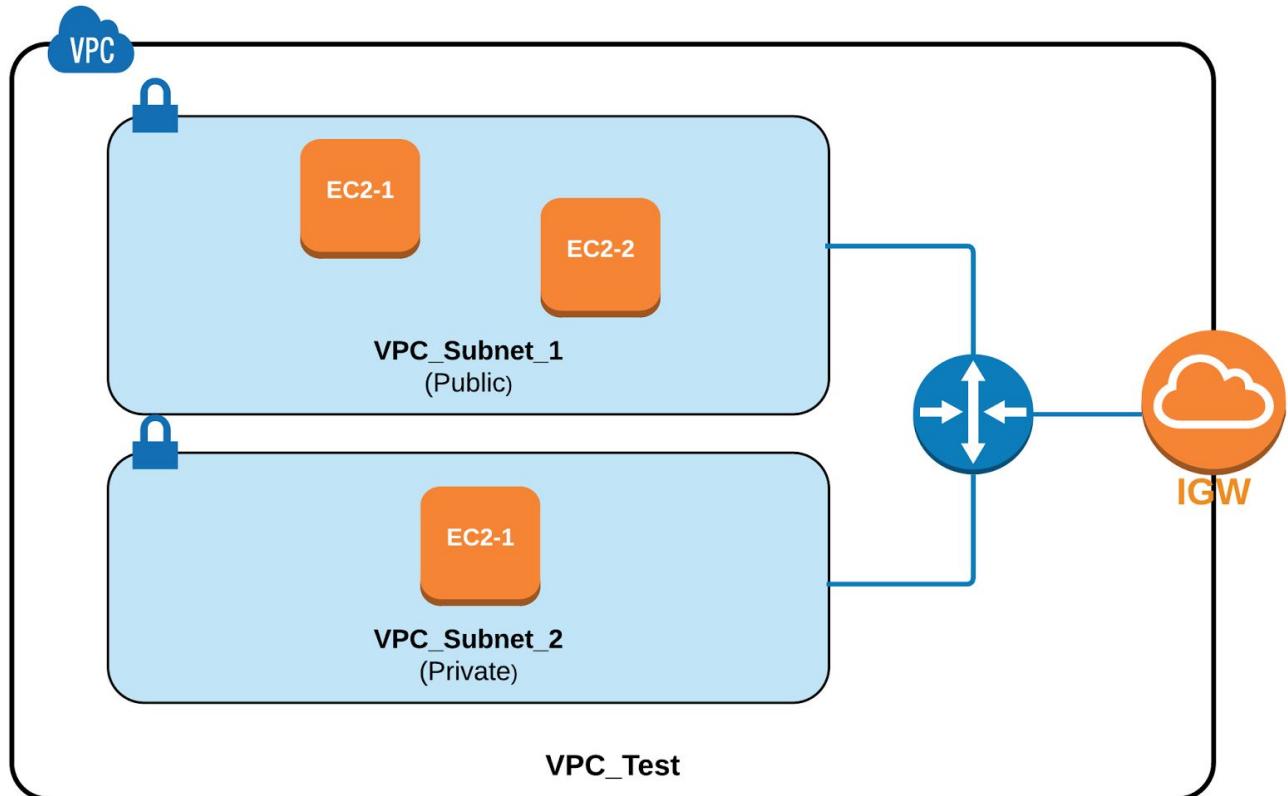


MODULE 4 – LAB EXERCISES

On this lab you will get a practice in Virtual Private Cloud (VPC) with Amazon Linux EC2 Instances management and operation.

During the practice we will build-up the network infrastructure similar to the one shown below:



1. VPC Getting Started: Virtual Network, Subnet and IGW

Open AWS Management Console, find and click “VPC” link to access the service management:

The screenshot shows the AWS Management Console home page. On the left sidebar, under the 'Networking & Content Delivery' section, the 'VPC' link is circled in red. Other items in this section include CloudFront, Route 53, API Gateway, and Direct Connect. The main content area displays various AWS services categorized into Business Productivity, Analytics, Desktop & App Streaming, and Internet of Things.

VPC Dashboard page will be opened:

The screenshot shows the VPC Management Console dashboard. The left sidebar has 'Your VPCs' selected. The main content area is titled 'Resources' and includes sections for 'Start VPC Wizard' and 'Launch EC2 Instances'. It notes that instances will launch in the EU (Frankfurt) region. Below this, it lists Amazon VPC resources in the EU (Frankfurt) region, such as 2 VPCs, 2 Internet Gateways, 0 Egress-only Internet Gateways, 7 Subnets, 2 Route Tables, 2 Network ACLs, 0 Elastic IPs, 0 VPC Peering Connections, 0 Endpoints, 0 Nat Gateways, 20 Security Groups, 3 Running Instances, 0 VPN Connections, 0 Virtual Private Gateways, and 0 Customer Gateways. There is also a 'VPN Connections' section and a note about connecting isolated resources within the AWS cloud to your own.

Click on “Your VPCs” link at left-side menu ribbon to open VPC explorer page:

The screenshot shows the AWS VPC Management console. On the left, there's a sidebar with links like 'Virtual Private Cloud', 'Your VPCs', and 'Subnets'. In the main area, there's a search bar and a table with columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, and DHCP. A large blue button labeled 'Create VPC' is highlighted with a red box. Below the table, it says 'Select a VPC above'.

Click on “Create VPC” button and then specify VPC parameters:

The dialog box has the title 'Create VPC'. It contains the following fields:

- Name tag:** VPC_Test
- IPv4 CIDR block***: 10.0.0.0/24
- IPv6 CIDR block***: Radio buttons for 'No IPv6 CIDR Block' (selected) and 'Amazon provided IPv6 CIDR block'
- Tenancy**: Default

At the bottom right, there are 'Cancel' and 'Yes, Create' buttons. The 'Yes, Create' button is highlighted with a red box.

In the given exercise we are creating VPC named “VPC_Test” and we are assigning CIDR block 10.0.0.0/24 (256 of IP addresses are available).

Click on “Yes, Create” button and find your VPC on the next page, then click on “Subnets” link at the left-side menu ribbon:

The screenshot shows the AWS VPC Management console. The left sidebar lists various VPC components: Virtual Private Cloud, Your VPCs (with 'Subnets' highlighted), Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main area displays the VPC Dashboard, featuring a search bar, a table of existing VPCs, and detailed information for the selected VPC ('VPC_Test'). The 'Summary' tab is active, showing details like VPC ID, State, CIDR ranges, and associated resources.

Click on “Create Subnet” button:

The screenshot shows the AWS VPC Subnets management interface. The top navigation bar includes 'Services', 'Resource Groups', 'Frankfurt', and 'Support'. On the left, a sidebar lists 'Virtual Private Cloud' options: Your VPCs, Subnets (which is selected and highlighted with an orange bar), Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main area has a 'Create Subnet' button highlighted with a red box. Below it is a search bar and a table with columns: Name, Subnet ID, State, VPC, and IPv4 CIDR. A message at the bottom says 'Select a subnet above'.

and then specify subnet parameters:

The screenshot shows the 'Create Subnet' dialog box. It contains fields for 'Name tag' (VPC_Subnet_1), 'VPC' (vpc-d51efcbe | VPC_Test), 'CIDR' (10.0.0.0/24), 'Status' (associated), 'Availability Zone' (No Preference), and 'IPv4 CIDR block' (10.0.0.0/28). At the bottom right are 'Cancel' and 'Yes, Create' buttons, with 'Yes, Create' highlighted with a red box.

In the given exercise we are creating subnet named “VPC_Subnet_1” within “VPC_Test” VPC and we are ordering CIDR block 10.0.0.0/28 (16 of IP addresses are available for subnet).

Click on “Yes, Create” button and find your subnet on next page, then click on “Internet Gateways”

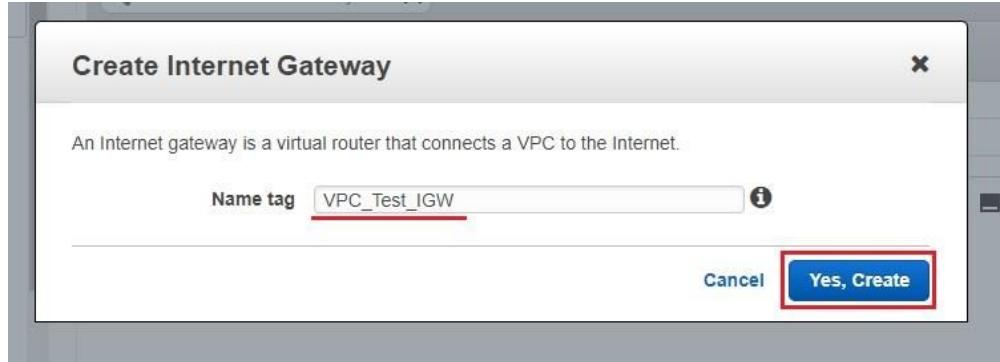
link at left-side menu ribbon:

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Subnets' section, the 'Internet Gateways' link is circled in red and has a red arrow pointing to it. The main content area displays a table for a subnet named 'VPC_Subnet_1'. The table includes columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. The subnet details shown are: Name: VPC_Subnet_1, Subnet ID: subnet-6e05a123, State: available, VPC: vpc-d51efcbe | VPC_Test, IPv4 CIDR: 10.0.0.0/28.

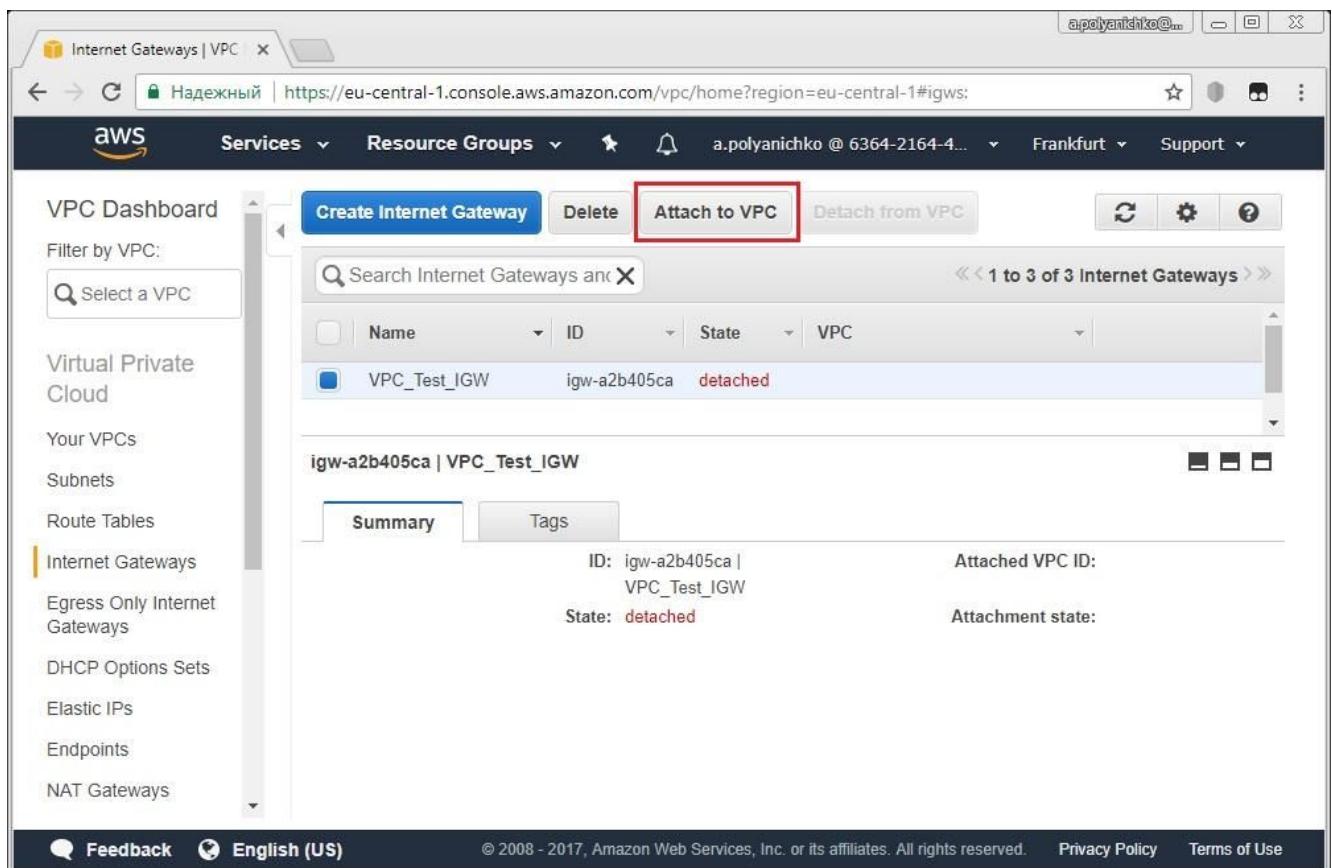
Click on “Create Internet Gateway” button:

The screenshot shows the AWS Internet Gateways page. On the left sidebar, the 'Internet Gateways' link is circled in red and has a red arrow pointing to it. The main content area features a 'Create Internet Gateway' button, which is also highlighted with a red box. Below the button is a search bar and a table for existing Internet Gateways. The table includes columns for Name, ID, State, and VPC. The message 'Select an Internet gateway above' is displayed below the table.

specify IGW name and then click “Yes, Create” button:



On the next page select your IGW and then click on “Attach to VPC” button:

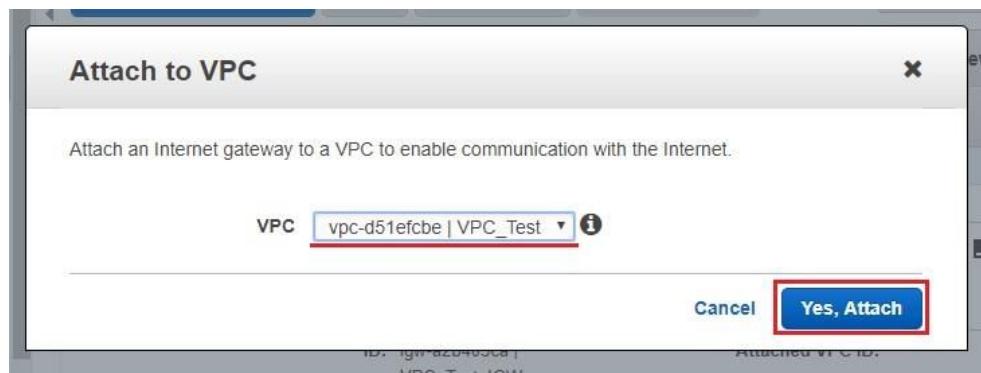


Name	ID	State	VPC
VPC_Test_IGW	igw-a2b405ca	detached	

Summary **Tags**

ID: igw-a2b405ca | VPC_Test_IGW Attached VPC ID:
State: detached Attachment state:

Select your VPC from scroll-down list and then click on “Yes, Attach” button:



Finally you can see your IGW is attached to VPC:

Name	ID	State	VPC
VPC_Test_IGW	igw-a2b405ca	attached	vpc-d51efcbe VPC_Test
igw-a2b405ca	igw-a2b405ca	available	vpc-d51efcbe VPC_Test
igw-a2b405cb	igw-a2b405cb	available	vpc-d51efcbe VPC_Test

Summary

ID: igw-a2b405ca | VPC_Test_IGW
State: attached
Attached VPC ID: vpc-d51efcbe | VPC_Test
Attachment state: available

Revert back to your VPC and check main (default) routing table by clicking on Route table name link:

VPC Dashboard

Filter by VPC:

Virtual Private Cloud

Your VPCs

- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- NAT Gateways

Create VPC Actions

Search VPCs and their properties

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHC
VPC_Test	vpc-d51efcbe	available	10.0.0.0/24		dopt-a049a6c9

vpc-d51efcbe | VPC_Test

Summary CIDR Blocks Flow Logs Tags

VPC ID: vpc-d51efcbe | VPC_Test
 State: available
 IPv4 CIDR: 10.0.0.0/24
 IPv6 CIDR:
 DHCP options set: dopt-a049a6c9
 Route table: rtb-07e6636c

Network ACL: acl-4c2b8627
 Tenancy: Default
 DNS resolution: yes
 DNS hostnames: no

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Check the content of default routing table:

VPC Dashboard

Filter by VPC:

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Create Route Table Delete Route Table Set As Main Table

Search rtb-07e6636c

Name	Route Table ID	Explicitly Associated	Main	VPC
rtb-07e6636c	rtb-07e6636c	0 Subnets	Yes	vpc-d51efcbe VPC_Test

rtb-07e6636c

Summary Routes Subnet Associations Route Propagation Tags

Edit View: All rules

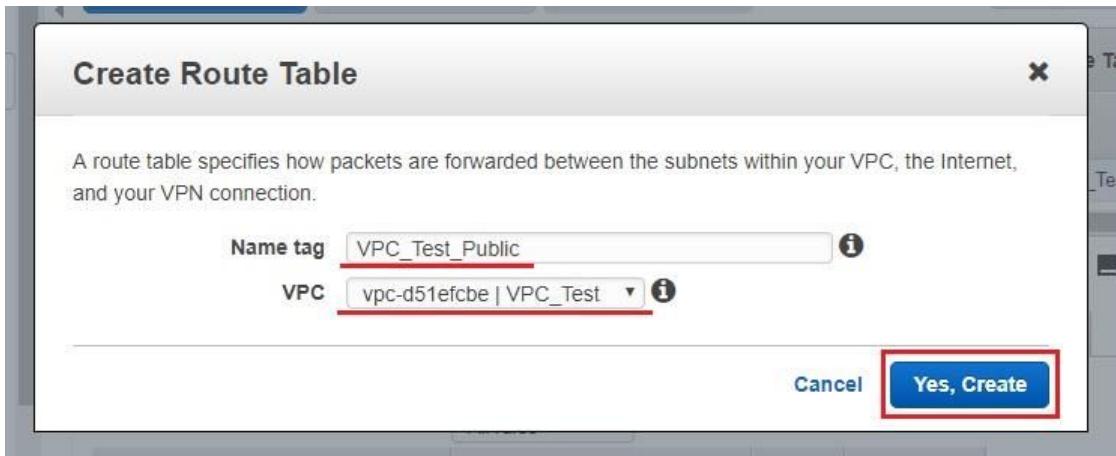
Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

As you can see, the local route only is specified by default for each subnet of our VPC and so we need

to create customer route table with IGW access granted.

Click on “Create Route Table” button at the top: **Create Route Table** and then specify Name tag for new routing table as shown:



Click on “Yes, Create” button, then select your routing table in the list, jump to “Routes” tab and click on “Edit” button below:

Name	Route Table ID	Explicitly Associated	Main	VPC
VPC_Test_Public	rtb-093db862	0 Subnets	No	vpc-d51efcbe VPC_Test
	rtb-9224effb	0 Subnets	Yes	vpc-1d7eb074

Click on “Add another route” button:

The screenshot shows the AWS Route Tables interface. On the left, a sidebar lists options: Route Tables (selected), Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main area has tabs: Summary, Routes (selected), Subnet Associations, Route Propagation, and Tags. Below the tabs, there are 'Cancel' and 'Save' buttons. A message says 'View: All rules'. A table lists routes: Destination (10.0.0.0/24), Target (local), Status (Active), and Propagated (No). At the bottom is a button labeled 'Add another route'.

and specify the route to your IGW as shown below (IGW name is selectable from drop-down list); then click on “Save” button:

The screenshot shows the AWS Route Tables interface for route table 'rtb-093db862 | VPC_Test_Public'. The sidebar includes Peering Connections. The main area shows the 'Routes' tab selected. A red box highlights the 'Save' button. A message says 'View: All rules'. A table lists routes: Destination (10.0.0.0/24), Target (local), Status (Active), and Propagated (No). A new route is added: Destination (0.0.0.0/0), Target (igw-a2b405ca), Status (Active), and Propagated (No). The 'Add another route' button is at the bottom.

Ensure that the route to IGW is present under Routes tab, then switch to “Subnet Associations” tab and click on “Edit” button:

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

NAT Gateways

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and their

Name	Route Table ID	Explicitly Associated	Main	VPC
	rtb-07e6636c	0 Subnets	Yes	vpc-d51efcbe VPC_Test
	rtb-d0ad96b8	0 Subnets	Yes	vpc-545d263c zaiatslaba

Summary Routes **Subnet Associations** Route Propagation Tags

Edit

You do not have any subnet associations.

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Subnet	IPv4 CIDR	IPv6 CIDR
subnet-6e05a123 VPC_Subnet_1	10.0.0.0/28	-

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Associate your VPC_Subnet_1 with the route table by selecting its checkbox and then click on “Save” button:

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

NAT Gateways

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and their

Name	Route Table ID	Explicitly Associated	Main	VPC
	rtb-07e6636c	0 Subnets	Yes	vpc-d51efcbe VPC_Test
	rtb-d0ad96b8	0 Subnets	Yes	vpc-545d263c zaiatslaba

rtb-093db862 | VPC_Test_Public

Summary Routes **Subnet Associations** Route Propagation Tags

Associate **Save**

Subnet	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-6e05a123 VPC_Subnet_1	10.0.0.0/28	-	Main

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Now you may ensure that at the moment your VPC and your Subnet_1 use different routing tables:

Screenshots of the AWS VPC Management Console showing the creation of a VPC and its association with a route table.

VPC Dashboard:

- Filter by VPC: Select a VPC
- Virtual Private Cloud
- Your VPCs (selected)
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints

Create VPC:

- Search: Search VPCs and their proper X
- Results: 1 to 3 of 3 VPCs

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set
VPC_Test	vpc-d51efcbe	available	10.0.0.0/24		dopt-a049a6c9

VPC Test Details:

- VPC ID: vpc-d51efcbe | VPC_Test
- VPC ID: vpc-d51efcbe | VPC_Test
- State: available
- IPv4 CIDR: 10.0.0.0/24
- IPv6 CIDR:
- DHCP options set: dopt-a049a6c9
- Route table: rtb-07e6636c (circled)
- Network ACL: acl-4c2b8627
- Tenancy: Default
- DNS resolution: yes
- DNS hostnames: no

Screenshots of the AWS VPC Management Console showing the creation of a subnet and its association with a route table.

VPC Dashboard:

- Filter by VPC: Select a VPC
- Virtual Private Cloud
- Your VPCs
- Subnets (selected)
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- NAT Gateways

Create Subnet:

- Search: subnet-6e05a123
- Results: 1 to 1 of 1 Subnet

Name	Subnet ID	State	VPC	IPv4 CIDR
VPC_Subnet_1	subnet-6e05a123	available	vpc-d51efcbe VPC_Test	10.0.0.0/28

Subnet-6e05a123 Details:

- Summary
- Route Table (selected)
- Network ACL
- Flow Logs
- Tags

Edit Route Table:

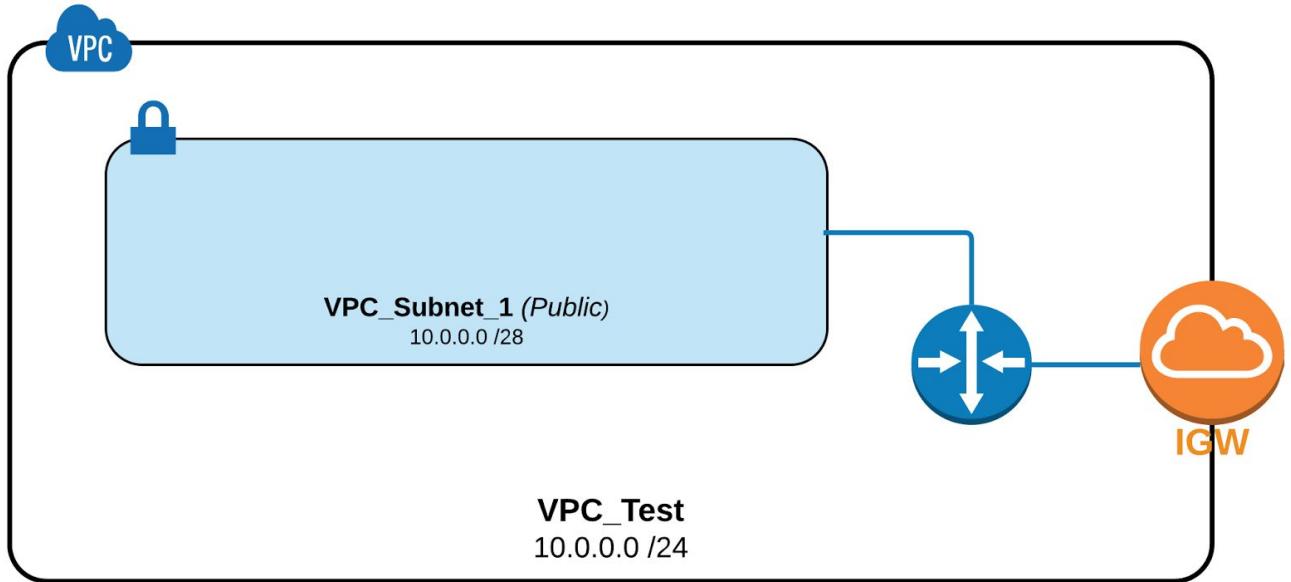
- Route Table: rtb-093db862 | VPC_Test_Public (circled)
- Destination Target

10.0.0.0/24	local
0.0.0.0/0	igw-a2b405ca

Page Footer:

- Feedback
- English (US)
- © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
- Privacy Policy
- Terms of Use

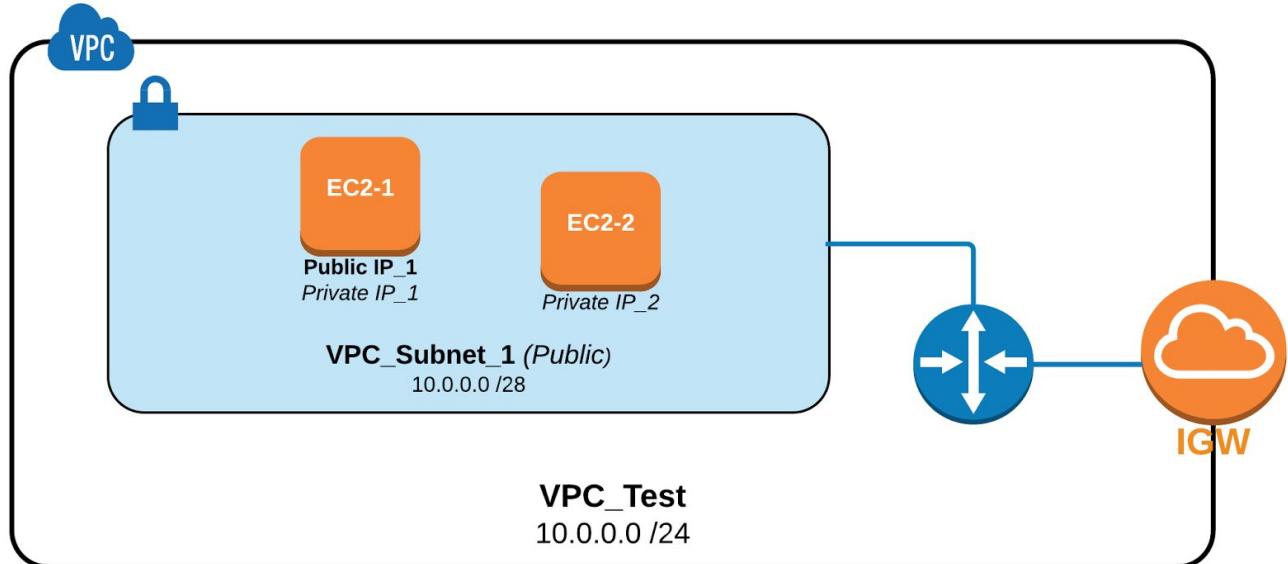
At the moment we have created the basic of our network infrastructure:



and we may add EC2 instances in the subnet.

2. Adding public and private EC2 instances to VPC in public subnet

In this exercise we will add two EC2 instances to our public subnet and grant the public access for one of them for external accessibility:



Switch to EC2 Dashboard and create new Amazon Linux AMI EC2 instance as was described in Module 3 hands-on lab except for the options mentioned below:

Name	Instance ID	Type	Availability Zone	Instance State	Status Check
EC2 Test	i-0a36c4336c545bc59	t2.micro	eu-central-1b	running	2/2 che

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

On “Configure Instance” tab: specify your VPC as Network and Subnet_1 as Subnet as well as enable Auto-assigning of Public IP (the last one is mandatory if you are planning to connect your instance from outside of VPC):

Number of instances: 1

Purchasing option: Request Spot instances

Network: vpc-d51efcbe | VPC_Test

Subnet: subnet-6e05a123 | VPC_Subnet_1 | eu-central-1c

Auto-assign Public IP: Enable

IAM role: None

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Storage

On “Add Tags” tab: set the user-friendly name for instance:

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key	(127 characters maximum)	Value	(255 characters maximum)	Instances	Volumes
Name		VPC_Sub1_Ec2-1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add another tag (Up to 50 tags maximum)					

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

[Feedback](#) [English \(US\)](#) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

On “Configure Security Group” tab: add the rule for all-ICMP traffic (it is mandatory when you want to use ping command for the instance):

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: Create a **new** security group
 Select an **existing** security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	0.0.0.0/0
All ICMP - IPv4	ICMP	0 - 65535	Custom	0.0.0.0/0

[Add Rule](#)

[Cancel](#) [Previous](#) [Review and Launch](#)

[Feedback](#) [English \(US\)](#) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



You may assign existing key prepared on previous lab exercises to secure SSH connection when instance is launching.

Launch your instance and then find it on EC2 Dashboard:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
EC2 Test	i-0a36c4336c545bc59	t2.micro	eu-central-1b	running	2/2 checked
VPC_Sub1_Ec2-1	i-06d1cb1ddf9229b1e	t2.micro	eu-central-1c	running	Initial

Instance: i-06d1cb1ddf9229b1e (VPC_Sub1_Ec2-1) Public IP: 52.57.192.224

Instance ID: i-06d1cb1ddf9229b1e	Public DNS (IPv4): -
Instance state: running	IPv4 Public IP: 52.57.192.224
Instance type: t2.micro	IPv6 IPs: -
Elastic IPs: -	Private DNS: ip-10-0-0-10.eu-central-1.compute.internal
Availability zone: eu-central-1c	Private IPs: 10.0.0.10

Note Private IP and Public IP of your instance than connect to the instance by PuTTY via SSH as it was described in MODULE 3 lab:

```

Login as: ec2-user
Authenticating with public key "imported-openssh-key"
Amazon Linux AMI
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ 

```

So your public EC2 instance is alive and accessible from outside and then you may create one more EC2 instance with private settings.

Repeat all steps as was described here for Amazon Linux AMI EC2 instance creation except two options below.

Disable Auto-assigning of Public IP on “Configure Instance” tab:

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group (i)
Purchasing option	<input checked="" type="checkbox"/> Request Spot instances	
Network	vpc-d51efcbe VPC_Test	Create new VPC
Subnet	subnet-6e05a123 VPC_Subnet_1 eu-central-1c	Create new subnet 10 IP Addresses available
Auto-assign Public IP	Disable	
IAM role	<input type="text" value="None"/>	Create new IAM role

Cancel Previous **Review and Launch** Next: Add Storage

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

And, of course, specify appropriate user-friendly name on “Add Tags” tab:

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(127 characters maximum)	Value	(255 characters maximum)	Instances	Volumes
Name		VPC_Sub1_EC2-2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)



Please don't forget to enable ICMP traffic on “Configure Security Group” tab.

Find your second instance on EC2 Dashboard:

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, and Network & Security. The Instances section is currently selected. In the main pane, there's a search bar with 'Name : EC2' and a results table showing three instances. The first instance, 'VPC_Sub1_EC2-2', is highlighted with a blue square icon. Below the table, a detailed view of this instance is provided, including its Instance ID, State, Type, and various DNS and IP addresses.

Note Private IP address of the instance (public IP must be absent).

Login again to your first (public) EC2-1 instance using PuTTY:

```
R1
login as: ec2-user
Authenticating with public key "importedOpenssh-key"
[ec2-user@ip-10-0-0-10 ~]$
```

Try to ping EC2-2 from EC2-1 using its private IP address:

\$>	[ec2-user@ip-<EC2-1_Privat_IP>~]\$ ping <EC2-2_Privat_IP>
-----	---



Press “Ctrl”+”C” keys to stop output of ping command.

If all configurations were done correctly, you will see the output like this:

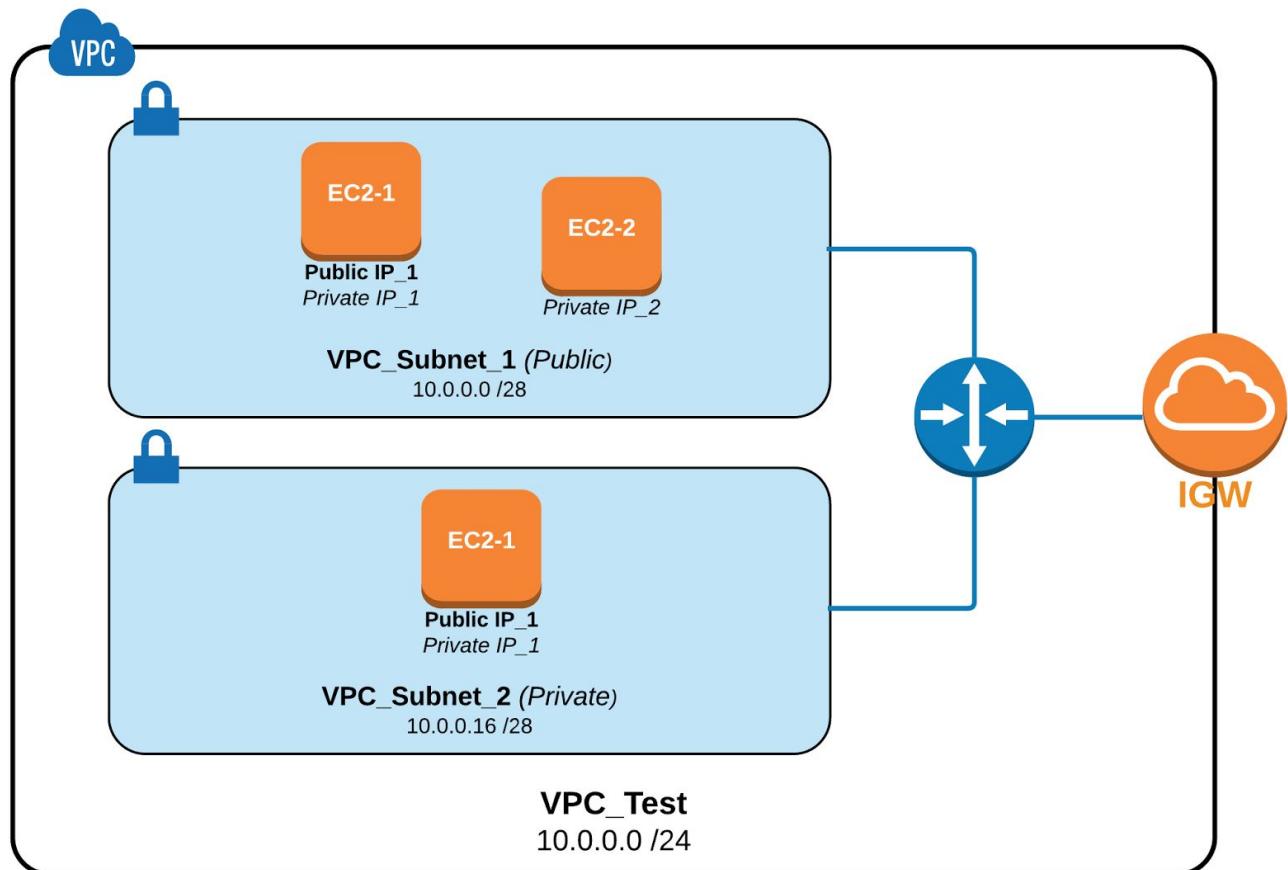
```
R1
Login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last Login: Tue Nov 28 09:28:46 2017 from 31.13.22.89

      _\   _|
      _| (   _/
      _\|_ |__| Amazon Linux AMI

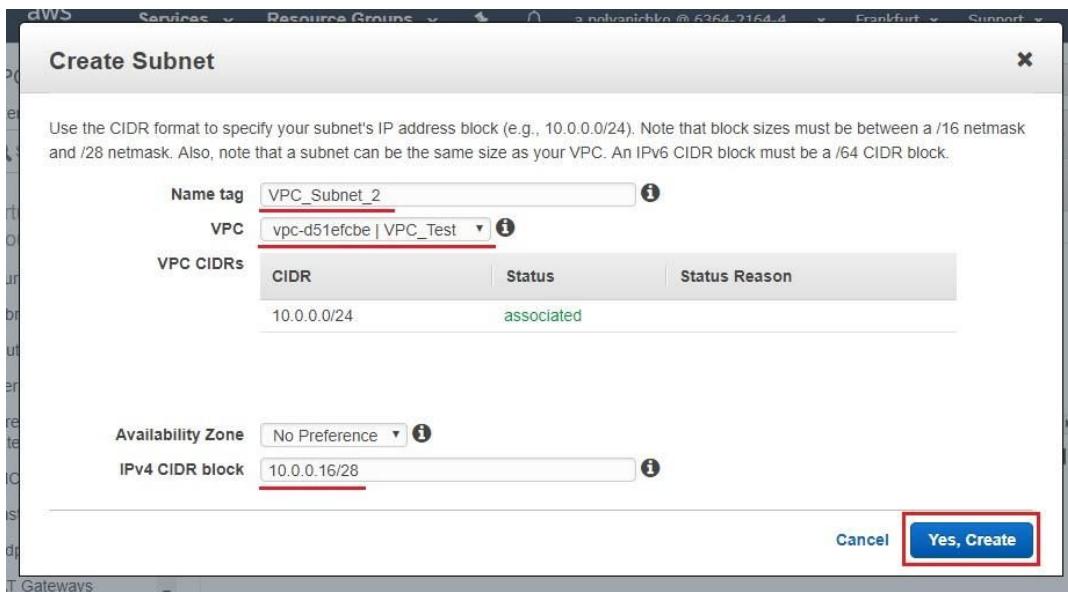
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-10-0-0-10 ~]$
[ec2-user@ip-10-0-0-10 ~]$
[ec2-user@ip-10-0-0-10 ~]$ ping 10.0.0.7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=255 time=0.615 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=255 time=0.413 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=255 time=0.368 ms
64 bytes from 10.0.0.7: icmp_seq=4 ttl=255 time=0.408 ms
64 bytes from 10.0.0.7: icmp_seq=5 ttl=255 time=0.396 ms
64 bytes from 10.0.0.7: icmp_seq=6 ttl=255 time=0.419 ms
64 bytes from 10.0.0.7: icmp_seq=7 ttl=255 time=0.417 ms
64 bytes from 10.0.0.7: icmp_seq=8 ttl=255 time=0.442 ms
64 bytes from 10.0.0.7: icmp_seq=9 ttl=255 time=0.423 ms
64 bytes from 10.0.0.7: icmp_seq=10 ttl=255 time=0.484 ms
64 bytes from 10.0.0.7: icmp_seq=11 ttl=255 time=0.381 ms
64 bytes from 10.0.0.7: icmp_seq=12 ttl=255 time=0.445 ms
^C
--- 10.0.0.7 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11268ms
rtt min/avg/max/mdev = 0.368/0.434/0.615/0.063 ms
[ec2-user@ip-10-0-0-10 ~]$
[ec2-user@ip-10-0-0-10 ~]$
```

3. Adding Private Subnet and EC2 Instance to VPC

In the next exercise, we will supplement the existing network infrastructure with a new Private subnet and add one more EC2 Instance to new subnet therefore our network will look like this:



Start from VPC Dashboard and create new subnet named “VPC_Subnet_2” in your VPC based on CIDR 10.0.0.16/28:



Find the subnet you created in the list of subnets and ensure that it uses default Route table for your VPC:

Name	Subnet ID	State	VPC	IPv4 CIDR
VPC_Subnet_1	subnet-6e05a123	available	vpc-d51efcbe VPC_Test	10.0.0.0/28
VPC_Subnet_2	subnet-999235d4	available	vpc-d51efcbe VPC_Test	10.0.0.16/28

Then create new Amazon Linux AMI EC2 Instance within Subnet_2 and enable Public IP for it (the process is the same as for two previous instances and differences are shown below explicitly):

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot instances

Network: vpc-d51efcbe | VPC_Test

Subnet: subnet-999235d4 | VPC_Subnet_2 | eu-central-1c 11 IP Addresses available

Auto-assign Public IP: Enable

IAM role: None

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Storage

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

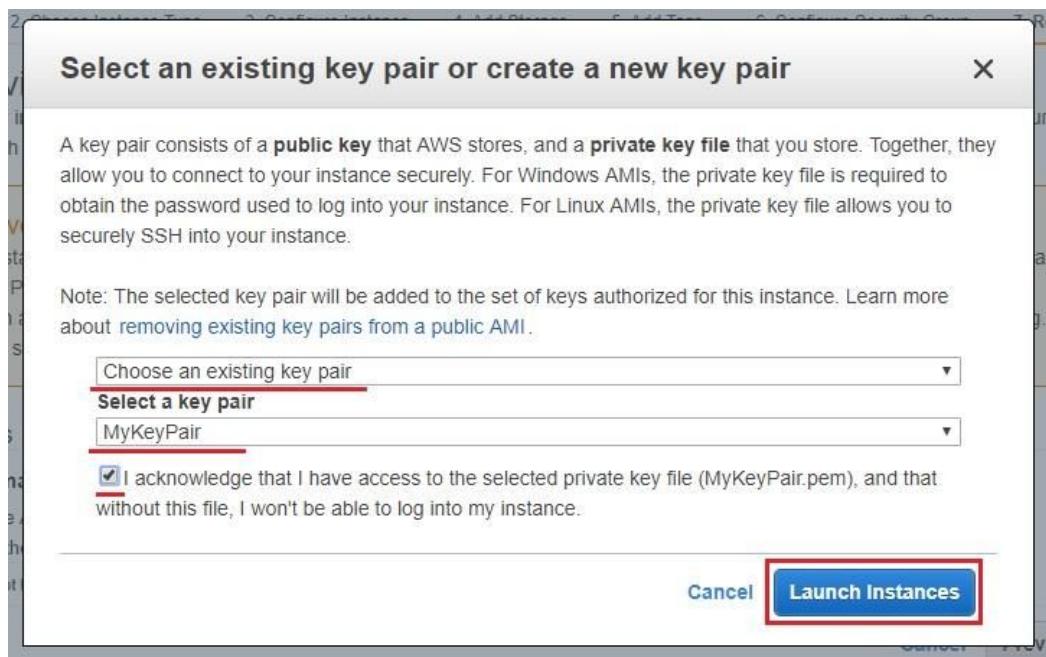
Key	(127 characters maximum)	Value	(255 characters maximum)	Instances	Volumes
Name		VPC_Sub2_EC2-1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Buttons: Cancel, Previous, **Review and Launch**, Next: Configure Security Group



Assign the existing key pair to your instance in launching:



Find your new instance in EC2 Dashboard and note both its Public IP and Private IP:

Name	Instance ID	Type	Zone	Status
VPC_Sub2_EC2-1	i-046d116fb1c6c4082	t2.micro	eu-central-1c	running
VPC_Sub1_EC2-2	i-064801c9be2633d56	t2.micro	eu-central-1c	running
VPC_Sub1_EC2-1	i-06d1cb1dd9229b1e	t2.micro	eu-central-1c	running

Instance: i-046d116fb1c6c4082 (VPC_Sub2_EC2-1) **Public IP: 18.195.87.84**

Description	Status Checks	Monitoring	Tags
Instance ID	i-046d116fb1c6c4082	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	18.195.87.84
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-10-0-0-20.eu-central-1.compute.internal
Availability zone	eu-central-1c	Private IPs	10.0.0.20

Feedback English (US) © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Login to your VPC_Subn1_EC2-1 (the very first) instance via PuTTY SSH using its Public IP:

```

login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Tue Nov 28 11:40:37 2017 from 31.13.22.89

 _|_(_|_) Amazon Linux AMI
__|_\_|_||

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ 

```

then try to ping the second instance in Subnet_1 using its Private IP:

```

[ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ ping 10.0.0.7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=255 time=0.594 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=255 time=0.496 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=255 time=0.411 ms
64 bytes from 10.0.0.7: icmp_seq=4 ttl=255 time=0.446 ms
64 bytes from 10.0.0.7: icmp_seq=5 ttl=255 time=0.474 ms
^C
--- 10.0.0.7 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4090ms
rtt min/avg/max/mdev = 0.411/0.484/0.594/0.063 ms
[ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ 

```

and finally try to ping EC2 instance in Subnet_2 using its Private IP:

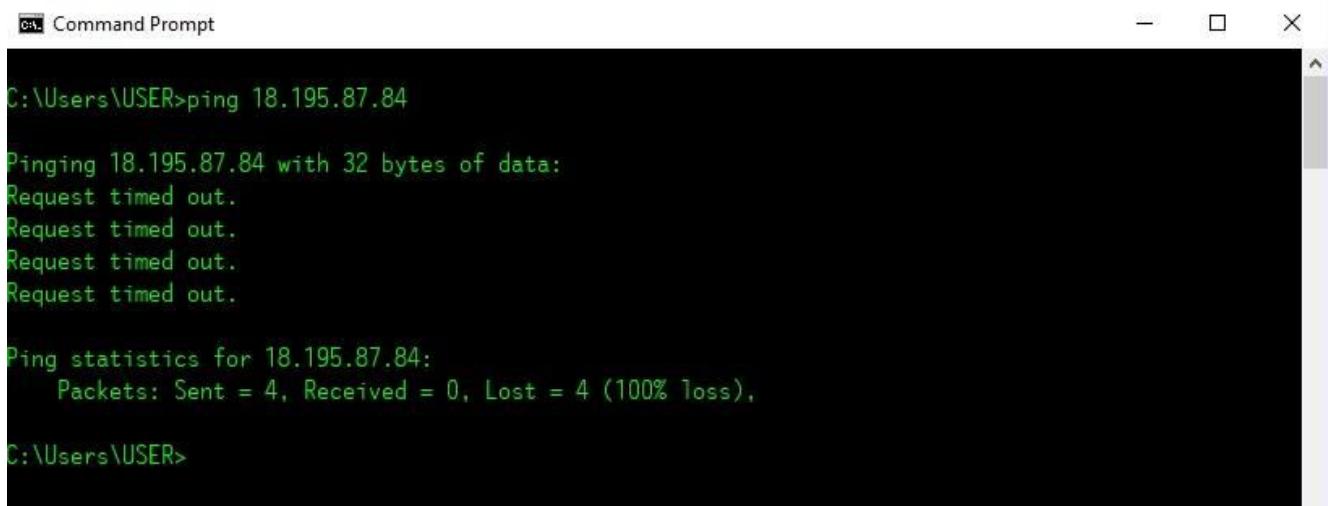
```

[ec2-user@ip-10-0-0-10 ~]$ [ec2-user@ip-10-0-0-10 ~]$ ping 10.0.0.20
PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data.
64 bytes from 10.0.0.20: icmp_seq=1 ttl=255 time=0.355 ms
64 bytes from 10.0.0.20: icmp_seq=2 ttl=255 time=0.416 ms
64 bytes from 10.0.0.20: icmp_seq=3 ttl=255 time=0.398 ms
64 bytes from 10.0.0.20: icmp_seq=4 ttl=255 time=0.367 ms
64 bytes from 10.0.0.20: icmp_seq=5 ttl=255 time=0.408 ms
64 bytes from 10.0.0.20: icmp_seq=6 ttl=255 time=0.368 ms
64 bytes from 10.0.0.20: icmp_seq=7 ttl=255 time=0.458 ms
^C
--- 10.0.0.20 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6144ms
rtt min/avg/max/mdev = 0.355/0.395/0.458/0.040 ms
[ec2-user@ip-10-0-0-10 ~]$ 

```

So, as you can see, all of EC2 instances in both subnets are mutually accessible and therefore they may exchange the traffic one to another.

However, if you will try to connect VPC_Subn2_EC2-1 instance from outside using its Public IP, the connection attempt will be failed (in the example below Windows Command prompt was used to issue ping command):



```
Command Prompt

C:\Users\USER>ping 18.195.87.84

Pinging 18.195.87.84 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 18.195.87.84:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\USER>
```

Finally we can make the following conclusion about our instances and their accessibility in our network environment:

- All of EC2 instances in both subnets are mutually accessible for traffic exchange;
- EC2-1 instance in Subnet_1 has Internet access because it has Public IP assigned and Subnet_1 uses customer's routing table where the route to IGW is present;
- EC2-2 instance in Subnet_1 has not Internet access because it has not Public IP address assigned and it uses only Private IP for internal exchange;
- EC2-1 instance in Subnet_2 has not Internet access although it has Public IP address assigned because Subnet_2 uses VPC default routing table where there is not the route to IGW.