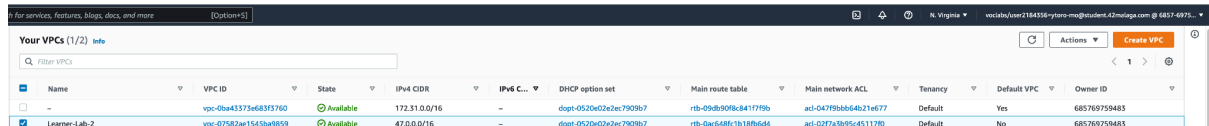


Practical case 1

Student: Yago Toro Molina (ytoro-mo).

E-Mail: ytoro-mo@student.42malaga.com

Task 1



The screenshot shows the AWS VPC console with a table of VPCs. The table has columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP option set, Main route table, Main network ACL, Tenancy, Default VPC, and Owner ID. Two VPCs are listed: 'vpc-0ba43373e683f3760' and 'vpc-07582ae1545ba9859'. The second VPC is selected and highlighted in blue.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL	Tenancy	Default VPC	Owner ID
-	vpc-0ba43373e683f3760	Available	172.31.0.0/16	-	default-0520e02a2ec7909b7	rtb-09b5095c841779b0	acl-047f9b0b546421e477	Default	Yes	685769759483
Learner-Lab-2	vpc-07582ae1545ba9859	Available	47.0.0.0/16	-	default-0520e02a2ec7909b7	rtb-0ac548fc1b18b6d4	acl-c27fa3b95c45117d0	Default	No	685769759483

First we create a new VPC and assign it an IPv4 range (47.0.0.0/16).

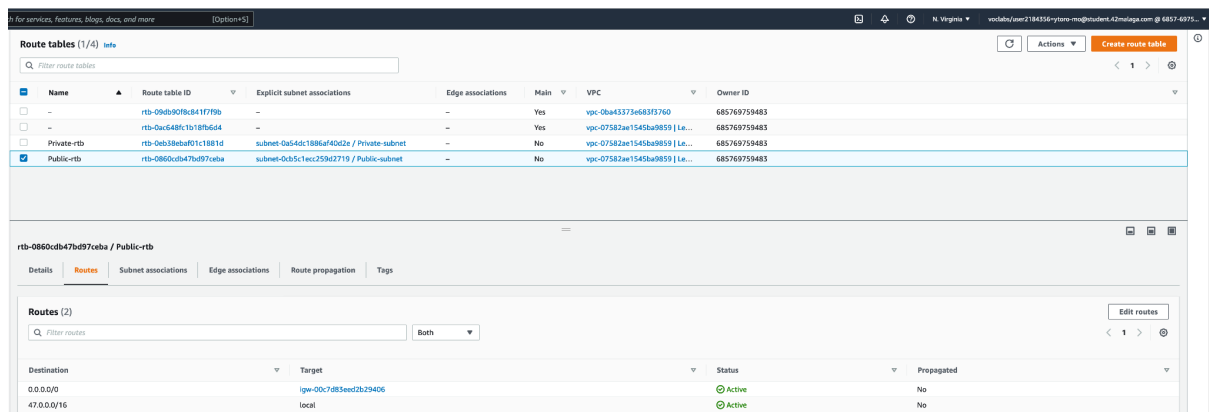


The screenshot shows the AWS Subnets console with a table of subnets. The table has columns for Name, Subnet ID, State, VPC, IPv4 CIDR, IPv6 CIDR, Available IPv4 addresses, Availability Zone, Availability Zone ID, and Network border group. Two subnets are listed: 'Public-subnet' and 'Private-subnet', both associated with 'vpc-07582ae1545ba9859'.

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Availability Zone	Availability Zone ID	Network border group
Public-subnet	subnet-0cb5c1ecc259d2719	Available	vpc-07582ae1545ba9859 Learner-Lab-2	47.0.1.0/24	-	251	us-east-1a	use1-az6	us-east-1
Private-subnet	subnet-0a54dc1886af40d2e	Available	vpc-07582ae1545ba9859 Learner-Lab-2	47.0.2.0/24	-	251	us-east-1b	use1-az1	us-east-1

Then we create 2 different subnets, we assign it to the previously created VPC and different availability zones of the VPC region, at last, we assign an IPv4 range for both subnets (47.0.1.0/24 for public-subnet and 47.0.2.0/24 for private-subnet).

Later, we create the route tables for the subnets.



The screenshot shows the AWS Route Tables console with a table of route tables. The table has columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Owner ID. Four route tables are listed: 'rtb-09b0f0f8c841779b', 'rtb-0ac548fc1b18b6d4', 'rtb-0eb3beba01c1881d', and 'rtb-0860db47b097ceba'. The last one is selected and highlighted in blue. Below the table, the details for 'rtb-0860db47b097ceba / Public-rtb' are shown, including a list of routes.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
-	rtb-09b0f0f8c841779b	-	-	Yes	vpc-0ba43373e683f3760	685769759483
-	rtb-0ac548fc1b18b6d4	-	-	Yes	vpc-07582ae1545ba9859 Le...	685769759483
Private-rtb	rtb-0eb3beba01c1881d	subnet-0a54dc1886af40d2e / Private-subnet	-	No	vpc-07582ae1545ba9859 Le...	685769759483
Public-rtb	rtb-0860db47b097ceba	subnet-0cb5c1ecc259d2719 / Public-subnet	-	No	vpc-07582ae1545ba9859 Le...	685769759483

Destination	Target	Status	Propagated
0.0.0.0/0	lgw-00c7d85eend2b29406	Active	No
47.0.0.0/16	local	Active	No

First, we create the route table for the public-subnet, we assign it to our VPC and our public-subnet, and we edit his routes, by default local rule is assigned, but we need include the internet gateway, that we should have created before, this allows to our public-subnet connect to internet. And now our public subnet can connect to the internet and other subnets in the VPC.

for services, features, blogs, docs, and more [Options]

Route tables (1/4) info

Filter route tables

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
-	rtb-09db90f8c4117f9b	-	-	Yes	vpc-0ba43373e683f3760	685769759483
-	rtb-0ac5488f1b18f6d4	-	-	Yes	vpc-07582ae1545ba9859 Learner-Lab-2	685769759483
Private-rtb	rtb-0eb38ebaf01c1881d	subnet-0a54dc1886af40d2e / Private-subnet	-	No	vpc-07582ae1545ba9859 Learner-Lab-2	685769759483
Public-rtb	rtb-0860c0b47ba97ceba	subnet-0cb5c1ecc25d7219 / Public-subnet	-	No	vpc-07582ae1545ba9859 Learner-Lab-2	685769759483

rtb-0eb38ebaf01c1881d / Private-rtb

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Filter routes Both

Destination	Target	Status	Propagated
47.0.0.0/16	local	Active	No

Later, we create the route table for the private-subnet, we assign it to our VPC and our private-subnet, and we don't need to edit his routes, because by default local rule is assigned, and we don't need to connect to internet only to other subnets in the VPC.

At last, we create the security groups that will be applied to the EC2 instances.

for services, features, blogs, docs, and more [Options]

VPC > Security Groups > sg-0ae1e3bdb52f14f71 - WebServerGroup

sg-0ae1e3bdb52f14f71 - WebServerGroup

Details

Security group name WebServerGroup	Security group ID sg-0ae1e3bdb52f14f71	Description Allows SSH and HTTP access	VPC ID vpc-07582ae1545ba9859
Owner 685769759483	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Inbound rules (2)

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sg-0880b3f7f06540587	IPv4	SSH	TCP	22	0.0.0.0/0	Allows SSH access
-	sg-039e2f16d8926cae3	IPv4	HTTP	TCP	80	0.0.0.0/0	Allows HTTP access

First, we create the group that will be applied to the instance in our public-subnet, we assign it to our VPC and create the inbound rules to allow HTTP connections from all internet and SSH connections from all internet and we don't touch the outbound rules.

for services, features, blogs, docs, and more [Options]

VPC > Security Groups > sg-00d2d528ee386afb0 - PrivateSSHGroup

sg-00d2d528ee386afb0 - PrivateSSHGroup

Details

Security group name PrivateSSHGroup	Security group ID sg-00d2d528ee386afb0	Description Allows SSH access	VPC ID vpc-07582ae1545ba9859
Owner 685769759483	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

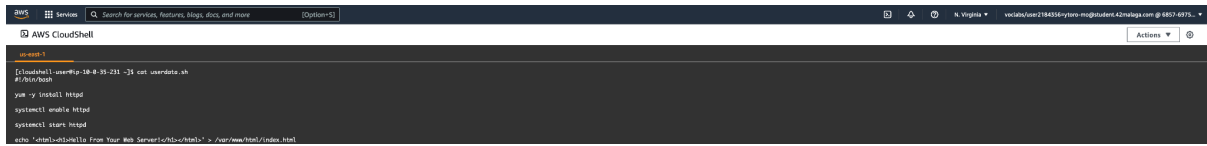
Inbound rules (1/1)

Filter security group rules

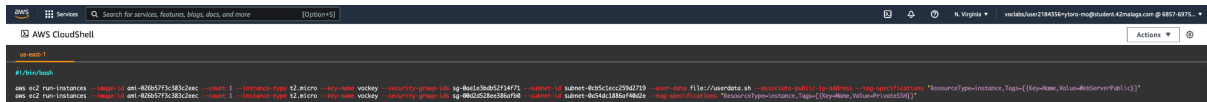
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sg-0ea338efb9f59611	IPv4	SSH	TCP	22	47.0.1.0/24	Allows SSH access from public subnet

And later, we create the group that will be applied to the instance in our private-subnet, we assign it to our VPC and create the inbound rules to allow SSH connections only from the IPv4 range of our public-subnet and nothing more.

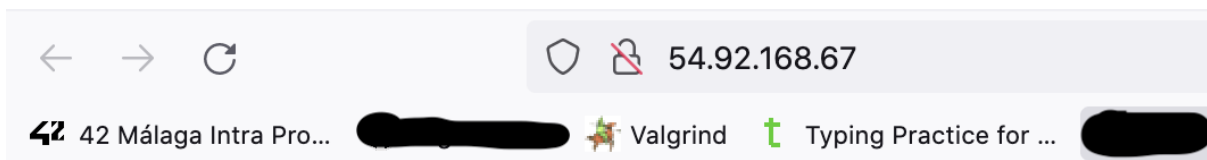
Task 2



First, we create the script that will install Apache service in our public instance.

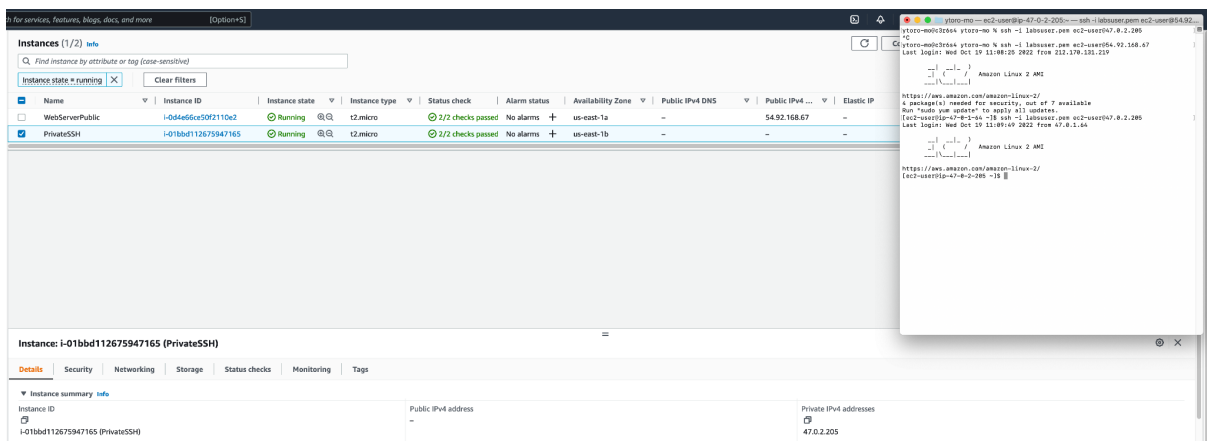


Then, we create the instances using CLI, first public instance, we include in the command line: AMI id, numbers of instances, instance type, certification key, security group id to apply in the instance, subnet where instance will be running, script to install services, a tag name and we indicate that must have associate a public IP. For the private instance it is the same, but without script and public IP.



Hello From Your Web Server!

We connect to Apache service via public IP of the public instance and check that it works.



Finally, we check that we can connect to the public instance via SSH through the internet and we check that we can connect to the private instance via SSH only when we are using an IP associated with the public subnet.