

CASO PRÁCTICO 2

Tarea 1 (50 puntos): Creación de la VPC y las subredes.

Para la **creación de la VPC** y las **subredes**, he hecho uso del wizard, el cual me da la opción de generar la VPC, con 2 AZ y 2 subnets, generando el IGW y las tablas de enrutamiento de forma automática.

The screenshot shows the AWS VPC console 'VPC settings' page. On the left, the 'Resources to create' section is set to 'VPC and more'. The 'Name tag auto-generation' is set to 'Auto-generate' with the name 'Case2'. The 'IPv4 CIDR block' is '10.0.0.0/16' (65,536 IPs). The 'IPv6 CIDR block' is set to 'No IPv6 CIDR block'. The 'Tenancy' is 'Default'. The 'Number of Availability Zones (AZs)' is set to '2'. The 'Number of public subnets' is '2' and the 'Number of private subnets' is '2'. The 'Preview' section shows a diagram of the VPC configuration: a VPC named 'Case2-vpc' with two subnets in 'us-east-1a' (public and private) and two subnets in 'us-east-1b' (public and private). It also shows three route tables: 'Case2-rtb-public', 'Case2-rtb-private1-us-east-1a', and 'Case2-rtb-private2-us-east-1b'. On the right, the 'Network connections' section shows 'Case2-igw' and 'Case2-vpc-s3'. Below the preview, there are sections for 'Number of public subnets', 'Number of private subnets', 'Customize subnets CIDR blocks', 'NAT gateways', and 'VPC endpoints'.

Tras la creación, me aseguro de borrar las 2 subnets que no voy a utilizar, además de borrar las tablas de enrutamiento que sobran (al borrar las subnets no necesarias).

The screenshot shows the AWS VPC console 'Subnets (2/10)' and 'Route tables (1/5)' pages. The 'Subnets' page has a table with 10 subnets. The 'Route tables' page has a table with 5 route tables.

Name	Subnet ID
Case2-subnet-public2-us-east-1b	subnet-0ec4554d009192c7a
Case2-subnet-public1-us-east-1a	subnet-01fcee584574e0483
Case2-subnet-private2-us-east-1b	subnet-01589d79cb9e05dcb
Case2-subnet-private1-us-east-1a	subnet-003e65a2d14d87608
-	subnet-07ef83b4c64a46e27
-	subnet-09251b5ce02c63710
-	subnet-082a7af5dc4b0a630

Name	Route table ID
-	rtb-026818ae87b900895
Case2-rtb-public	rtb-03b0fcd001dad3aa
Case2-rtb-private2-us-east-1b	rtb-0dbf0f4c29d442d5a
Case2-rtb-private1-us-east-1a	rtb-024d5ebbceaf44622
-	rtb-06e3fdd4409af084a

Así quedarían las **tablas de enrutamiento** de la subnet pública y de la privada:

for services, features, blogs, docs, and more

Options+5

N. Virginia

vocalbox/user2189442:enumez-ng@student.42maga.com @ 6582-2017...

VPC > Route tables > rtb-0dbf0f4c29d442d5a

rtb-0dbf0f4c29d442d5a / Case2-rtb-private2-us-east-1b

Actions ▾

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

✕

Details info

Route table ID rtb-0dbf0f4c29d442d5a	Main No	Explicit subnet associations subnet-01589d79cb9e05dcb / Case2-subnet-private2-us-east-1b	Edge associations -
VPC vpc-0b72d126089b56a4d Case2-vpc	Owner ID 658220173762		

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (1)

Edit routes

Filter routes

Both ▾

< 1 > ⓘ

Destination ▾	Target ▾	Status ▾	Propagated ▾
10.0.0.0/16	local	Active	No

for services, features, blogs, docs, and more

Options+5

N. Virginia

vocalbox/user2189442:enumez-ng@student.42maga.com @ 6582-2017...

VPC > Route tables > rtb-03b0fcdad01dad3aa

rtb-03b0fcdad01dad3aa / Case2-rtb-public

Actions ▾

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

✕

Details info

Route table ID rtb-03b0fcdad01dad3aa	Main No	Explicit subnet associations subnet-01fcee584574e0483 / Case2-subnet-public1-us-east-1a	Edge associations -
VPC vpc-0b72d126089b56a4d Case2-vpc	Owner ID 658220173762		

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (2)

Edit routes

Filter routes

Both ▾

< 1 > ⓘ

Destination ▾	Target ▾	Status ▾	Propagated ▾
0.0.0.0/0	igw-0c47a2e4e43d7a64	Active	No
10.0.0.0/16	local	Active	No

Tarea 2 (50 puntos): Creación de las instancias EC2 y los Security Groups.

Defino ambos **grupos de seguridad** para las dos subredes:

Services, features, blogs, docs, and more

[Option+5]

N. Virginia

votcabs/user2189442-nuneez-ng@student.42mataga.com @ 6582-2017...

VPC > Security Groups > sg-0c1ff2f5e6bfb46f4 - SSH-private

sg-0c1ff2f5e6bfb46f4 - SSH-private

Details

Security group name	Security group ID	Description	VPC ID
SSH-private	sg-0c1ff2f5e6bfb46f4	Allow SSH from public subnet	vpc-0b72d126089b56a4d
Owner	Inbound rules count	Outbound rules count	
658220173762	1 Permission entry	1 Permission entry	

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (1/1)

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sg-09bca7ecd5e583f3	IPv4	SSH	TCP	22

Services, features, blogs, docs, and more

[Option+5]

N. Virginia

votcabs/user2189442-nuneez-ng@student.42mataga.com @ 6582-2017...

VPC > Security Groups > sg-00cf069ea5d84828f - SSH-HTTP

sg-00cf069ea5d84828f - SSH-HTTP

Details

Security group name	Security group ID	Description	VPC ID
SSH-HTTP	sg-00cf069ea5d84828f	Allow SSH & HTTP	vpc-0b72d126089b56a4d
Owner	Inbound rules count	Outbound rules count	
658220173762	2 Permission entries	1 Permission entry	

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (2)

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sg-0fca636cad3faefbd	IPv4	HTTP	TCP	80
-	sg-0ba218e8e9990f3b1	IPv4	SSH	TCP	22

El grupo para la subnet privada, con reglas de acceso interno SSH desde las IPs de la subnet pública, y el grupo para la subnet pública, con reglas de acceso SSH y HTTP abierto a todas las IPs entrantes.

Una vez creado todo lo anterior, abro **CLI** para lanzar las **dos instancias**:

```
aws
Services
N. Virgi
voclabs/user2189442=enunez-n@student.42malaga.com @ 65

AWS CloudShell
Actions

us-east-1

#!/bin/bash
##EC2 instance in public subnet with webserver
aws ec2 run-instances \
  --image-id ami-09d3b3274b6c5d4aa \
  --instance-type t2.micro \
  --count 1 \
  --key-name vockey \
  --security-group-id sg-00cf069ea5d84828f \
  --subnet-id subnet-01fcee584574e0483 \
  --user-data file://userdata-web.sh \
  --associate-public-ip-address \
  --tag-specifications 'ResourceType=instance, Tags=[{Key=Name,Value=public-web-server}]' \
  && \
##EC2 instance in private subnet
aws ec2 run-instances \
  --image-id ami-09d3b3274b6c5d4aa \
  --instance-type t2.micro \
  --count 1 \
  --key-name vockey \
  --security-group-id sg-0c1ff2f5e6bfb46f4 \
  --subnet-id subnet-01589d79cb9e05dcb \
  --tag-specifications 'ResourceType=instance, Tags=[{Key=Name,Value=private-server}]'
```

El script se compone de las siguientes partes:

- **aws ec2 run-instances:** comando para lanzar instancias ec2 al que le doy los siguientes flags con los detalles de cada instancia:
 - **--image-id** → ID del AMI a montar.
 - **--instance-type** → el tipo de procesador de la instancia.
 - **--count** → cantidad de instancias a montar.
 - **--key-name** → nombre de la llave de encriptación para el acceso a la instancia.
 - **--security-group-id** → ID del grupo de seguridad asignado a la instancia.
 - **--subnet-id** → ID de la subnet donde será montada la instancia.
 - **--user-data** → script (en este caso archivo con script) que será lanzado al inicio del primer montaje de la instancia.
 - **--associate-public-ip-address** → indica que se le asigne una IP pública a la instancia (en caso de necesitar ser accesible desde internet).
 - **--tag-specifications** → flag con el que poder asignar diferentes tags a la instancia.

El script genera ambas instancias, cada una con las IDs de subnet y SG apropiadas.


















```
us-east-1
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-09d3b32746c5d4aa",
      "InstanceId": "i-0af77868a1df48e2e",
      "InstanceType": "t2.micro",
      "KeyName": "vockey",
      "LaunchTime": "2022-10-20T21:32:58+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1b",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-10-0-144-9.ec2.internal",
      "PrivateIpAddress": "10.0.144.9",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-01589d79c9e85dcb",
      "VpcId": "vpc-0b72d126889b56a4d",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "45af9771-c1e2-4dc4-bf3a-5b64a47ff5cb",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2022-10-20T21:32:58+00:00",
            "AttachmentId": "eni-attach-835773f5f199182a9",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching",
            "NetworkCardIndex": 0
          },
          "Description": "",
          "Groups": [
            {
              "GroupName": "SSH-private",
              "GroupId": "sg-0c1ff2f5e68fb46f4"
            }
          ]
        }
      ]
    }
  ]
}
```

```
us-east-1
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-09d3b32746c5d4aa",
      "InstanceId": "i-0af77868a1df48e2e",
      "InstanceType": "t2.micro",
      "KeyName": "vockey",
      "LaunchTime": "2022-10-20T21:32:34+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1a",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-10-0-1-36.ec2.internal",
      "PrivateIpAddress": "10.0.1.36",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-d1fee584574e0483",
      "VpcId": "vpc-0b72d126889b56a4d",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "1a53f953-bbb9-4cf6-g94d-8247bfb7a57",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2022-10-20T21:32:34+00:00",
            "AttachmentId": "eni-attach-0f406180a0e3eb345",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching",
            "NetworkCardIndex": 0
          },
          "Description": "",
          "Groups": [
            {
              "GroupName": "SSH-HTTP",
              "GroupId": "sg-00cf069ea5d84828f"
            }
          ]
        }
      ]
    }
  ]
}
```

Instancias en estado activo:

Instances (2) Info							
Find instance by attribute or tag (case-sensitive)							
Instance state = running		Clear filters					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	public-web-server	i-0af77868a1df48e2e	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
<input type="checkbox"/>	private-server	i-0c7d10ad065e114ab	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b

Detalle de las dos instancias:

Instance summary for i-0c7d10ad065e114ab (private-server) Info			Instance summary for i-0af77868a1df48e2e (public-web-server) Info		
Updated less than a minute ago			Updated less than a minute ago		
Instance ID  i-0c7d10ad065e114ab (private-server)	Public IPv4 address -	Private IPv4 addresses  10.0.144.9	Instance ID  i-0af77868a1df48e2e (public-web-server)	Public IPv4 address  54.89.182.159 open address	Private IPv4 addresses  10.0.1.36
IPv6 address -	Instance state  Running	Public IPv4 DNS -	IPv6 address -	Instance state  Running	Public IPv4 DNS  ec2-54-89-182-159.compute-1.amazonaws.com open address
Hostname type IP name: ip-10-0-144-9.ec2.internal	Private IP DNS name (IPv4 only)  ip-10-0-144-9.ec2.internal		Hostname type IP name: ip-10-0-1-36.ec2.internal	Private IP DNS name (IPv4 only)  ip-10-0-1-36.ec2.internal	
Answer private resource DNS name -	Instance type t2.micro	Elastic IP addresses -	Answer private resource DNS name -	Instance type t2.micro	Elastic IP addresses -
Auto-assigned IP address -	VPC ID  vpc-0b72d126089b56a4d (Case2-vpc) Link	AWS Compute Optimizer finding  Opt-in to AWS Compute Optimizer for recommendations. Learn more	Auto-assigned IP address  54.89.182.159 [Public IP]	VPC ID  vpc-0b72d126089b56a4d (Case2-vpc) Link	AWS Compute Optimizer finding  Opt-in to AWS Compute Optimizer for recommendations. Learn more
IAM Role -	Subnet ID  subnet-01589d79cb9e05dcb (Case2-subnet-private-us-east-1b) Link	Auto Scaling Group name -	IAM Role -	Subnet ID  subnet-01fcee584574e0483 (Case2-subnet-public-us-east-1a) Link	Auto Scaling Group name -

La instancia en la subnet pública tiene instalado un web server, por lo que podemos acceder a ella desde internet debido a que dispone de una IP pública:

← → ↻ ⚠ Not Secure | 54.89.182.159

Hello AWS, this is a simple web server created by enunez-n!

Del mismo modo, podemos acceder mediante SSH a esta subnet pública con su IP a la que le hemos dado derechos mediante su SG:

```
→ aws chmod 400 labsuser.pem
→ aws ssh -i labsuser.pem ec2-user@54.89.182.159
The authenticity of host '54.89.182.159 (54.89.182.159)' can't be established.
ECDSA key fingerprint is SHA256:p1HodwQjn0rwwkuEc4nPvOB1Vtl1As0767WqEj0LJb0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.89.182.159' (ECDSA) to the list of known hosts.
```

```
__|__|_ )
_| ( /   Amazon Linux 2 AMI
___|\___|___|
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-36 ~]$ █
```

Compruebo que se puede acceder a la subnet privada desde la subnet pública mediante SSH y no existe ese acceso desde IPs externas a la subnet pública:

```
→ aws chmod 400 labsuser.pem
→ aws ssh -i labsuser.pem ec2-user@54.89.182.159
The authenticity of host '54.89.182.159 (54.89.182.159)' can't be established.
ECDSA key fingerprint is SHA256:p1HodwQjn0rwwkuEc4nPv0B1Vt11As0767WqEj0LJb0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.89.182.159' (ECDSA) to the list of known hosts.
```

```
__| __|_ )
_| (    /  Amazon Linux 2 AMI
___|\___|___|
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-36 ~]$ ls
[ec2-user@ip-10-0-1-36 ~]$ vim labsuser.pem
[ec2-user@ip-10-0-1-36 ~]$ chmod 400 labsuser.pem
[ec2-user@ip-10-0-1-36 ~]$ ssh -i labsuser.pem ec2-user@10.0.144.9
The authenticity of host '10.0.144.9 (10.0.144.9)' can't be established.
ECDSA key fingerprint is SHA256:YD/dMUFCsRu5TAGLhYaw4PnsmnCmvM53UInVvewBX2I.
ECDSA key fingerprint is MD5:a6:53:0b:ef:be:95:89:91:ab:1d:21:75:bc:d4:08:28.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.144.9' (ECDSA) to the list of known hosts.
```

```
__| __|_ )
_| (    /  Amazon Linux 2 AMI
___|\___|___|
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-144-9 ~]$ █
```

```
[ec2-user@ip-10-0-144-9 ~]$ exit
logout
Connection to 10.0.144.9 closed.
[ec2-user@ip-10-0-1-36 ~]$ exit
logout
Connection to 54.89.182.159 closed.
→ aws ssh -i labsuser.pem ec2-user@10.0.144.9
ssh: connect to host 10.0.144.9 port 22: No route to host
→ aws █
```