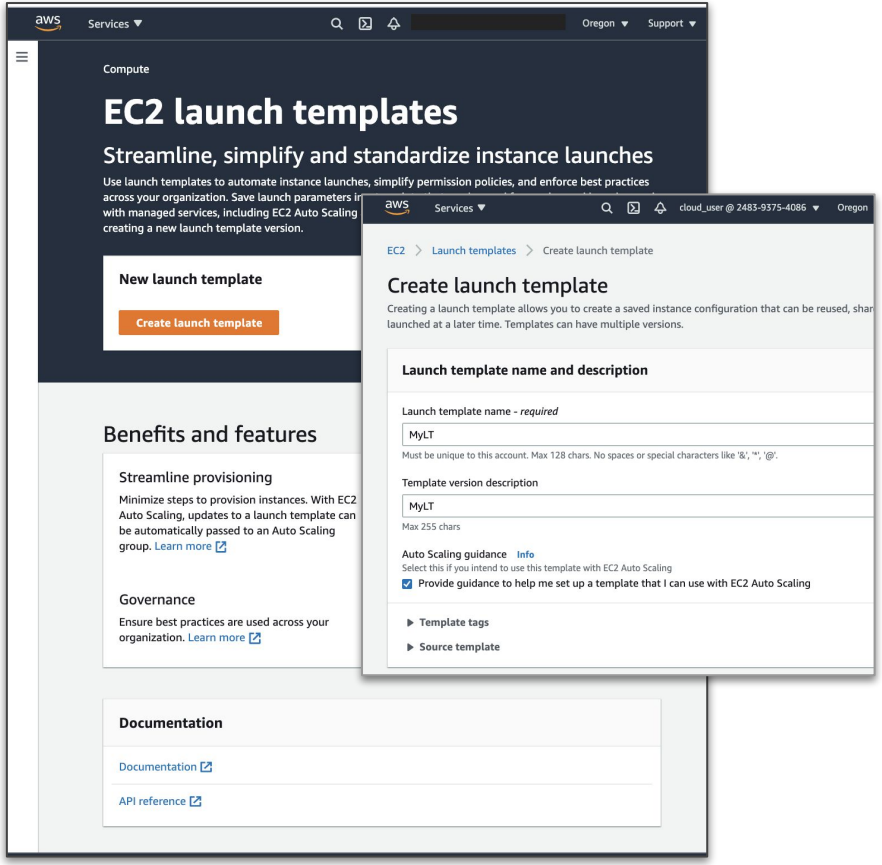
ALB - Target Group

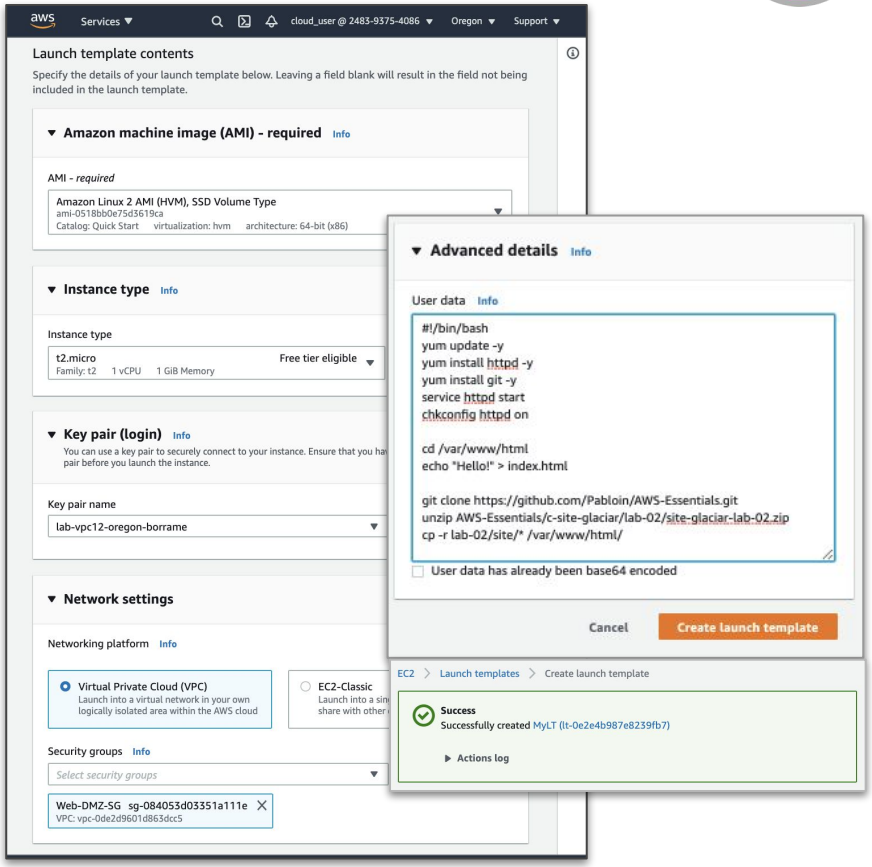




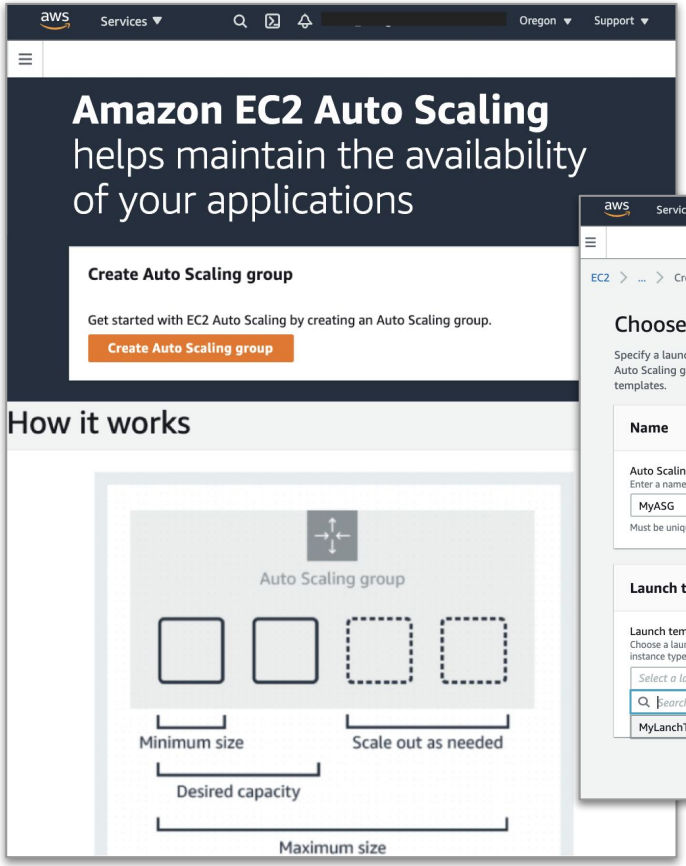


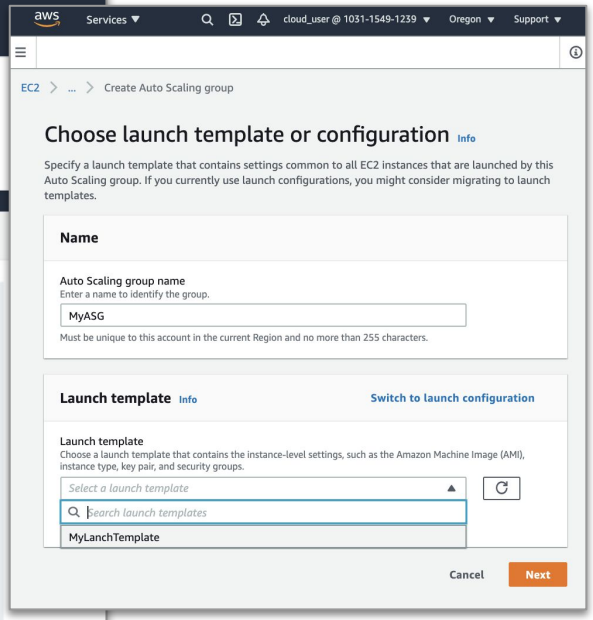
Launch Template

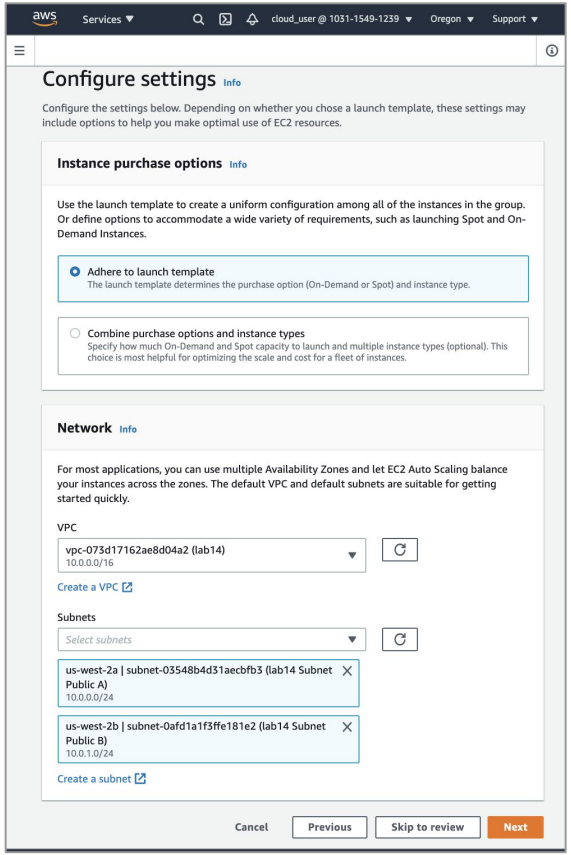




Auto Scaling Group







Auto Scaling Group

Configure advanced options Info

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - *optional* Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- No load balancer
  Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer
  Choose from your existing load balancers.
- Attach to a new load balancer
  Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

- Choose from your load balancer target groups
  This option allows you to attach Application, Network, or Gateway Load Balancers.
- Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

MyTG | HTTP
Application Load Balancer: MyALB

Health checks - *optional*

Health check type Info
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☑ EC2 ☐ ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300 seconds

Additional settings - *optional*

Monitoring Info

☐ Enable group metrics collection within CloudWatch

Cancel | Previous | Skip to review | Next

Configure group size and scaling policies Info

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

Group size - *optional* Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
2

Minimum capacity
1

Maximum capacity
5

Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

- Target tracking scaling policy
  Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.
- None

Scaling policy name
Target Tracking Policy

Metric type
Average CPU utilization

Target value
30

Instances need
300 seconds warm up before including in metric

☐ Disable scale in to create only a scale-out policy

Instance scale-in protection - *optional*

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

Cancel | Previous | Skip to review | Next

Muchos Éxitos!!!