

AWS VPC NACL Lab - Case study

Level: **Advanced**

Amazon EC2 Amazon VPC Amazon Web Services

English ▼

Required Points

💎 10

Lab Duration

01:00:00

Average Start time

Less than a minute

Start Guided Lab →

Lab Overview

Lab Steps



Cloud Architect



Compute, Networking

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.

2. On the AWS sign-in page,

- Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
- Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.

3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1**.

[Privacy](#) - [Terms](#)

Task 2: Creating a New VPC

1. Navigate to VPC by clicking on the **Services** button on the top of the AWS Console.
2. Click on **VPC** (under **Networking & Content Delivery** section) or you can also search for VPC.
3. Click on **Your VPCs** from the left menu.
4. Here you can see the list of all VPC. No need to do anything yet. We will create a new VPC for this lab.
5. Click on **Create VPC**.
 - **Name tag:** Enter **MyVPC**
 - **IPv4 CIDR block:** Enter **10.0.0.0/16**
 - **IPv6 CIDR block:** No need to change this, make sure **No IPv6 CIDR Block** is checked.
 - **Tenancy:** No need to change this, just be sure **Default** is selected.
 - Click on **Create VPC**.
6. Once the VPC is created, it will look like the example below:

<input checked="" type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR
<input checked="" type="checkbox"/>	MyVPC	vpc-0b42ca1898f75e49c	Available	10.0.0.0/16

Task 3: Creating Subnets

Note: In this lab, we will create one public subnet and a private subnet in us-east-1a and us-east-1b Availability Zones.

1. For the Public Subnet, click on **Subnets** from the left menu and click on **Create subnet**.
 - **VPC ID** : Select **MyVPC** from the list.
 - **Subnet Name** : Enter **MyPublicSubnet**
 - **Availability Zone** : Select **us-east-1a**
 - **IPv4 CIDR block** : Enter the range **10.0.1.0/24**
 - Click on **Create Subnet**
2. For Private Subnet, click on **Create Subnet** again.
 - **VPC ID** : Select **MyVPC** from the list.

- **Subnet Name** : Enter **MyPrivateSubnet**
- **Availability Zone** : Select **us-east-1b**
- **IPv4 CIDR block** : Enter the range **10.0.2.0/24**
- Click on **Create subnet**.

<input type="checkbox"/>	MyPublicSubnet	subnet-01c1aa6cf0e8a8bb2	✔ Available	vpc-0b42ca1898f75e49c My...	10.0.1.0/24
<input type="checkbox"/>	MyPrivateSubnet	subnet-0785e551074b3149b	✔ Available	vpc-0b42ca1898f75e49c My...	10.0.2.0/24

Task 4: Create and attach an Internet Gateway

Note: By default, instances that are launched in a VPC cannot communicate with the Internet.

To enable Internet access, an Internet gateway needed to be attached to the VPC.

1. Click on **Internet Gateways** from the left menu and click **Create Internet Gateway**.

- **Name Tag** : Enter **MyInternetGateway**
- Click on **Create Internet Gateway**.

2. Select the Internet gateway you created from the list.

- Click on **Actions**.
- Click on **Attach to VPC**.
- Select MyVPC and click on **Attach to VPC**.

✔ The following internet gateway was created: igw-07c68ed0f807e819a . You can now attach to a VPC to enable the VPC to communicate with the internet.
Attach to a VPC

VPC > Internet gateways > igw-07c68ed0f807e819a

igw-07c68ed0f807e819a / MyInternetGateway

Actions
Attach to VPC
Detach from VPC
Manage tags
Delete

Details
Info

Internet gateway ID	State	VPC ID	Owner
igw-07c68ed0f807e819a	Detached	-	571174007677

Tags
Manage tags

Search tags

Key	Value
Name	MyInternetGateway

Task 5: Create Route Tables and Associate them it with Subnets

1. Go to **Route Tables** from the left menu and click on **Create route table**.

- **Name Tag:** Enter **PublicRouteTable**.
- **VPC:** Select **MyVPC** from the list.
- Click on **Create route table**.

2. We will be using the **default (main) Route Table** created by VPC for the RDS database tier.

<input type="checkbox"/>	Name ▾	Route table ID ▾	Explicit subnet associat...	Edge associations	Main ▾	VPC ▾
<input type="checkbox"/>	Default VPC RT	rtb-cb7364b4	–	–	Yes	vpc-0de26a77 Default VPC
<input type="checkbox"/>	–	rtb-0ec4aff97b65279d	–	–	Yes	vpc-0b42ca1898f75e49c My...
<input type="checkbox"/>	PublicRouteTable	rtb-023a53454fa2269c5	–	–	No	vpc-0b42ca1898f75e49c My...

- You will be able to see the Route table with **VPC ID MyVPC** and **Main** as **Yes**
- Select the Route Table and rename it.
- **Name Tag:** Enter **PrivateRouteTable** and [Enter]

Name ▾	Route table ID ▾	Explicit subnet associat...	Edge associations	Main ▾	VPC ▾
Default VPC RT	rtb-cb7364b4	–	–	Yes	vpc-0de26a77 Default VPC
PrivateRouteTable	rtb-0ec4aff97b65279d	–	–	Yes	vpc-0b42ca1898f75e49c My..
PublicRouteTable	rtb-023a53454fa2269c5	–	–	No	vpc-0b42ca1898f75e49c My..

3. Now **associate the subnets** to the route tables.

4. Click on **PublicRouteTable** and go to the **Action** and in that go to **Edit Subnet Associations tab**.

- Click on **Edit Subnet Associations**.
- Select **MyPublicSubnet** from the list.
- Click on **Save Associations**

5. Click on **PrivateRouteTable** and go to the **Action** and in that go to **Edit Subnet Associations tab**.

- Click on **Edit Subnet Associations**.
- Select **MyPrivateSubnet** from the list.
- Click on **Save Associations**

Task 6: Update Route Table and Configure the Internet Gateway

1. **PublicRouteTable** : Add a route to allow Internet traffic to the VPC.

- Select **PublicRouteTable**.

- Go to the **Routes** tab click on **Edit routes**. On the next page, click on **Add route**.
- Specify the following values:
 - **Destination:** Enter **0.0.0.0/0**
 - **Target:** Select **Internet Gateway** from the dropdown menu to select **MyInternetGateway**.
 - Click on **Save changes**.

Task 7: Enabling Auto-Assign Public IP for Public Subnets

Note: This setting will allow you to automatically assign public IP for all the EC2 instances launched in the public subnet

Click on **Subnets** from the left menu on VPC.

- Select **MyPublicSubnet** from the Subnet list
- Click on **Actions** and then select **Edit subnet settings**
- Check the **Enable Auto-assign IPv4 address** check box
- Check the **Enable resource name DNS A record on launch** check box
- Now click on **Save**

Task 8: Launching an EC2 Instance in the Public Subnet

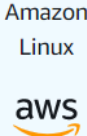
1. Navigate to **EC2** by clicking on the **Services** menu in the top, then click on **EC2** in the **Compute** section.
2. Navigate to **Instances** from the left side menu and click on **Launch Instance**.
3. Enter name as **MyPublicEC2Server**
4. Choose an Amazon Machine Image (AMI): Select **Amazon Linux 2 AMI** in the drop-down.
 - Choose **architecture** as **64-bit(x86)**

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

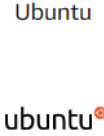
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images


Recents | **Quick Start**




Amazon Linux




Ubuntu



Windows



Red Hat



SUSE Linux

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86_64 HVM gp2

Architecture AMI ID

64-bit (x86) ▼

ami-0cff7528ff583bf9a

5. Choose an **Instance Type**: Select **t2.micro**.

▼ **Instance type** [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible ▼

[Compare instance types](#)

6. For **Key pair**: Select **Create a new key pair** Button

- Key pair name: **WhizKey**
- Key pair type: **RSA**
- Private key file format: **.pem**

7. Select **Create key pair** Button.

Create key pair

×

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

WhizKey

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

Cancel

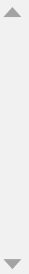
Create key pair

8. In Network Settings Click on **Edit** Button:

- VPC : **MyVPC**
- Subnet : Choose **MyPublicSubnet**
- Auto-assign public IP: **Enable**
- Select **Create new Security group**
- Security group name : Enter **MyWebserverSG**
- Description : Enter **My EC2 Security Group**
- Check Allow SSH from and Select Anywhere from dropdown
 - Source: Select **Anywhere**
 - Choose Type: **SSH**
- For **HTTP**, Select **Add Security rule** Button
 - Choose Type: **HTTP**
 - Source: Select **Anywhere**

9. Under the **Advanced Details**, scroll down to the User data section, enter the following script to create an HTML page served by Apache:

```
#!/bin/bash
sudo su
yum update -y
yum install httpd -y
echo "<html><h1>Welcome to Whizlabs Server </h1><html>" >> /var/www/html/index.html
systemctl start httpd
systemctl enable httpd
```



10. Keep Rest thing Default and Click on **Launch Instance** Button.

11. Select **View all Instances** to View Instance you Created

12. **Launch Status:** Your instance is now launching, Click on the instance ID and wait for complete initialization of the instance till status changes to **Running**.

Task 9: Launching an EC2 Instance in the Private Subnet

1. Click on **Launch Instances** again at the top right of the EC2 dashboard.
2. Enter name as **MyPrivateEC2Server**
3. Choose an Amazon Machine Image (AMI): Select **Amazon Linux 2 AMI** in the drop-down.
 - Choose **architecture** as **64-bit(x86)**

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **Quick Start**

Amazon Linux

Ubuntu

Windows

Red Hat

SUSE Linux

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86_64 HVM gp2

Architecture AMI ID

64-bit (x86) ▼

ami-0cff7528ff583bf9a

5. Choose an **Instance Type**: Select **t2.micro**.

▼ **Instance type** [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible ▼

[Compare instance types](#)

6. For **Key pair**: Select **the existing key pair**.

7. In Network Settings Click on **Edit** Button:

- VPC : **MyVPC**
- Subnet : Choose **MyPrivateSubnet**
- Auto-assign public IP: **Disable**
- Select **Create new Security group**
- Security group name : Enter **MyServerSG**

- Description : Enter **My EC2 Security Group**
- Check Allow SSH from and Select Anywhere from dropdown
 - Choose Type: **SSH**
 - Source: Select **Anywhere**
- For **ALL ICMP IPv4** , Select **Add Security rule** Button
 - Choose Type: **All ICMP IPv4**.
 - Source: Select **Anywhere**

8. Keep Rest thing Default and Click on **Launch Instance** Button.

9. Select **View all Instances** to View Instance you Created

10. Note the Private IP Address of **MyPrivateEC2Server**.

11. Two servers are launched and ready.

<input type="checkbox"/>	MyPrivateEC2...	i-06fb2249d846c1237	✓ Running	🔍	t2.micro
<input type="checkbox"/>	MyPublicEC2S...	i-04bd2f06c2a61abb4	✓ Running	🔍	t2.micro

Task 10: Testing Both EC2 instances

1. **Public EC2 instances:** We have installed a web application on this server.

- Select the **MyPublicEC2Server** EC2 instance from the instance list.
- From the Description tab, copy the **IPv4 Public IP**.

Instance summary for i-04bd2f06c2a61abb4 (MyPublicEC2Server) Info

Updated less than a minute ago

Instance ID: i-04bd2f06c2a61abb4 (MyPublicEC2Server) **54.162.212.109** open address

Instance state: **Running**

Instance type: t2.micro

Private IPv4 addresses: 10.0.1.208

Private IPv4 DNS: ip-10-0-1-208.ec2.internal

VPC ID: vpc-08cd445d44e195fc7 (MyVPC)

- Now paste this IP in you Web Browser and click [Enter]
- You will be able to see the following page:





Welcome to Whizlabs Server

2. Next, we will try to ping the Private EC2 from the Public EC2 instance.

- SSH into EC2 Instance
 - Please follow the steps in [SSH into EC2 Instance](#).
- Once connected to the server:
 - Change to root user:

```
sudo su
```

- Copy the Private IP of **MyPrivateEC2Server** from the Description tab.

Instance summary for i-06fb2249d846c1237 (MyPrivateEC2Server) Info		
Updated less than a minute ago		
Instance ID	Public IPv4 address	Private IPv4 addresses
 i-06fb2249d846c1237 (MyPrivateEC2Server)	-	 10.0.2.161
Instance state	Public IPv4 DNS	Private IPv4 DNS
 Running	-	 ip-10-0-2-161.ec2.internal
Instance type	Elastic IP addresses	VPC ID

- Ping the Private Instance using the Private IPv4
- Example:

```
[root@ip-10-0-1-208 ec2-user]# ping 10.0.2.161
PING 10.0.2.161 (10.0.2.161) 56(84) bytes of data.
64 bytes from 10.0.2.161: icmp_seq=1 ttl=255 time=0.653 ms
64 bytes from 10.0.2.161: icmp_seq=2 ttl=255 time=0.656 ms
64 bytes from 10.0.2.161: icmp_seq=3 ttl=255 time=0.657 ms
64 bytes from 10.0.2.161: icmp_seq=4 ttl=255 time=0.762 ms
64 bytes from 10.0.2.161: icmp_seq=5 ttl=255 time=0.679 ms
64 bytes from 10.0.2.161: icmp_seq=6 ttl=255 time=0.796 ms
64 bytes from 10.0.2.161: icmp_seq=7 ttl=255 time=0.610 ms
64 bytes from 10.0.2.161: icmp_seq=8 ttl=255 time=0.776 ms
64 bytes from 10.0.2.161: icmp_seq=9 ttl=255 time=0.657 ms
64 bytes from 10.0.2.161: icmp_seq=10 ttl=255 time=0.610 ms
64 bytes from 10.0.2.161: icmp_seq=11 ttl=255 time=0.661 ms
64 bytes from 10.0.2.161: icmp_seq=12 ttl=255 time=0.623 ms
64 bytes from 10.0.2.161: icmp_seq=13 ttl=255 time=0.645 ms
```

- Press [Ctrl] + C to stop instead of pause.

- **Note:** You were able to do these tasks because the Default NACL that was created during VPC creation allows both INBOUND and OUTBOUND by Default.

Task 11: Creating Custom NACL and Associate it to the Subnet

Note: By default, both subnets will be associated with the Default NACL of **MyVPC**. Once you create a custom NACL and attach it to the public subnet and private Subnet.

1. Navigate to **VPC** under the Services menu. Click on **Network ACLs** under **Security**

2. Click on **Create Network ACL**

3. Create Network ACL:

- Name tag: Enter **MyPublicNACL**
- VPC: Select **MyVPC** from the dropdown list.
- Click on **Create**.

4. Associating MyPublicNACL to the Public Subnet

- Select the Action tab and click on **Edit subnet associations**
- Select both the **Public and Private** subnets from the table.
- Click on **Save changes**

5. Renaming the Main NACL

- Select the **Default NACL** of the VPC MyVPC

Network ACL ID	Associated with	Default	VPC ID
acl-0124111a8fd1c30a6	2 Subnets	No	vpc-08cd445d44e195fc7 / MyVPC
acl-3650a24b	6 Subnets	Yes	vpc-15048f6f / Default VPC
acl-0d3c0f51bdf787b59	–	Yes	vpc-08cd445d44e195fc7 / MyVPC

- Enter the name **MyPrivateNACL** and click on **Save**

Task 12: Testing the Public and Private Server

1. Public EC2 Instance:

- Navigate to the **EC2 Instance Dashboard**. Click on **Instances** from the left side menu.
- Select the **MyPublicEC2Server** EC2 instance from the instance list.

Instance summary for i-04bd2f06c2a61abb4 (MyPublicEC2Server) Info

Updated less than a minute ago

Public IPv4 address copied

Instance ID: i-04bd2f06c2a61abb4 (MyPublicEC2Server) 54.162.212.109 open address

Instance state: **Running**

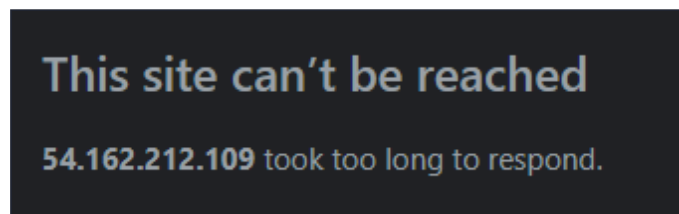
Instance type: t2.micro

Private IPv4 addresses: 10.0.1.208

Private IPv4 DNS: ip-10-0-1-208.ec2.internal

VPC ID: vpc-08cd445d44e195fc7 (MyVPC)

- From the Description tab, copy the **IPv4 Public IP**.
- Now paste this IP into your web browser and click [Enter]
- You will see the following page:



Note: This is because the Custom NACL which is attached to your Public subnet restricts both **INBOUND** and **OUTBOUND** traffic.

2. Private EC2 Instance:

- Since the Public NACL restricts all traffic, you won't be able to SSH into the public EC2 Instance to ping the Private Instance.
- Next, we are going to solve this.

Task 13: Adding Rules to Custom NACL (MyPublicNACL)

1. Navigate to **VPC** under the **Services** menu. Click on **Network ACLs** under **Security**.
2. Select **MyPublicNACL** from the list.
3. In the Inbound rules, click **Edit inbound rules**
4. Add the following rules:

- **HTTP** click on **Add rules**,
 - Rule#: Enter **100**
 - Type: Choose **HTTP (80)**
 - Source: Enter **0.0.0.0/0**

- Allow / Deny: Select Allow
- For **ALL ICMP- IPv4**, click on **Add rules**,
 - Rule# : Enter **150**
 - Type: Choose **ALL ICMP - IPv4**
 - Source: Enter **0.0.0.0/0**
 - Allow / Deny: Select Allow
- For **SSH**, click on **Add rules**,
 - Rule# : Enter **200**
 - Type: Choose **SSH (22)**
 - Source: Enter **0.0.0.0/0**
 - Allow / Deny: Select Allow

Rule number Info	Type Info	Protocol Info	Port range Info	Source Info	Allow/Deny Info
100	HTTP (80) ▼	TCP (6) ▼	80	0.0.0.0/0	Allow ▼
150	ALL ICMP - IPv4 ▼	ICMP (1) ▼	All	0.0.0.0/0	Allow ▼
200	SSH (22) ▼	TCP (6) ▼	22	0.0.0.0/0	Allow ▼

- Click on **Save changes**

5. In the **Outbound rules** Tab, Click Edit outbound rules

6. Add the following rules:

- **Custom Port** is already available,
 - Rule# : Enter **100**
 - Type: Choose **Custom TCP Rule**
 - Port Range: Enter **1024 - 65535**
 - Source: Enter **0.0.0.0/0**
 - Allow / Deny: Select **Allow**
- For **ALL ICMP- IPv4**, click on **Add rules**,
 - Rule# : Enter **150**
 - Type: Choose **ALL ICMP - IPv4**
 - Source: Enter **0.0.0.0/0**

- Allow / Deny: Select Allow
- For **SSH**, click on **Add rules** ,
 - Rule# : Enter **200**
 - Type: Choose **SSH (22)**
 - Source: Enter **0.0.0.0/0**
 - Allow / Deny: Select Allow

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
150	All ICMP - IPv4	ICMP (1)	All	0.0.0.0/0	Allow
200	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow

- Click on **Save**

Task 14: Testing Both EC2 instances

1. We will try to ping the Private EC2 from the Public EC2 instance.

- SSH into EC2 Instance
 - Please follow the steps in [SSH into EC2 Instance](#).
- Once connected to the server:
 - Change to root user:

```
sudo su
```



- Copy the Private IP of **MyPrivateEC2Server** from the Description tab.

Instance summary for i-06fb2249d846c1237 (MyPrivateEC2Server) Info			Refresh	Connect	Instance state ▼
Updated less than a minute ago					
Instance ID i-06fb2249d846c1237 (MyPrivateEC2Server)	Public IPv4 address -	Private IPv4 addresses 10.0.2.161			
Instance state Running	Public IPv4 DNS -	Private IPv4 DNS ip-10-0-2-161.ec2.internal			
Instance type	Elastic IP addresses	VPC ID			

- Ping to the Private Instance using the Private IPv4
 - Example:

```
[root@ip-10-0-1-208 ec2-user]# ping 10.0.2.161
PING 10.0.2.161 (10.0.2.161) 56(84) bytes of data.
64 bytes from 10.0.2.161: icmp_seq=1 ttl=255 time=0.653 ms
64 bytes from 10.0.2.161: icmp_seq=2 ttl=255 time=0.656 ms
64 bytes from 10.0.2.161: icmp_seq=3 ttl=255 time=0.657 ms
64 bytes from 10.0.2.161: icmp_seq=4 ttl=255 time=0.762 ms
64 bytes from 10.0.2.161: icmp_seq=5 ttl=255 time=0.679 ms
64 bytes from 10.0.2.161: icmp_seq=6 ttl=255 time=0.796 ms
64 bytes from 10.0.2.161: icmp_seq=7 ttl=255 time=0.610 ms
64 bytes from 10.0.2.161: icmp_seq=8 ttl=255 time=0.776 ms
64 bytes from 10.0.2.161: icmp_seq=9 ttl=255 time=0.657 ms
64 bytes from 10.0.2.161: icmp_seq=10 ttl=255 time=0.610 ms
64 bytes from 10.0.2.161: icmp_seq=11 ttl=255 time=0.661 ms
64 bytes from 10.0.2.161: icmp_seq=12 ttl=255 time=0.623 ms
64 bytes from 10.0.2.161: icmp_seq=13 ttl=255 time=0.645 ms
```

- Press [Ctrl] + C again to cancel the process instead of pausing it.
- Note: You were able to do these tasks because we added NACL Rules.

Do you know?

By participating in the AWS VPC NACL Lab, users acquire a range of practical skills and knowledge, including the ability to create NACLs, define rule sets for specific subnets, evaluate traffic patterns, implement allow and deny rules, and analyze the impact of NACL configurations on network communication. Additionally, participants gain insight into optimizing NACL configurations for different use cases, securing sensitive workloads, and mitigating potential security risks within their AWS VPCs.

Task 15: Validation Test

1. Once the lab steps are completed, please click on the **Validation button** on the Right side panel.
2. This will validate the resources in the AWS account and shows you whether you have completed this lab successfully or not.
3. Sample output :

Completion and Conclusion

- You have created a VPC using the VPC Wizard.
- You have created an Internet Gateway.
- You have created a private and public subnet for the VPC.
- You have created and associated Route tables.
- You have added routes to the Route table
- You have launched some EC2 instances into the Public and Private subnets.
- You have created a Custom NACL.
- You have associated the NACL with the subnets.
- You added inbound and outbound rules to the custom NACL.
- You have tested our VPC.

End Lab

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs dashboard.

