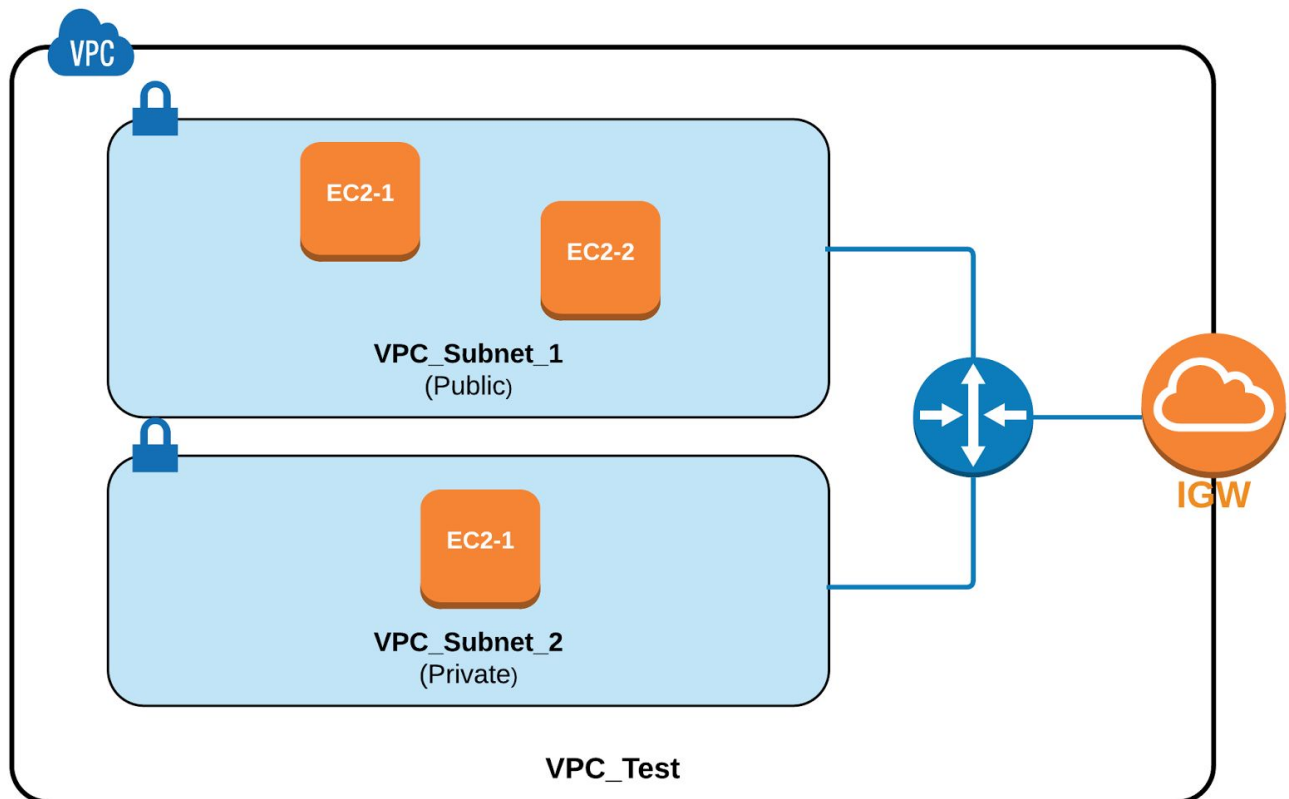


## MODULE 4 – LAB EXERCISES

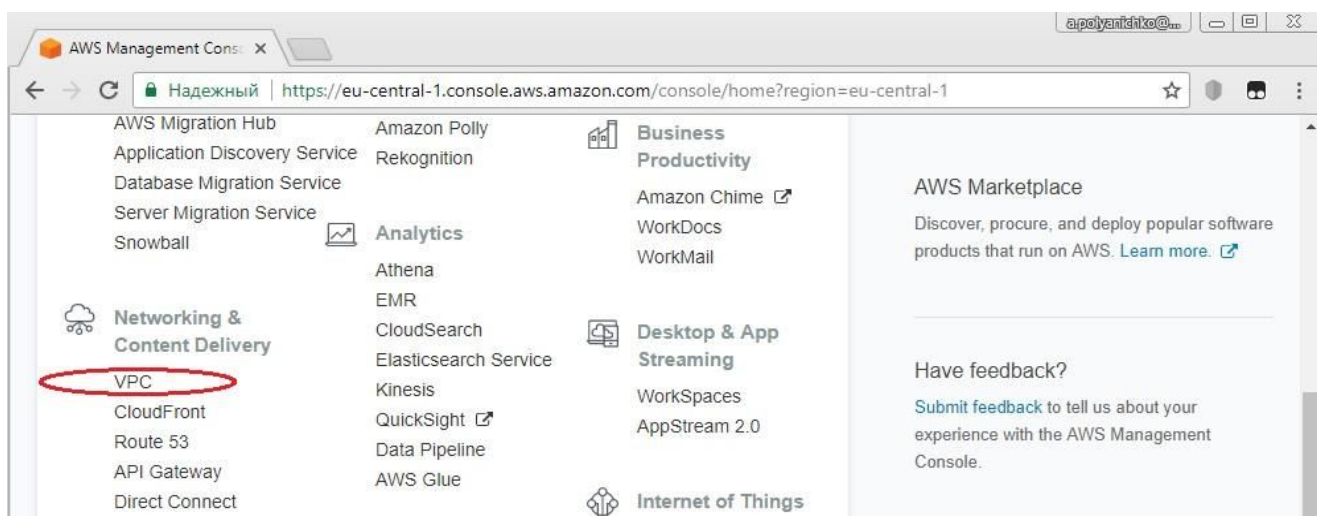
On this lab you will get a practice in Virtual Private Cloud (VPC) with Amazon Linux EC2 Instances management and operation.

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

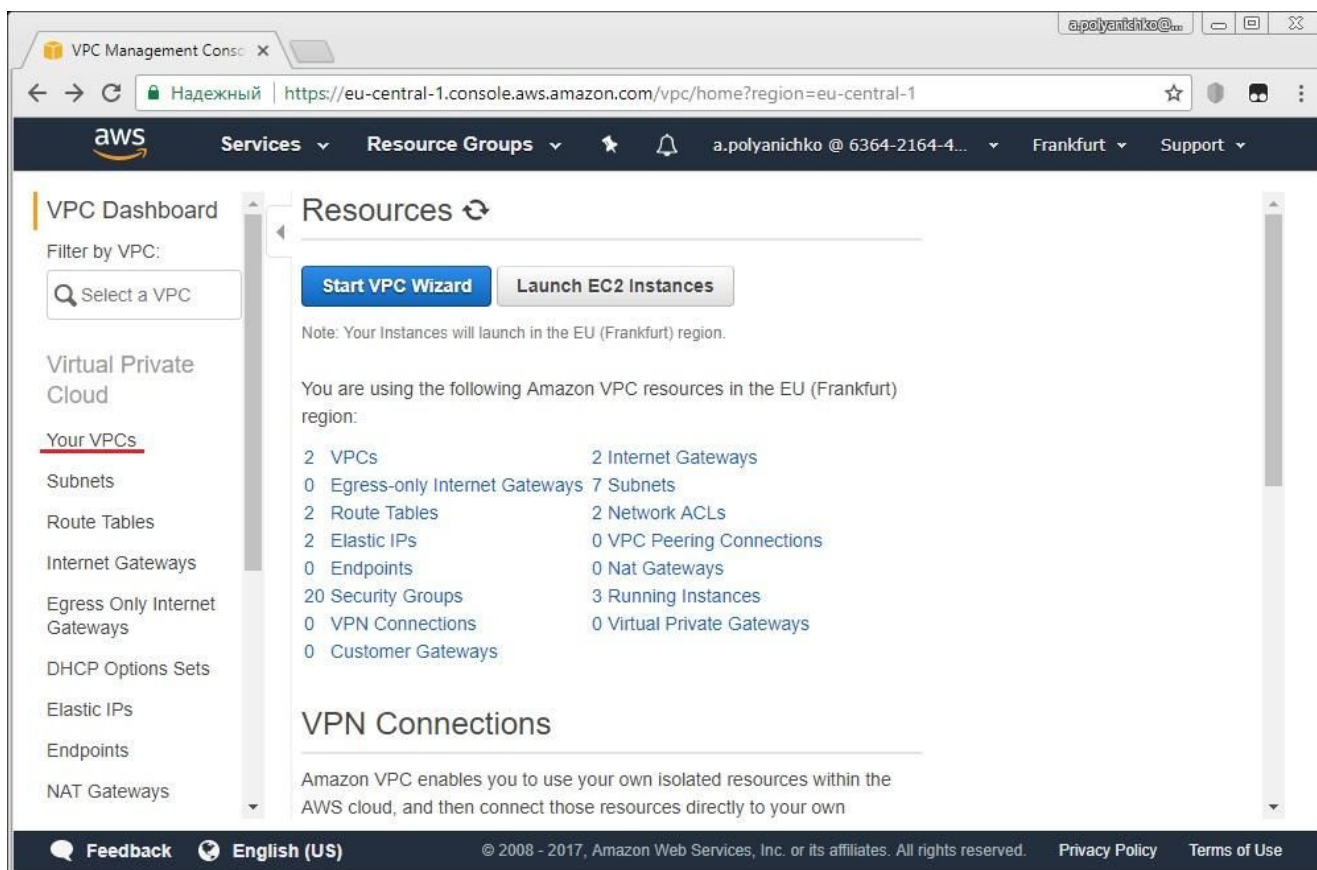


### 1. VPC Getting Started: Virtual Network, Subnet and IGW

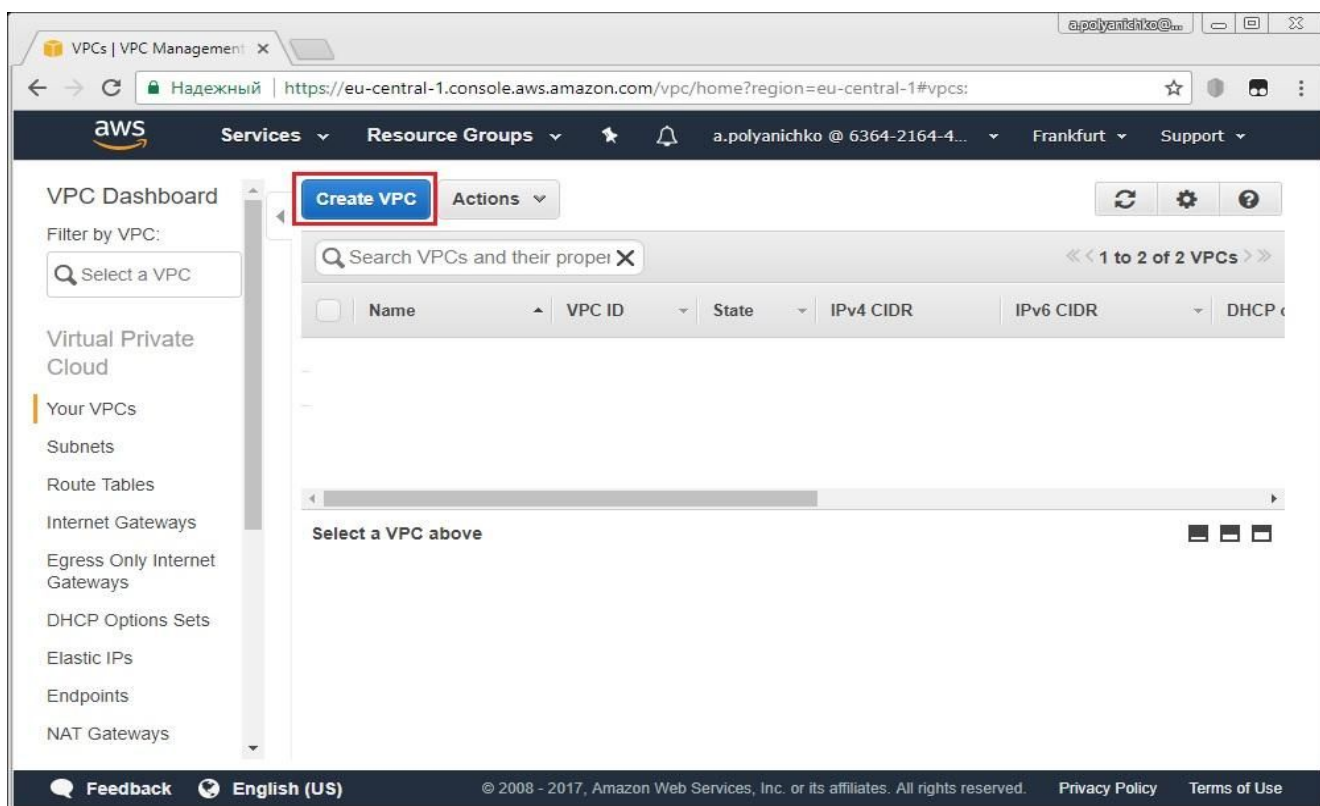
Open AWS Management Console, find and click “VPC” link to access the service management:



VPC Dashboard page will be opened:



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



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

A screenshot of the 'Create VPC' dialog box. It contains a description of VPCs and several input fields. The 'Name tag' field is filled with 'VPC\_Test'. The 'IPv4 CIDR block\*' field is filled with '10.0.0.0/24'. The 'IPv6 CIDR block\*' section has two radio buttons: 'No IPv6 CIDR Block' (selected) and 'Amazon provided IPv6 CIDR block'. The 'Tenancy' dropdown is set to 'Default'. At the bottom right, there are 'Cancel' and 'Yes, Create' buttons, with the 'Yes, Create' button highlighted by a red rectangle.

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. On the left, the 'VPC Dashboard' sidebar lists various resources: Virtual Private Cloud, Your VPCs, Subnets (highlighted with a red circle), Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main area displays a table of VPCs with columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, and DHCP. A single VPC, 'VPC\_Test' (vpc-d51efcbe), is listed with a state of 'available' and an IPv4 CIDR of 10.0.0.0/24. Below the table, the 'vpc-d51efcbe | VPC\_Test' details are shown under the 'Summary' tab. The details include: VPC ID: vpc-d51efcbe | VPC\_Test, State: available, IPv4 CIDR: 10.0.0.0/24, IPv6 CIDR: (empty), DHCP options set: dopt-a049a6c9, Route table: rtb-07e6636c, Network ACL: acl-4c2b8627, Tenancy: Default, DNS resolution: yes, and DNS hostnames: no. The bottom of the console shows the 'Feedback' button, 'English (US)' language selector, and copyright information for Amazon Web Services, Inc. (© 2008 - 2017).

VPCs | VPC Management

Надежный | https://eu-central-1.console.aws.amazon.com/vpc/home?region=eu-central-1#vpcs:

aws Services Resource Groups a.polyanichko @ 6364-2164-4... Frankfurt Support

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 VPC Actions

Search VPCs and their properties

<< 1 to 3 of 3 VPCs >>

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

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

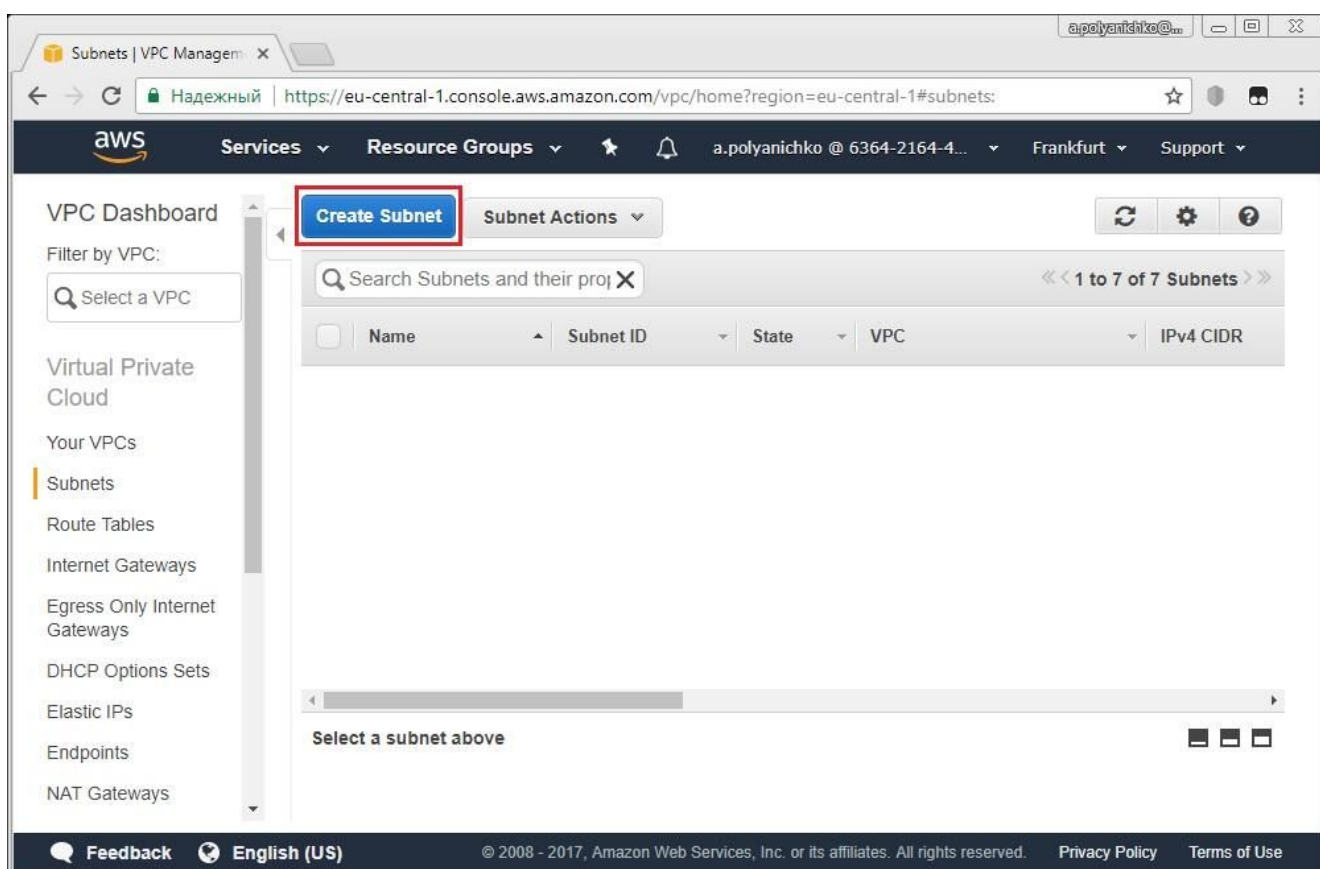
Tenancy: Default

DNS resolution: yes

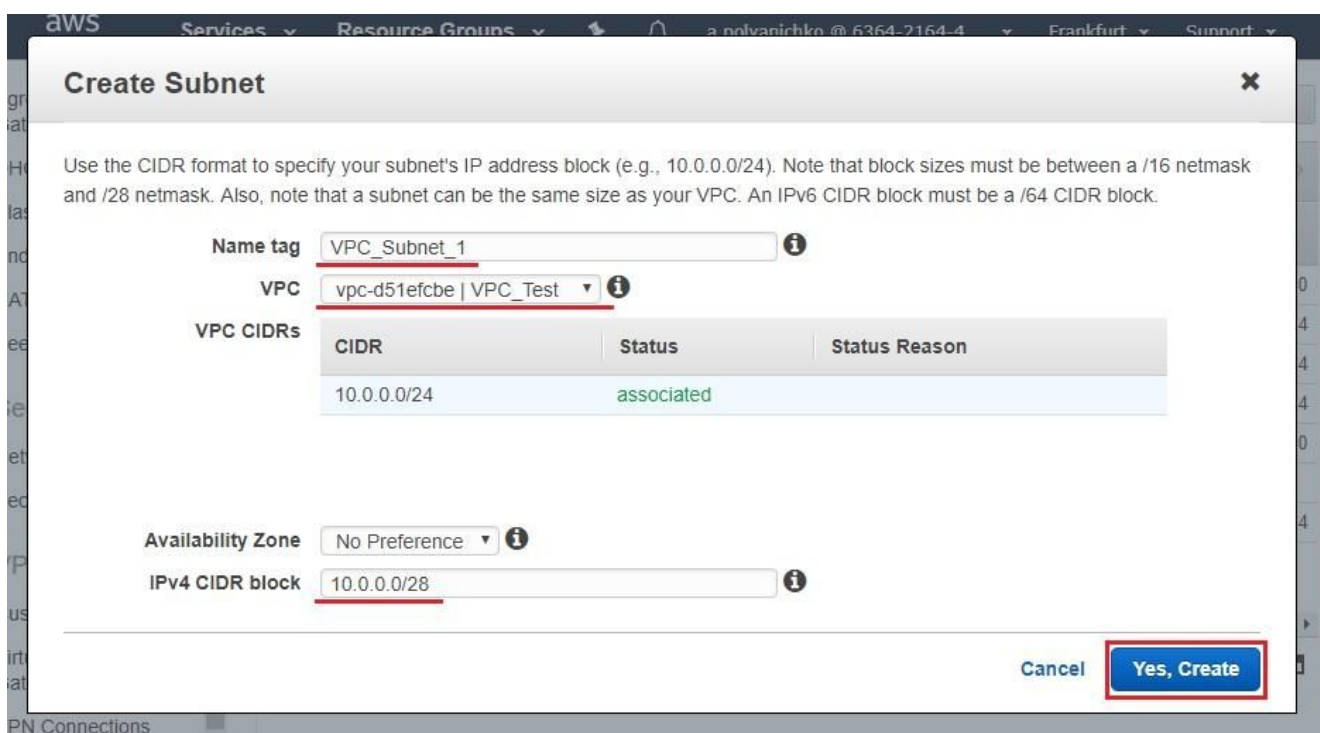
DNS hostnames: no

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

Click on “Create Subnet” button:



and then specify subnet parameters:



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 console interface. On the left, the navigation menu includes 'VPC Dashboard', 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways' (circled in red), 'Egress Only Internet Gateways', 'DHCP Options Sets', 'Elastic IPs', 'Endpoints', and 'NAT Gateways'. The main content area displays a list of subnets. A red arrow points to the 'VPC\_Subnet\_1' entry in the table. Below the table, the details for 'subnet-6e05a123 | VPC\_Subnet\_1' are shown, including its ID, CIDR, state, and associated VPC.

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

subnet-6e05a123 | VPC\_Subnet\_1

**Summary** | Route Table | Network ACL | Flow Logs | Tags

Subnet ID: subnet-6e05a123 | VPC\_Subnet\_1  
IPv4 CIDR: 10.0.0.0/28  
IPv6 CIDR:   
State: available  
VPC: vpc-d51efcbe | VPC\_Test  
Available IPs: 11

Availability Zone: eu-central-1c  
Route table: rtb-07e6636c  
Network ACL: acl-4c2b8627  
Default subnet: no  
Auto-assign Public IP: no  
Auto-assign IPv6 address: no

Click on “Create Internet Gateway” button:

The screenshot shows the AWS VPC console interface for the 'Internet Gateways' section. The left-hand navigation menu has 'Internet Gateways' highlighted. The main content area displays a table for listing Internet Gateways. The 'Create Internet Gateway' button is highlighted with a red box. Below the button, there is a search bar and a table for listing Internet Gateways.

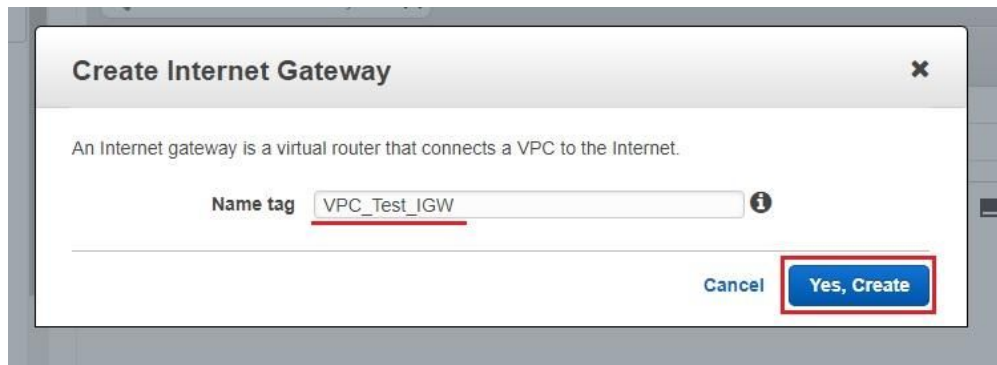
**Create Internet Gateway** | Delete | Attach to VPC | Detach from VPC

Search Internet Gateways and X

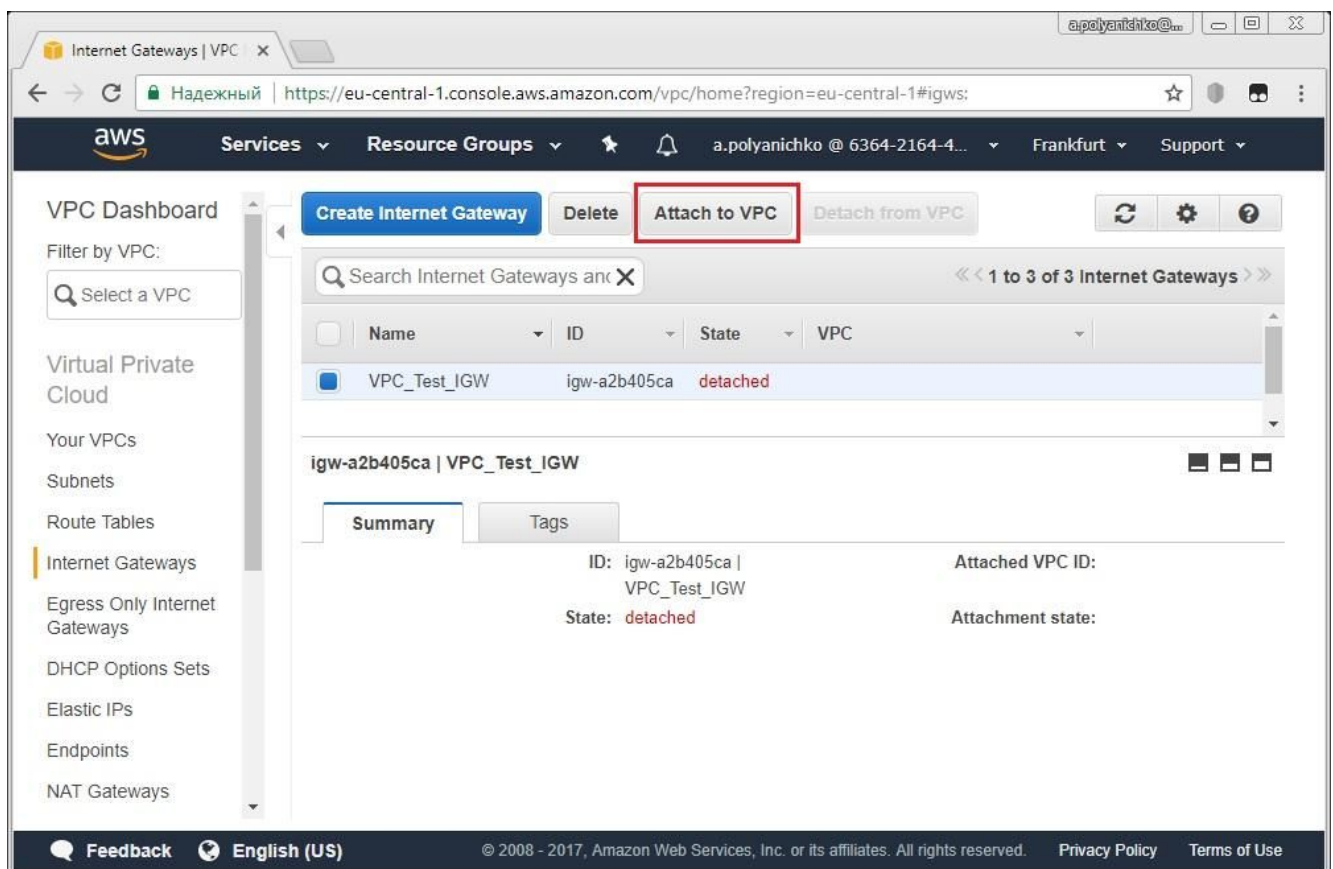
<< 1 to 2 of 2 Internet Gateways >>

Select an Internet gateway above

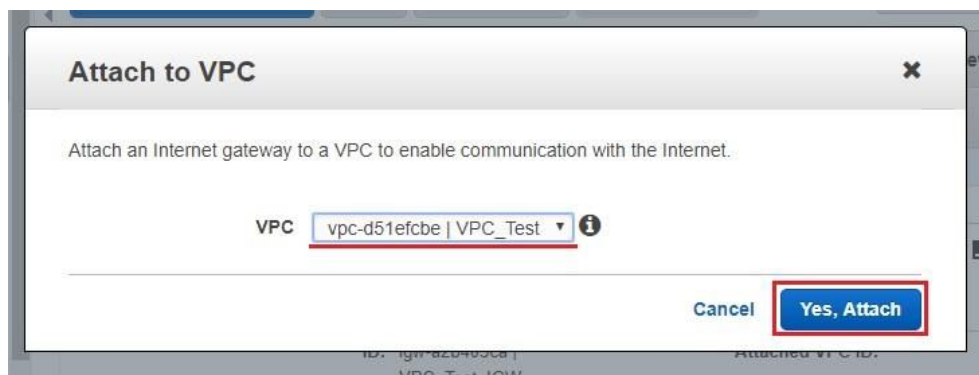
specify IGW name and then click “Yes, Create” button:



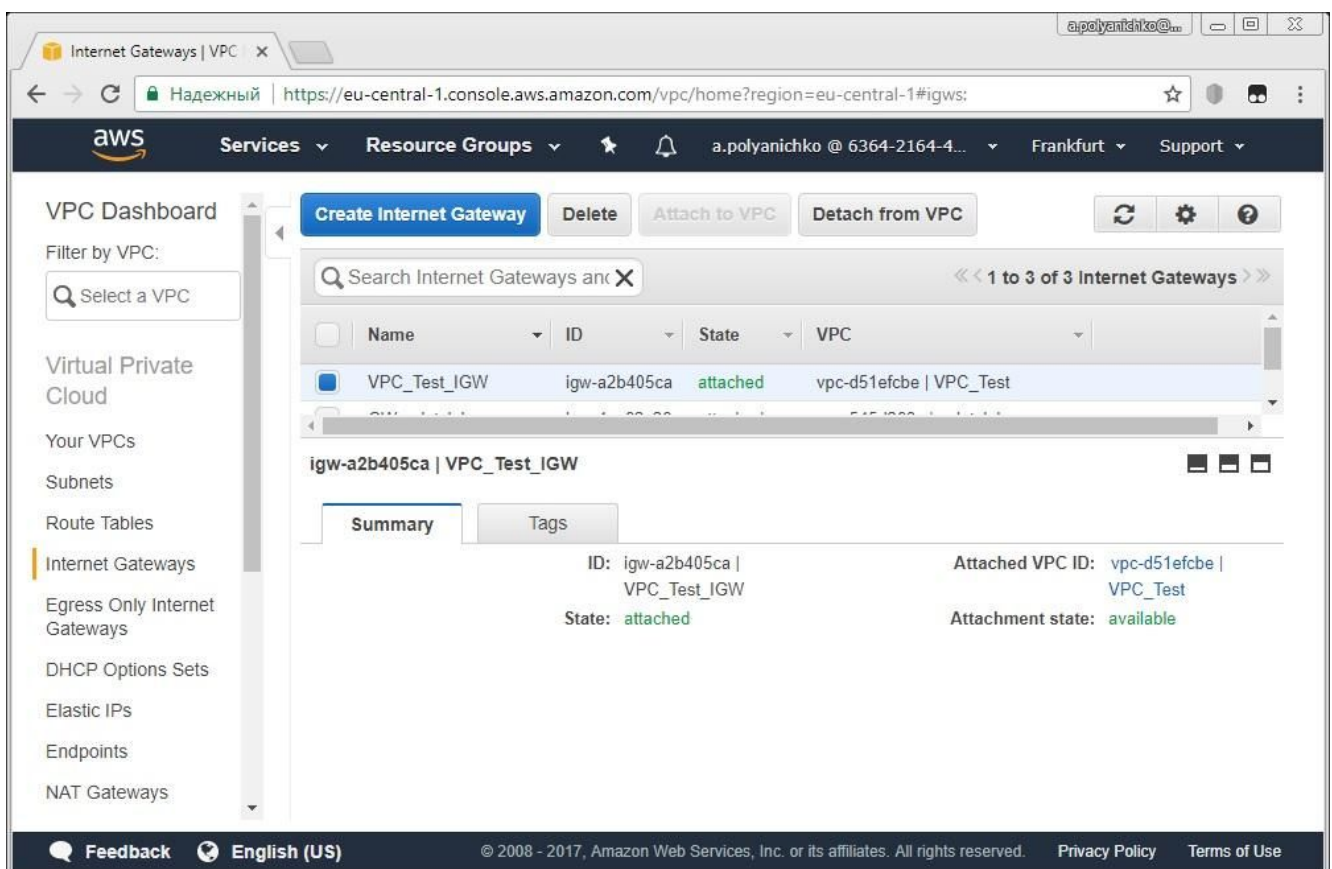
On the next page select your IGW and then click on “Attach to VPC” button:



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



Finally you can see your IGW is attached to VPC:



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



The screenshot shows the AWS VPC Management console for the region eu-central-1. The left sidebar contains navigation links for VPC Dashboard, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main content area displays the details for VPC vpc-d51efcbe (VPC\_Test). The 'Route table' field is circled in red, showing it is set to 'rtb-07e6636c'.

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

**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

Check the content of default routing table:

The screenshot shows the AWS VPC Management console for the region eu-central-1, specifically the 'Route Tables' section. The left sidebar contains navigation links for VPC Dashboard, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main content area displays the details for Route Table rtb-07e6636c. The 'Routes' tab is selected, showing a single route for destination 10.0.0.0/24 with target 'local'.

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

**rtb-07e6636c**

**Summary** | **Routes** | Subnet Associations | Route Propagation | Tags

**Edit**

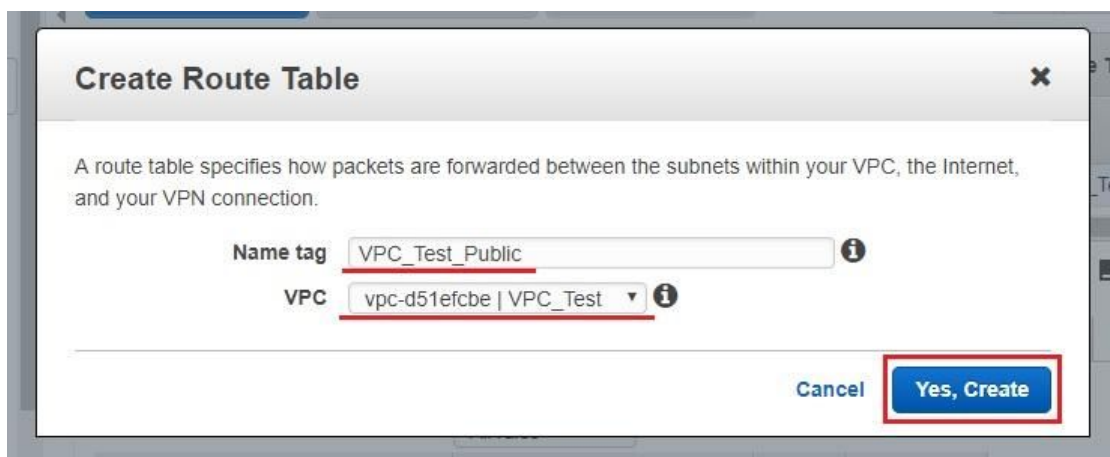
View: All rules

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No

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:  and then specify Name tag for new routing table as shown:



**Create Route Table**

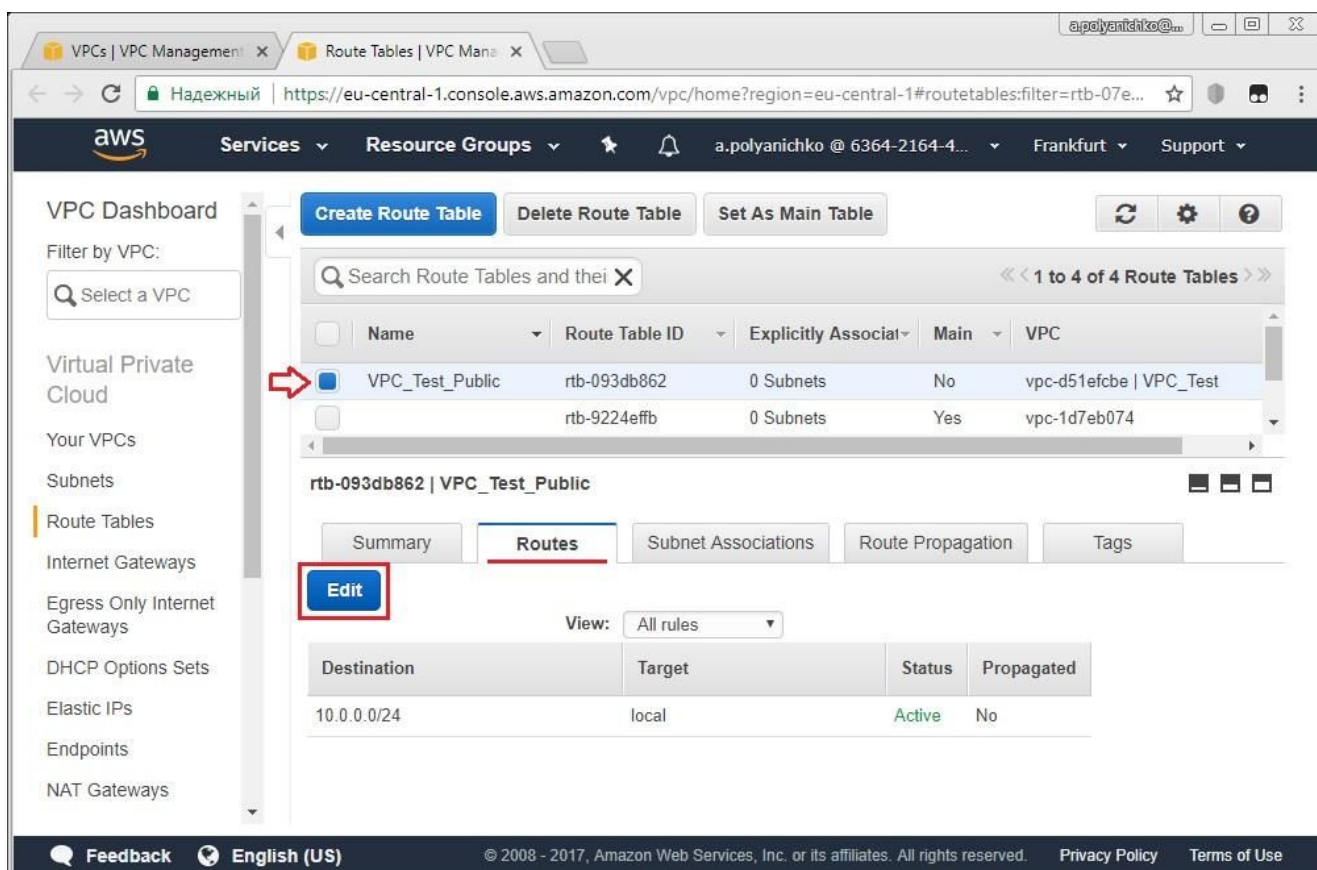
A route table specifies how packets are forwarded between the subnets within your VPC, the Internet, and your VPN connection.

Name tag:  ⓘ

VPC:  ⓘ

[Cancel](#) [Yes, Create](#)

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



The screenshot shows the AWS VPC console interface. The left sidebar contains navigation links for VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main content area displays a list of route tables. The first route table, "VPC\_Test\_Public" (ID: rtb-093db862), is selected. Below the list, the "Routes" tab is active, showing a single route with destination 10.0.0.0/24 and target "local". The "Edit" button is highlighted with a red box.

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

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No

Click on “Add another route” button:

The screenshot shows the AWS Management Console interface for a route table. On the left, a sidebar lists various AWS services: Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main panel displays the 'Routes' tab for a specific route table. At the top, there are tabs for 'Summary', 'Routes', 'Subnet Associations', 'Route Propagation', and 'Tags'. Below these tabs, there are 'Cancel' and 'Save' buttons. A 'View: All rules' dropdown is present. A table lists the current routes with columns: Destination, Target, Status, Propagated, and Remove. The table contains one entry: Destination '10.0.0.0/24', Target 'local', Status 'Active', and Propagated 'No'. Below the table, the 'Add another route' button is highlighted with a red rectangular box.

Destination	Target	Status	Propagated	Remove
10.0.0.0/24	local	Active	No	

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

This screenshot shows the AWS Management Console interface for a route table, specifically the 'Routes' tab. The title bar indicates the route table ID 'rtb-093db862' and the VPC name 'VPC\_Test\_Public'. The interface includes tabs for 'Summary', 'Routes', 'Subnet Associations', 'Route Propagation', and 'Tags'. Below the tabs are 'Cancel' and 'Save' buttons, with the 'Save' button highlighted by a red box. A 'View: All rules' dropdown is also visible. The route table contains two entries: the first is '10.0.0.0/24' pointing to 'local' with status 'Active'; the second is '0.0.0.0/0' pointing to 'igw-a2b405ca' with status 'Active'. The '0.0.0.0/0' destination is entered in a text input field, and 'igw-a2b405ca' is selected from a dropdown menu. An 'Add another route' button is located at the bottom of the table. The sidebar on the left lists various AWS services, including 'Peering Connections' at the bottom.

Destination	Target	Status	Propagated	Remove
10.0.0.0/24	local	Active	No	
0.0.0.0/0	igw-a2b405ca	Active	No	

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

The screenshot shows the AWS Management Console interface for the 'Route Tables' section. The left sidebar contains the 'VPC Dashboard' and a list of VPC resources. The main content area displays a table of route tables. The 'Subnet Associations' tab is selected, showing a message: '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:'. Below this message is a table listing subnets and their associated route tables. The 'Edit' button is highlighted with a red box.

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

Associate your VPC\_Subnet\_1 with the route table by selecting its checkbox and then click on “Save” button:

The screenshot shows the AWS Management Console interface for the 'Route Tables' section. The left sidebar contains the 'VPC Dashboard' and a list of VPC resources. The main content area displays a table of route tables. The 'Subnet Associations' tab is selected, showing a message: '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:'. Below this message is a table listing subnets and their associated route tables. The 'Save' button is highlighted with a red box.

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

Now you may ensure that at the moment your VPC and your Subnet\_1 use different routing tables:



The screenshot shows the AWS VPC Dashboard for the eu-central-1 region. The left sidebar contains navigation links for VPC Dashboard, Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, and Endpoints. The main content area displays a list of VPCs with a search bar and filters. A red arrow points to the 'VPC\_Test' entry in the list. Below the list, the details for 'vpc-d51efcbe | VPC\_Test' are shown, including its ID, state (available), IPv4 CIDR (10.0.0.0/24), and DHCP options set (dopt-a049a6c9). The 'Route table' is highlighted with a red circle and labeled as 'rtb-07e6636c'.

**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

**Create VPC** **Actions**

Search VPCs and their properties

<< 1 to 3 of 3 VPCs >>

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options
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

The screenshot shows the AWS VPC Dashboard for the eu-central-1 region, specifically the Subnets page. The left sidebar contains navigation links for VPC Dashboard, Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, and NAT Gateways. The main content area displays a list of subnets with a search bar and filters. A red arrow points to the 'VPC\_Subnet\_1' entry in the list. Below the list, the details for 'subnet-6e05a123 | VPC\_Subnet\_1' are shown, including its ID, state (available), VPC (vpc-d51efcbe | VPC\_Test), and IPv4 CIDR (10.0.0.0/28). The 'Route Table' is highlighted with a red circle and labeled as 'rtb-093db862 | VPC\_Test\_Public'.

**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 Subnet** **Subnet Actions**

Search subnets and their properties

<< 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 | VPC\_Subnet\_1**

**Summary** **Route Table** **Network ACL** **Flow Logs** **Tags**

**Edit**

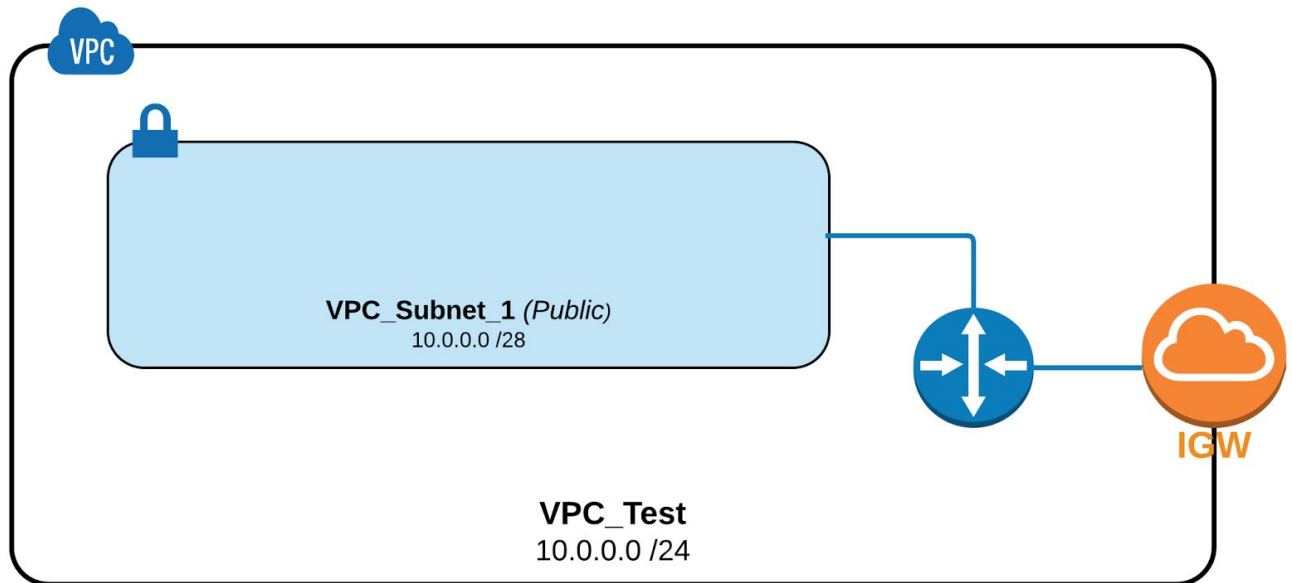
Route Table: **rtb-093db862 | VPC\_Test\_Public**

Destination	Target
10.0.0.0/24	local
0.0.0.0/0	igw-a2b405ca

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



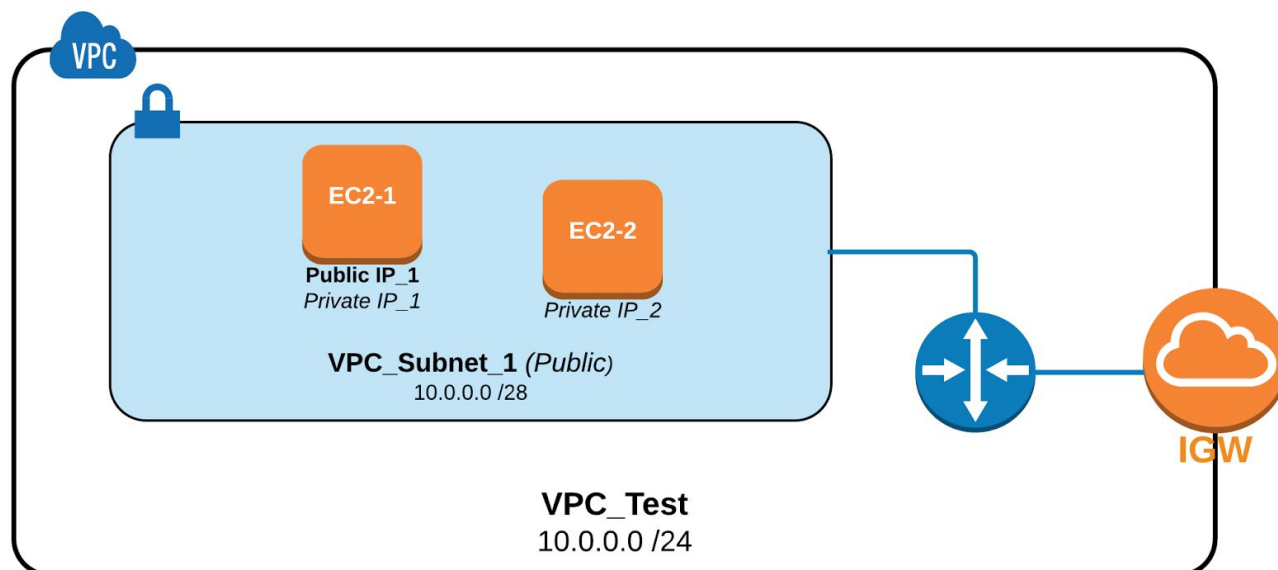
At the moment we have created the basic of our network infrastructure:



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

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):

The screenshot shows the AWS Management Console interface for the 'Step 3: Configure Instance Details' page. The browser address bar shows the URL: <https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LaunchInstanceWizard>. The page has a dark blue header with the AWS logo and navigation links. Below the header, there are seven tabs: '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance' (which is active), '4. Add Storage', '5. Add Tags', '6. Configure Security Group', and '7. Review'. The main content area is titled 'Step 3: Configure Instance Details' and includes a sub-header: '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.' The configuration options are as follows:

- Number of instances:** A text input field containing '1'. To its right is a link: 'Launch into Auto Scaling Group'.
- Purchasing option:** A checkbox labeled 'Request Spot instances' which is currently unchecked.
- Network:** A dropdown menu showing 'vpc-d51efcbe | VPC\_Test'. To its right is a link: 'Create new VPC'.
- Subnet:** A dropdown menu showing 'subnet-6e05a123 | VPC\_Subnet\_1 | eu-central-1c'. Below this, it says '11 IP Addresses available'. To its right is a link: 'Create new subnet'.
- Auto-assign Public IP:** A dropdown menu showing 'Enable'.
- IAM role:** A dropdown menu showing 'None'. To its right is a link: 'Create new IAM role'.

At the bottom of the configuration area, there are four buttons: 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Add Storage'. The footer of the page contains a 'Feedback' link, 'English (US)' language selection, copyright information '© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

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

Subnets | VPC Management | EC2 Management Console

Надежный | https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LaunchInstanceWizard:

aws Services Resource Groups a.polyanichko @ 6364-2164-4... Frankfurt Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### 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	Value	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):

Subnets | VPC Management | EC2 Management Console

Надежный | https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LaunchInstanceWizard:

aws Services Resource Groups a.polyanichko @ 6364-2164-4... Frankfurt Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### 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	e.g. SSH for Admin Des
All ICMP - IPv4	ICMP	0 - 65535	Custom 0.0.0.0/0	e.g. SSH for Admin Des

[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:

The screenshot shows the AWS Management Console for the eu-central-1 region. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and Snapshots. The main content area displays a table of EC2 instances. The instance 'VPC\_Sub1\_Ec2-1' is highlighted, and its details are shown below. The Public IP is 52.57.192.224.

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

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

Note Private IP and Public IP of your instance then 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:

The screenshot shows the AWS Management Console interface for the 'Configure Instance Details' step of an EC2 instance launch wizard. The browser address bar shows the URL: <https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LaunchInstanceWizard:>. The console header includes the AWS logo, navigation tabs (Services, Resource Groups), and user information (a.polyanichko @ 6364-2164-4...). The wizard progress bar shows seven steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance (active), 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The main content area is titled 'Step 3: Configure Instance Details' with a subtitle: '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.' The configuration fields are as follows: 'Number of instances' is set to 1, with a 'Launch into Auto Scaling Group' link; 'Purchasing option' has the 'Request Spot instances' checkbox unchecked; 'Network' is set to 'vpc-d51efcbe | VPC\_Test' with a 'Create new VPC' link; 'Subnet' is set to 'subnet-6e05a123 | VPC\_Subnet\_1 | eu-central-1c' with a 'Create new subnet' link and a note '10 IP Addresses available'; 'Auto-assign Public IP' is set to 'Disable' (highlighted with a red underline); and 'IAM role' is set to 'None' with a 'Create new IAM role' link. At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch' (in blue), and 'Next: Add Storage'. The footer contains a 'Feedback' link, 'English (US)' language selection, and copyright information: '© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.' along with 'Privacy Policy' and 'Terms of Use' links.




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.

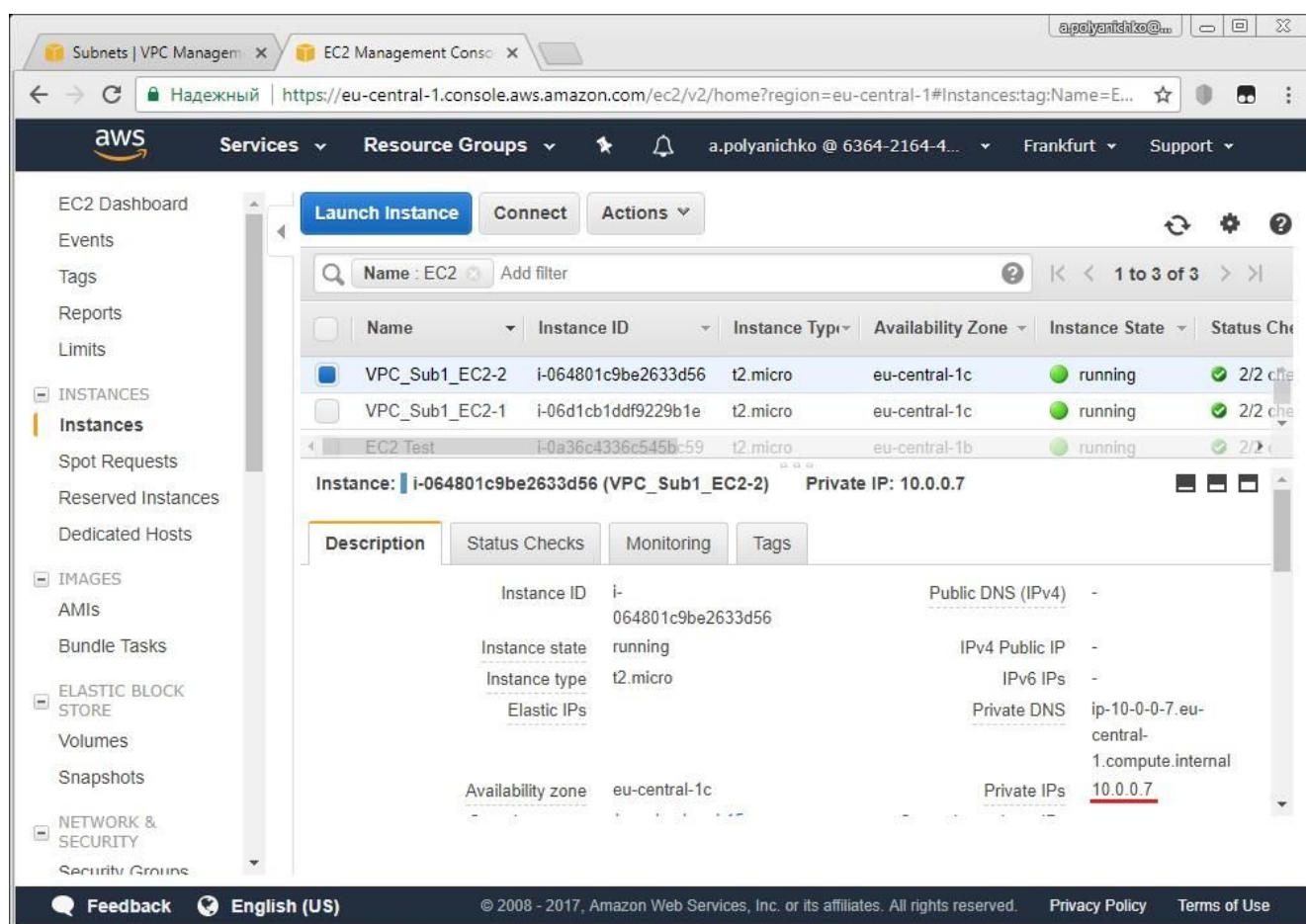
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances 	Volumes 	
Name	VPC_Sub1_EC2-2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<div>Add another tag (Up to 50 tags maximum)</div>				



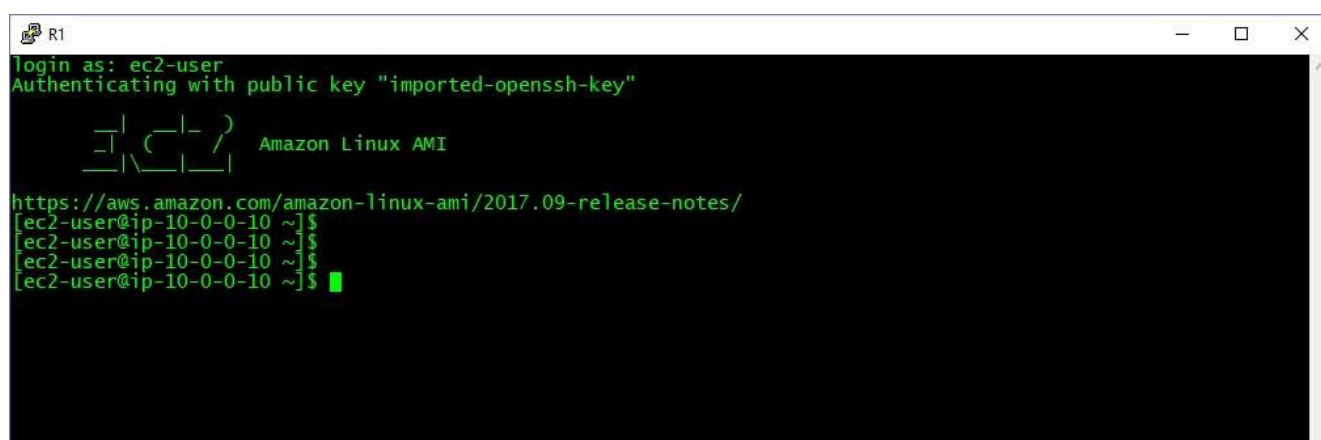
Please don't forget to enable ICMP traffic on "Configure Security Group" tab.

Find your second instance on EC2 Dashboard:

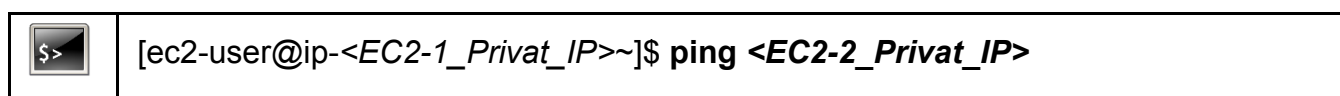


Note Private IP address of the instance (public IP must be absent).

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



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





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

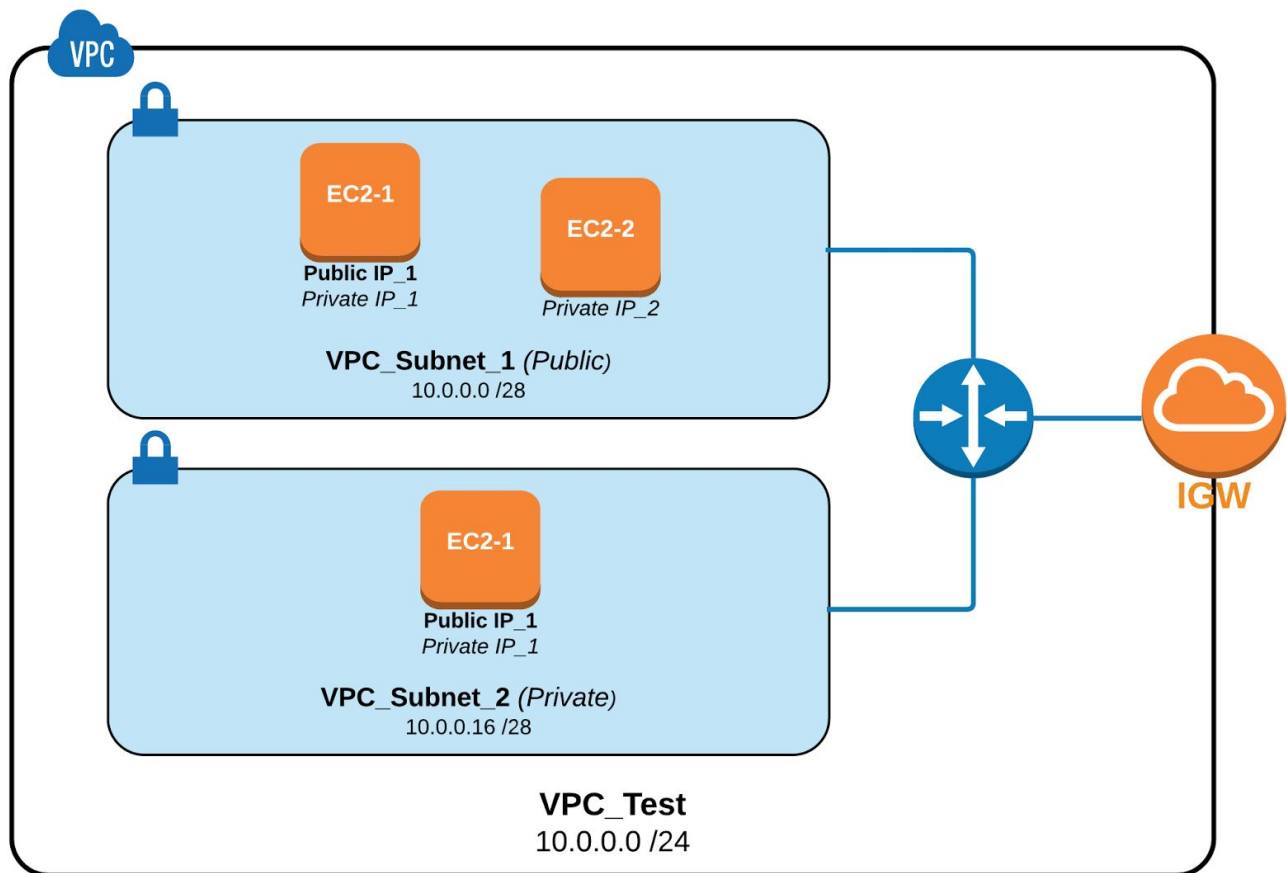
 _| _|_ )
 _| ( /  Amazon Linux AMI
 _|\_|_|

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:



**Create Subnet**

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag:

VPC:

CIDR	Status	Status Reason
10.0.0.0/24	associated	

Availability Zone:

IPv4 CIDR block:

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

Subnets | VPC Management

Filter by 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

**subnet-999235d4 | VPC\_Subnet\_2**

**Summary**

Subnet ID: subnet-999235d4 | VPC\_Subnet\_2

IPv4 CIDR: 10.0.0.16/28

IPv6 CIDR:

State: available

VPC: vpc-d51efcbe | VPC\_Test

Available IPs: 11

Availability Zone: eu-central-1c

Route table: rtb-07e6636c

Network ACL: acl-4c2b8627

Default subnet: no

Auto-assign Public IP: no

Auto-assign IPv6 address: no

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):

EC2 Management Console

Надежный | https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LaunchInstanceWizard:

Services Resource Groups a.polyanichko @ 6364-2164-4... Frankfurt Support

1. Choose AMI 2. Choose Instance Type **3. Configure Instance** 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### 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 Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network vpc-d51efcbe | VPC\_Test Create new VPC

Subnet subnet-999235d4 | VPC\_Subnet\_2 | eu-central-1c Create new subnet  
11 IP Addresses available

Auto-assign Public IP Enable

IAM role 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

EC2 Management Console

Надежный | https://eu-central-1.console.aws.amazon.com/ec2/v2/home?region=eu-central-1#LaunchInstanceWizard:

Services Resource Groups a.polyanichko @ 6364-2164-4... Frankfurt Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage **5. Add Tags** 6. Configure Security Group 7. Review

### 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)

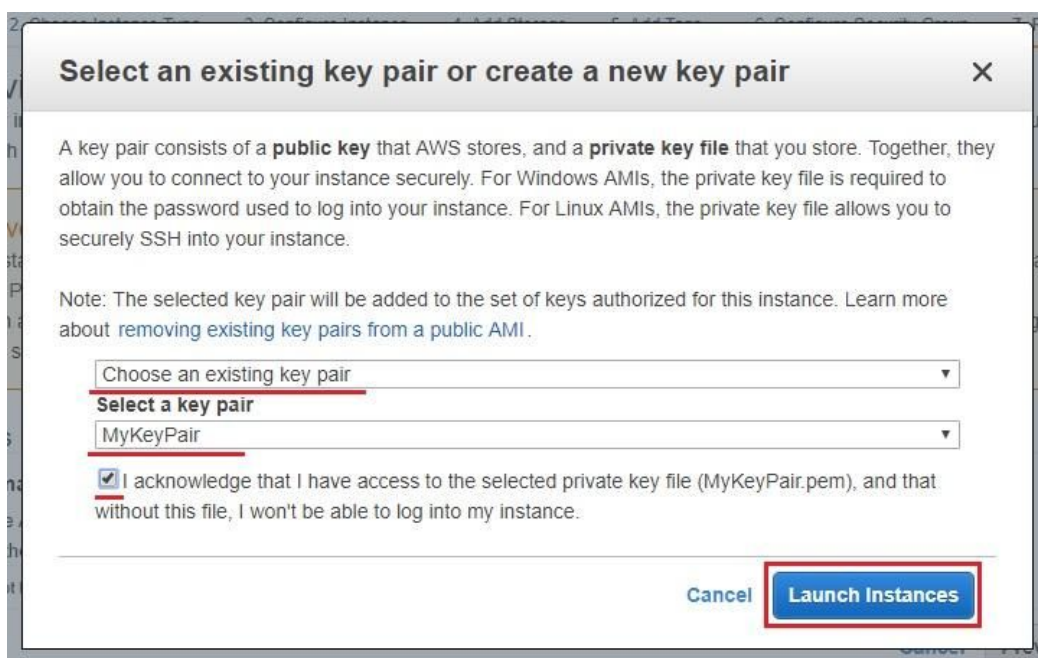
Cancel Previous Review and Launch Next: Configure Security Group

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



Don't forget to enable ICMP protocol in Security Group for the instance

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:

The screenshot shows the AWS EC2 Management Console. On the left, the navigation pane includes sections for INSTANCES, IMAGES, and ELASTIC BLOCK STORE. Under INSTANCES, 'Instances' is selected. The main area displays a table of instances:

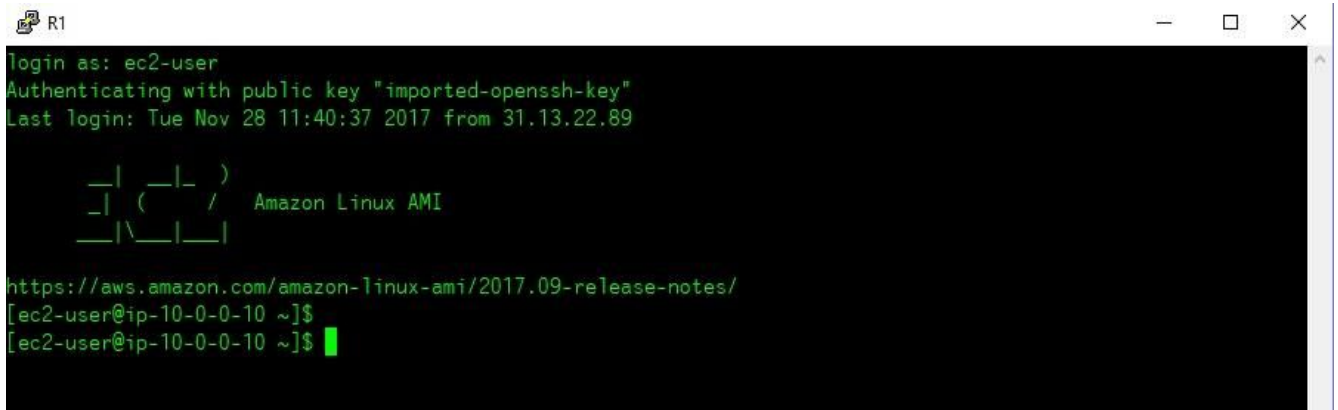
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
VPC_Sub2_EC2-1	i-046d116fb1c6c4082	t2.micro	eu-central-1c	running	Initial...
VPC_Sub1_EC2-2	i-064801c9be2633d56	t2.micro	eu-central-1c	running	2/2 che...
VPC_Sub1_EC2-1	i-06d1cb1ddf9229b1e	t2.micro	eu-central-1c	running	2/2 che...

Below the table, the details for the selected instance 'VPC\_Sub2\_EC2-1' (Instance ID: i-046d116fb1c6c4082) are shown. The 'Public IP: 18.195.87.84' is circled in red.

The details section includes tabs for Description, Status Checks, Monitoring, and Tags. The 'Description' tab is active, showing the following information:

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

Login to your VPC\_Subn1\_EC2-1 (the very first) instance via PuTTY SSH using its Public IP:



```

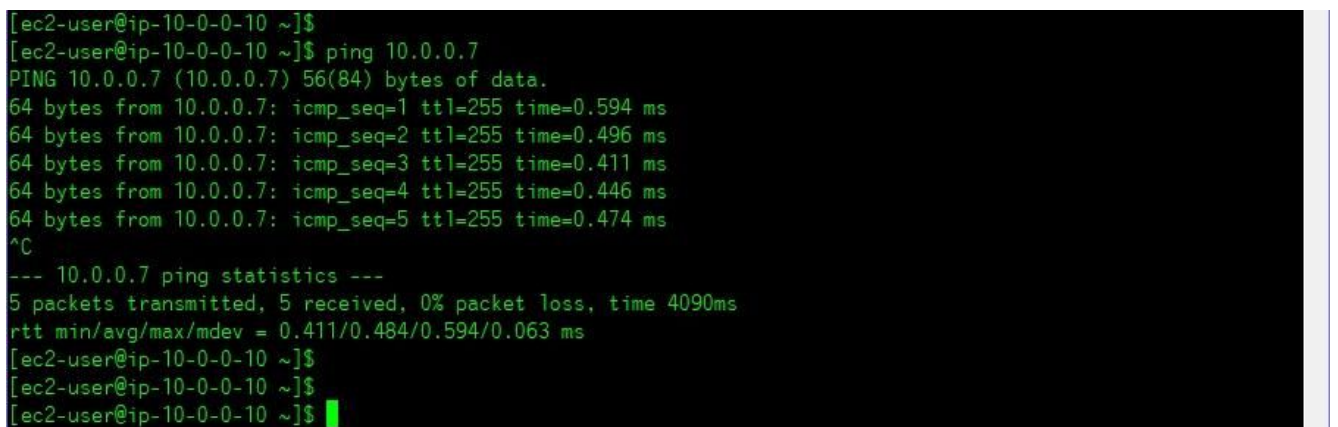
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Tue Nov 28 11:40:37 2017 from 31.13.22.89

  _| _|_ )
 _| (  /  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:

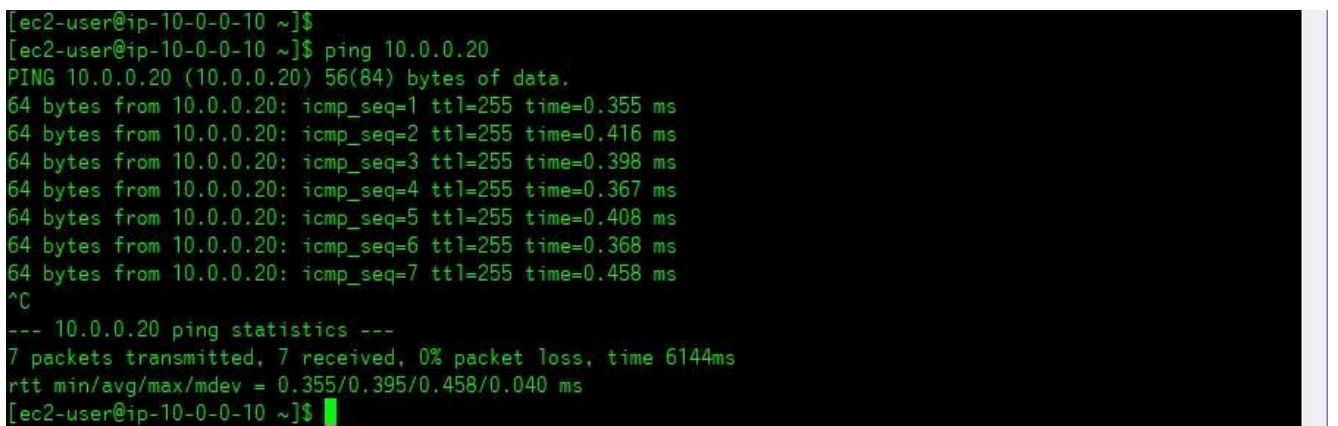


```

[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:



```

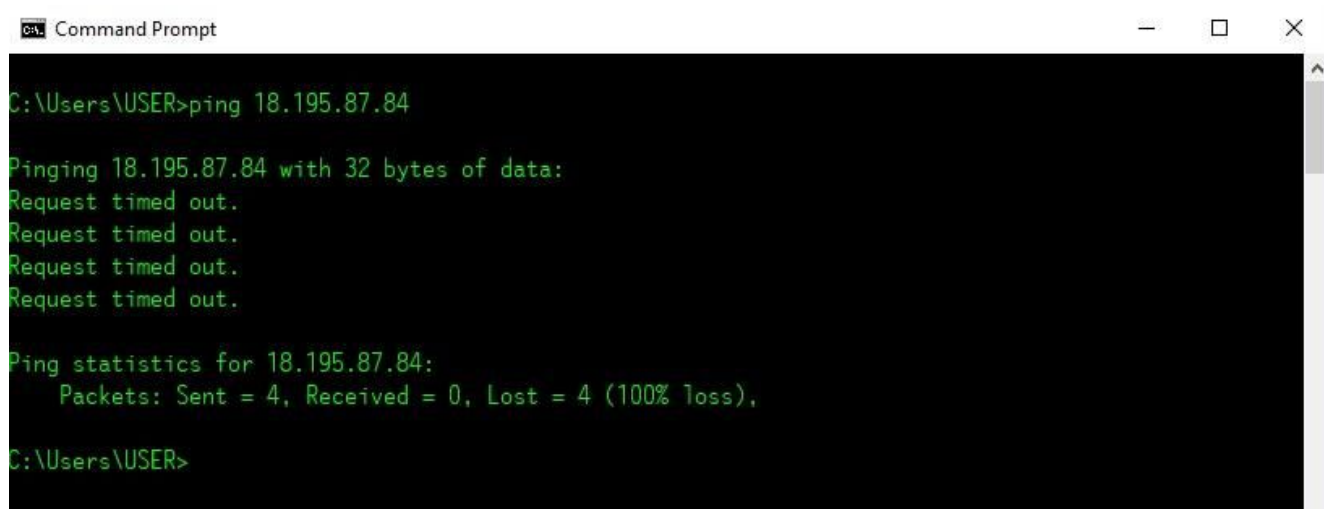
[ec2-user@ip-10-0-0-10 ~]$
[ec2-user@ip-10-0-0-10 ~]$ ping 10.0.0.20
PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data.
64 bytes from 10.0.0.20: icmp_seq=1 ttl=255 time=0.355 ms
64 bytes from 10.0.0.20: icmp_seq=2 ttl=255 time=0.416 ms
64 bytes from 10.0.0.20: icmp_seq=3 ttl=255 time=0.398 ms
64 bytes 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.