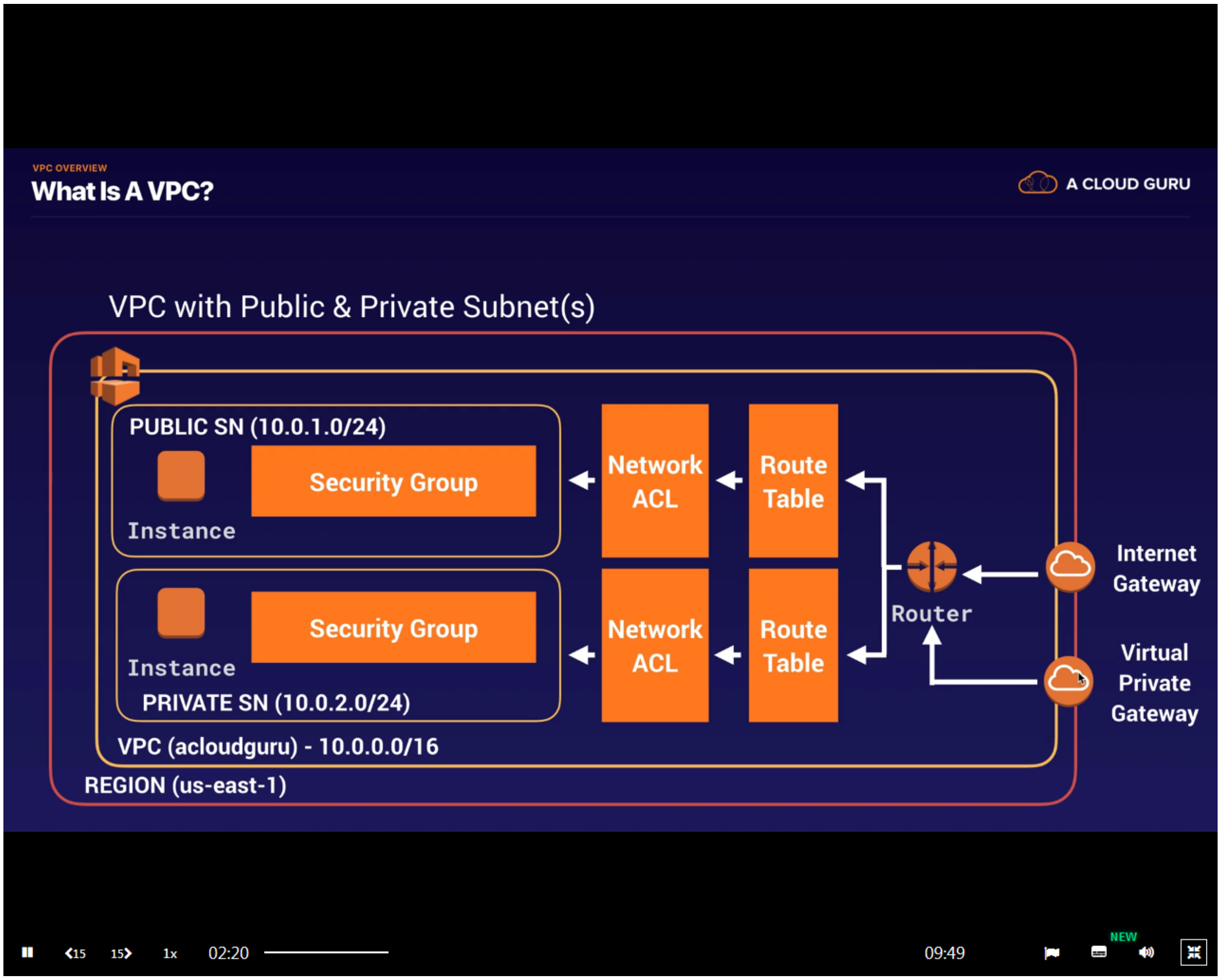
Tackle the NACLs

Network Access Control Lists & Security Groups Revisited



(Notice our instances are behind our security groups. Notice our subnets are behind our Network Access Control Lists…so Security Groups are for security at the instance level and NACLS are for security at the subnet level)

Beginning from where we left off in the last lesson. (We now have OurNewCustomVPC with a public and private subnet. We have an instance in each subnet. We have a NAT Gateway in the public subnet which gives the instance in our private subnet a route out to the internet)

Go to Click VPC

Click Network ACLs in the left margin

✓Select the ACL that is the default ACL that was create by default when we created OurNewCustomVPC. (It currently has 2 subnets associated with it)

Click the **inbound rules** tab

Click **Edit inbound rules**

You will see the following:

Rule#	Type	Protocol	Port Range	Source	Allow/Deny
100	All Traffic	All	All	0.0.0.0/0	Allow
101	All Traffic	All	All	::/0	Allow
*	All Traffic	All	All	0.0.0.0/0	Deny
*	All Traffic	All	All	::/0	Deny

The above rules mean that **all** inbound traffic is allowed. Don't be confused by the * rules. Those don't even get evaluated since rule 100 and 101 are evaluated first.

★The default NACL **allows all** outbound and inbound traffic. (A manually created NACL denies all traffic)

★NACLs contain a numbered list of rules that gets evaluated in order from lowest to highest.

★Amazon recommends using increments of 100. (So if we were to add another iPv4 rule we would call it rule 200. Anf if we were to add another IPv6 rules we would call it rule 201)

Now click the **outbound rules** tab

You will see the following:

Rule#	Type	Protocol	Port Range	Source	Allow/Deny
100	All Traffic	All	All	0.0.0.0/0	Allow
101	All Traffic	All	All	::/0	Allow
*	All Traffic	All	All	0.0.0.0/0	Deny
*	All Traffic	All	All	::/0	Deny

The above rules mean that **all** outbound traffic is allowed. Don't be confused by the * rules. Those don't even get evaluated since rule 100 and 101 are evaluated first.

Now click the **Create network ACL** button towards the top of the screen.

Name tag: PublicSubnetNACL ←named it this because we will be associating it with our public subnet

VPC* OurNewCustomVPC

Click **Create**

✓PublicSubnetNACL ← Select PublicSubnetNACL (which is NOT a default NACL, it is the NACL that we just created)

Click the **inbound rules** tab and you will see the following

*	All Traffic	All	All	0.0.0.0/0	Deny
*	All Traffic	All	All	::/0	Deny

Click the **outbound rules** tab and you will see the following

*	All Traffic	All	All	0.0.0.0/0	Deny
*	All Traffic	All	All	::/0	Deny

★Whenever you yourself create a new network ACL (**NOT** a default NACL), the rules will state that **everything** is denied by default for both inbound and outbound traffic. (This is in complete contrast to a default NACL which allows both inbound and outbound by default)

Let's SSH into our instance located in our PUBLIC subnet. (If you are still SSH'd into the instance in our Private subnet from the lesson previous you can simply type exit (and hit enter). Then once again type exit (and hit enter). You should now be in the command line for instance located in your Public subnet once again. Otherwise, go to the console and click **Services** and click **EC2.** Click **Running instances.**

✓WebServerinPublicSubnet ← select WebServerinPublicSubnet

Click the **Connect** button

✓EC2 Instance Connect (browser-based SSH connection)

Click **Connect**

Once you are SSH'd into your public instance type the following:

sudo su (and hit enter)

clear (and hit enter)

service httpd status (and hit enter)

(It will say not found)

Now type the following:

yum install httpd –y (and hit enter)(this will install Apache so that we can create a webpage on this instance/server)

clear (and hit enter)

chkconfig httpd on (and hit enter)

service httpd start (and hit enter)

cd /var/www/html (and hit enter)

ls (and hit enter)(that is a lower case L by the way and not a capital i)(nothing will be listed)

nano index.html (and hit enter)(we are creating an index.html file)

type the following into the file:

<html><h1>Tackle the NACLS</h1></html>

(then hit Ctrl + X at the same time in order to exit)

Then type y (and hit enter in order to save the file)

Then hit enter again.

Back at the regular command line type the following:

ls (and hit enter)(you will see the index.html file we just created)

cat index.html (and hit enter)(this will show us the html code inside the file)

Minimize your command line and go to the AWS console to retrieve the public IP address of your public instance.

Click **Services**, click **EC2**, click **running instances**, and select ✓ WebServerinPublicSubnet

Scroll down to the description tab and copy the **IPv4 Public IP** which in this case is 54.255.215.253 ← yours will be different so be careful of that

Plug the IP into a new url browser

It will say I am Spartucus!

Now return to the AWS console

Click **Services**, and then click **VPC**

Click **Network ACLs**

✓PublicSubnetNACL ← select PublicSubnetNACL (which is the NACL that we created)

Then click the **Subnet Associations** tab

✓10.0.1.0-ap-southeast-1a-PublicSubnet ←10.0.1.0-ap-southeast-1a-PublicSubnet

And then click **Edit**

(Now our public subnet is associate with the PublicSubnetNACL, and it is no longer associate with our Default NACL that was created when we first created our VPC)

Enter your public IP into a new browser URL and it will say "This site can't be reached"

The reason for this is because our PublicSubnetNACL is denying IPv4 and IPv6.

Let's edit the rules of our PublicSubnetNACL.

In the console click **services** and click **VPC** and click **Network ACLs** in the left margin

✓PublicSubnetNACL ← select PublicSubnetNACL

Click the **inbound rules** tab

Click **Edit inbound rules**

	Rule #	Type	Protocol	Port Range	Source	Allow/Deny
Add this rule→	100	HTTP(80)	TCP(6)	80	0.0.0.0/0	Allow
Add this rule→	200	HTTPS(443)	TCP(6)	443	0.0.0.0/0	Allow
Add this rule→	300	SSH(22)	TCP(6)	22	0.0.0.0/0	Allow
Already there →	*	All Traffic	All	All	0.0.0.0/0	Deny
Already there →	*	All Traffic	All	All	::/0	Deny

Click the **outbound rules** tab

Click **Edit outbound rules**

	Rule#	Type	Protocol	Port Range	Destination	Allow/Deny
Add this rule→	100	HTTP(80)	TCP(6)	80	0.0.0.0/0	Allow
Add this rule→	200	HTTPS(443)	TCP(6)	443	0.0.0.0/0	Allow
Add this rule→	300	Custom TCP Rule	TCP(6)	1024-65535	0.0.0.0/0	Allow
Already there →	*	All Traffic	All	All	0.0.0.0/0	Deny
Already there →	*	All Traffic	All	All	::/0	Deny

The reason we want outbound rule with port range 1024-65535 is because a NAT Gateway uses ports 1024-65535 (also known as ephemeral ports)

Now let's go back to our other browser tab and enter our public IP of 54.255.215.253

It will now once again say "I am Spartacus!"

We are connecting once again over port 80 so our site is now visible once again.

Let's do another experiment with our NACL to understand how the rule #'s work.

Go to the console and click **Services** and then click **VPC** then click **Network ACLS** in the left margin

✓PublicSubnetNACL ← select PublicSubnetNACL

Click the **Inbound Rules** Tab and click **Edit inbound rules**

	Rule #	Type	Protocol	Port Range	Source	Allow/Deny
Already there	→100	HTTP(80)	TCP(6)	80	0.0.0.0/0	Allow
Already there	→200	HTTPS(443)	TCP(6)	443	0.0.0.0/0	Allow
Already there	→300	SSH(22)	TCP(6)	22	0.0.0.0/0	Allow
Add this	→400	HTTP	TCP(6)	80	172.88.102.35/32	Deny
Already there	→*	All Traffic	All	All	0.0.0.0/0	Deny
Already there	→ *	All Traffic	All	All	::/0	Deny

Add rule 400 above. In the source put the ip address for your computer. Mine is 172.88.102.35/32

Now let's go back to our other browser tab and enter our public IP of 54.255.215.253

It will now once again say "I am Spartacus!" The reason our website is still visible is because Rule 100 supercedes rule 400. If we wanted to deny our own IP we would have to put it before rule 100 which is allowing 0.0.0.0/0 on port 80.

So let's change the rule 400 that we created to rule 99.

	Rule #	Type	Protocol	Port Range	Source	Allow/Deny
Add this	→99	HTTP(80)	TCP(6)	80	172.88.102.35/32	Deny
Already there	→100	HTTP(80)	TCP(6)	80	0.0.0.0/0	Allow
Already there	→200	HTTPS(443)	TCP(6)	443	0.0.0.0/0	Allow
Already there	→300	SSH(22)	TCP(6)	22	0.0.0.0/0	Allow
Already there	→*	All Traffic	All	All	0.0.0.0/0	Deny
Already there	→ *	All Traffic	All	All	::/0	Deny

Now let's go back to our other browser tab and enter our public IP of 54.255.215.253

And it will now say our site cannot be reached. The reason is because the deny rule #99 for port 80 will be evaluated before the allow rule #100 for port 80 since the rules are always evaluated in order.

Let’s go ahead and delete rule 99 and it was just an experiment to demonstrate how the rules are evaluated.

And before we end the lesson let’s add an inbound rule (to match our outbound rule) to open our ephemeral ports 1023-65535 on our PublicSubnetNACL. This way we can update our operating system on our public instance, or download software from the internet, etc.

✓PublicSubnetNACL ← select PublicSubnetNACL

Click the **Inbound Rules** Tab and click **Edit inbound rules**

Rule #	Type	Protocol	Port Range	Source	Allow/Deny
Already there →100	HTTP(80)	TCP(6)	80	0.0.0.0/0	Allow
Already there →200	HTTPS(443)	TCP(6)	443	0.0.0.0/0	Allow
Already there →300	SSH(22)	TCP(6)	22	0.0.0.0/0	Allow
Add this rule→ 400	Custom TCP Rule	TCP(6)	1024-65535	0.0.0.0/0	Allow
Already there →*****	All Traffic	All	All	0.0.0.0/0	Deny
Already there → *****	All Traffic	All	All	::/0	Deny

Click **Save**

Now click **service** click **ec2** click **running instances** and select our WebServerInPublicSubnet and click the **connect** button to SSH in to it.

Type the following:

sudo su (and hit enter)

clear (and hit enter)

yum update -y (and hit enter) (our operating system will update)

Mission Complete. The next lesson will be on VPC Endpoints and will pick up from this exact point. If you do not have time to continue on then delete your VPC and delete your instances. Below are important concepts and exam questions pertaining to NACLs.

★Lesson Review

-When we created our VPC a NACL was created by default and it is called our default network ACL.

-Every time we add a subnet to our VPC it is going to be associated with our Default NACL

-You can then associate the subnet with a new Network ACL but a subnet itself can only be associated with one NACL at any given time.

-You can associate a NACL with multiple subnets but a subnet can only be associated with one NACL at a time. When you associate a NACL with a subnet the previous association is removed.

-When we make changes the rules take effect immediately.

-Rules are evaluated in order.

-NACLs have inbound & outbound rules and can either allow or deny traffic.

-NACLS are stateless. Responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa)

-NACLS are evaluated before Security groups for inbound traffic. So if you deny a port on your NACL then it will never even reach your security group.

★NACLs (Network Access Control Lists) Cheat Sheet

-Network Access Control List is commonly known as NACL

-VPCs are automatically given a default NACL which **allows all** outbound and inbound traffic.

-However, When you create a NACL it will **deny** all traffic by default

-Each subnet within a VPC must be associated with a NACL

-Subnets can only be associated with 1 NACL at a time. Associating a subnet with a new NACL will remove the previous association.

-If a NACL is not explicitly associated with a subnet, the subnet will automatically be associated with the default NACL.

-NACL has inbound and outbound rules (just like Security Groups)

-Rule can either **allow** or **deny** traffic (unlike Security Groups which can only allow)

-NACLs are **STATELESS** (any allowed inbound traffic is also allowed outbound)

-NACLs contain a numbered list of rules that gets evaluated in order from lowest to highest

-If you needed to block a single IP address you could via NACLs (Security Groups cannot deny)

Practice Questions

Question
Your company hosts a popular web application that connects to an Amazon RDS MySQL DB instances running in a private subnet created with the default ACL settings. Your security department has identified a DoS attack originating from a suspicious IP address. How can you protect the subnets from this attack?

A) Change the Outbound NACL to deny access from the suspicious IP address.
B) Change the Inbound NACL to deny access from the suspicious IP address.
C) Change the Inbound Security Groups to deny access from the suspicious IP address.
D) Change the Outbound Security Groups to deny access from the suspicious IP address.

Explanation:
A network access control list (ACL) is another layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. Network ACLs and security groups apply different types of filtering and can be used together. Answer is B.

Question
You recently set up a website for customers to access over the Internet, but upon navigating to the URL, you keep getting a 'Connection timed out' error message. Which of the following answers will solve the problem?

A) The inbound and outbound rules in the security group are edited to include an HTTP connection.
B) The inbound and outbound rules in the network access control list (network ACL) are edited to support a HTTP connection.
C) Keepalives is enabled.
D) The website domain’s health check is refreshed to generate a Healthy status.

Explanation:
If you get a 'Connection timed out' error message when navigating to your website, you have to check the security group rules. You need rules that allow inbound and outbound traffic from the website’s address on the proper ports. In this case, since customers need to connect from the Internet, you will have to set inbound and outbound rules in the security group for an HTTP connection, which is through port 80. The Network Access Control List (NACLs) must also allow traffic to come in on port-80 and return back out on the ephemeral web ports. Enabling keepalives is for resolving the 'Server unexpectedly closed network connection' error, not 'Connection timed out'. Domain health checks is not a valid answer, since there’s no option to 'refresh' health checks. Answer is A and B.