

Network ACLs

A *network access control list (ACL)* is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC. For more information about the differences between security groups and network ACLs, see [Comparison of Security Groups and Network ACLs \(VPC_Security.html#VPC_Security_Comparison\)](#) .

Contents

- [Network ACL Basics \(#nacl-basics\)](#)
- [Network ACL Rules \(#nacl-rules\)](#)
- [Default Network ACL \(#default-network-acl\)](#)
- [Custom Network ACL \(#custom-network-acl\)](#)
- [Custom Network ACLs and Other AWS Services \(#nacl-other-services\)](#)
- [Ephemeral Ports \(#nacl-ephemeral-ports\)](#)
- [Working with Network ACLs \(#nacl-tasks\)](#)
- [Example: Controlling Access to Instances in a Subnet \(#nacl-examples\)](#)
- [API and Command Overview \(#nacl-api-cli\)](#)

Network ACL Basics

The following are the basic things that you need to know about network ACLs:

- Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
- You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
- Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
- You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
- A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest number that you can use for a rule is 32766. We recommend that you start by creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.

- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

For more information, see [Amazon VPC Limits \(amazon-vpc-limits.html\)](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#amazon-vpc-limits.html) .

Network ACL Rules

You can add or remove rules from the default network ACL, or create additional network ACLs for your VPC. When you add or remove rules from a network ACL, the changes are automatically applied to the subnets it's associated with.

The following are the parts of a network ACL rule:

- **Rule number.** Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.
- **Type.** The type of traffic; for example, SSH. You can also specify all traffic or a custom range.
- **Protocol.** You can specify any protocol that has a standard protocol number. For more information, see [Protocol Numbers \(http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml\)](http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml) . If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.
- **Port range.** The listening port or port range for the traffic. For example, 80 for HTTP traffic.
- **Source.** [Inbound rules only] The source of the traffic (CIDR range).
- **Destination.** [Outbound rules only] The destination for the traffic (CIDR range).
- **Allow/Deny.** Whether to ALLOW or DENY the specified traffic.

Default Network ACL

The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated. Each network ACL also includes a rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

The following is an example default network ACL for a VPC that supports IPv4 only.

Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW

*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

If you create a VPC with an IPv6 CIDR block or if you associate an IPv6 CIDR block with your existing VPC, we automatically add rules that allow all IPv6 traffic to flow in and out of your subnet. We also add rules whose rule numbers are an asterisk that ensures that a packet is denied if it doesn't match any of the other numbered rules. You can't modify or remove these rules. The following is an example default network ACL for a VPC that supports IPv4 and IPv6.

Note

If you've modified your default network ACL's inbound rules, we do not automatically add an ALLOW rule for inbound IPv6 traffic when you associate an IPv6 block with your VPC. Similarly, if you've modified the outbound rules, we do not automatically add an ALLOW rule for outbound IPv6 traffic.

Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
101	All IPv6 traffic	All	All	::/0	ALLOW
*	All traffic	All	All	0.0.0.0/0	DENY
*	All IPv6 traffic	All	All	::/0	DENY
Outbound					
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	ALLOW
101	All IPv6 traffic	All	All	::/0	ALLOW
*	All traffic	All	All	0.0.0.0/0	DENY
*	All IPv6 traffic	All	All	::/0	DENY

Custom Network ACL

The following table shows an example of a custom network ACL for a VPC that supports IPv4 only. It includes rules that allow HTTP and HTTPS traffic in (inbound rules 100 and

110). There's a corresponding outbound rule that enables responses to that inbound traffic (outbound rule 120, which covers ephemeral ports 32768-65535). For more information about how to select the appropriate ephemeral port range, see [Ephemeral Ports \(vpc-network-acls.html#nacl-ephemeral-ports\)](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports).

The network ACL also includes inbound rules that allow SSH and RDP traffic into the subnet. The outbound rule 120 enables responses to egress the subnet.

The network ACL has outbound rules (100 and 110) that allow outbound HTTP and HTTPS traffic out of the subnet. There's a corresponding inbound rule that enables responses to that outbound traffic (inbound rule 140, which covers ephemeral ports 32768-65535).

Note

Each network ACL includes a default rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other rules, it's denied. You can't modify or remove this rule.

Inbound						
Rule #	Type	Protocol	Port Range	Source	Allow/Deny	Comments
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	Allows inbound HTTP traffic from any IPv4 address.
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	Allows inbound HTTPS traffic from any IPv4 address.
120	SSH	TCP	22	192.0.2.0/24	ALLOW	Allows inbound SSH traffic from your home network's public IPv4 address range (over the Internet gateway).
130	RDP	TCP	3389	192.0.2.0/24	ALLOW	Allows inbound RDP traffic to the web servers from your home network's public IPv4 address range (over the Internet gateway).
140	Custom TCP	TCP	32768-65535	0.0.0.0/0	ALLOW	Allows inbound return IPv4 traffic from the Internet (that is, for requests that originate in the subnet).

						This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports (vpc-network-acls.html#nacl-ephemeral-ports) .
*	All traffic	All	All	0.0.0.0/0	DENY	Denies all inbound IPv4 traffic not already handled by a preceding rule (not modifiable).
Outbound						
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny	Comments
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	Allows outbound IPv4 HTTP traffic from the subnet to the Internet.
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	Allows outbound IPv4 HTTPS traffic from the subnet to the Internet.
120	Custom TCP	TCP	32768-65535	0.0.0.0/0	ALLOW	Allows outbound IPv4 responses to clients on the Internet (for example, serving web pages to people visiting the web servers in the subnet). This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports (vpc-network-acls.html#nacl-ephemeral-ports) .
*	All traffic	All	All	0.0.0.0/0	DENY	Denies all outbound IPv4 traffic not already handled by a preceding rule (not modifiable).

As a packet comes to the subnet, we evaluate it against the ingress rules of the ACL the subnet is associated with (starting at the top of the list of rules, and moving to the bottom). Here's how the evaluation goes if the packet is destined for the SSL port (443). The packet doesn't match the first rule evaluated (rule 100). It does match the second rule (110), which allows the packet into the subnet. If the packet had been destined for port 139 (NetBIOS), it doesn't match any of the rules, and the * rule ultimately denies the packet.

You might want to add a DENY rule in a situation where you legitimately need to open a wide range of ports, but there are certain ports within that range you want to deny. Just make sure to place the DENY rule earlier in the table than the rule that allows the wide range of port traffic.

You add ALLOW rules depending on your use case; for example, a rule that allows outbound TCP and UDP access on port 53 for DNS resolution. For every rule that you add, ensure that there is a corresponding inbound or outbound rule that allows response traffic.

The following table shows the same example of a custom network ACL for a VPC that has an associated IPv6 CIDR block. This network ACL includes rules for all IPv6 HTTP and HTTPS traffic. In this case, new rules were inserted between the existing rules for IPv4 traffic; however, you can also add the rules as higher number rules after the IPv4 rules. IPv4 and IPv6 traffic are separate; therefore, none of the rules for the IPv4 traffic apply to the IPv6 traffic.

Inbound						
Rule #	Type	Protocol	Port Range	Source	Allow/Deny	Comments
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	Allows inbound HTTP traffic from any IPv4 address.
105	HTTP	TCP	80	::/0	ALLOW	Allows inbound HTTP traffic from any IPv6 address.
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	Allows inbound HTTPS traffic from any IPv4 address.
115	HTTPS	TCP	443	::/0	ALLOW	Allows inbound HTTPS traffic from any IPv6 address.
120	SSH	TCP	22	192.0.2.0/24	ALLOW	Allows inbound SSH

						traffic from your home network's public IPv4 address range (over the Internet gateway).
130	RDP	TCP	3389	192.0.2.0/24	ALLOW	Allows inbound RDP traffic to the web servers from your home network's public IPv4 address range (over the Internet gateway).
140	Custom TCP	TCP	32768-65535	0.0.0.0/0	ALLOW	<p>Allows inbound return IPv4 traffic from the Internet (that is, for requests that originate in the subnet).</p> <p>This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports (vpc-network-acls.html#nacl-ephemeral-ports) .</p>
145	Custom TCP	TCP	32768-65535	::/0	ALLOW	<p>Allows inbound return IPv6 traffic from the Internet (that is, for requests that originate in the subnet).</p> <p>This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports (vpc-network-acls.html#nacl-ephemeral-ports) .</p>
*	All traffic	All	All	0.0.0.0/0	DENY	Denies all inbound IPv4 traffic not already handled by a preceding rule (not modifiable).

*	All traffic	All	All	::/0	DENY	Denies all inbound IPv6 traffic not already handled by a preceding rule (not modifiable).
Outbound						
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny	Comments
100	HTTP	TCP	80	0.0.0.0/0	ALLOW	Allows outbound IPv4 HTTP traffic from the subnet to the Internet.
105	HTTP	TCP	80	::/0	ALLOW	Allows outbound IPv6 HTTP traffic from the subnet to the Internet.
110	HTTPS	TCP	443	0.0.0.0/0	ALLOW	Allows outbound IPv4 HTTPS traffic from the subnet to the Internet.
115	HTTPS	TCP	443	::/0	ALLOW	Allows outbound IPv6 HTTPS traffic from the subnet to the Internet.
120	Custom TCP	TCP	32768-65535	0.0.0.0/0	ALLOW	Allows outbound IPv4 responses to clients on the Internet (for example, serving web pages to people visiting the web servers in the subnet). This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports (vpc-network-acls.html#nacl-ephemeral-ports) .
125	Custom TCP	TCP	32768-65535	::/0	ALLOW	Allows outbound IPv6 responses to clients on the Internet (for example, serving web pages to people visiting

						the web servers in the subnet). This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports (vpc-network-acls.html#nacl-ephemeral-ports) .
*	All traffic	All	All	0.0.0.0/0	DENY	Denies all outbound IPv4 traffic not already handled by a preceding rule (not modifiable).
*	All traffic	All	All	::/0	DENY	Denies all outbound IPv6 traffic not already handled by a preceding rule (not modifiable).

Custom Network ACLs and Other AWS Services

If you create a custom network ACL, be aware of how it might affect resources that you create using other AWS services.

With Elastic Load Balancing, if the subnet for your back-end instances has a network ACL in which you've added a DENY rule for all traffic with a source of `0.0.0.0/0` or the subnet's CIDR, then your load balancer can't carry out health checks on the instances. For more information about the recommended network ACL rules for your load balancers and back-end instances, see [Network ACLs for Load Balancers in a VPC](https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-security-groups.html#elb-vpc-nacl) (<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-security-groups.html#elb-vpc-nacl>) in the *User Guide for Classic Load Balancers*.

Ephemeral Ports

The example network ACL in the preceding section uses an ephemeral port range of 32768-65535. However, you might want to use a different range for your network ACLs depending on the type of client that you're using or with which you're communicating.

The client that initiates the request chooses the ephemeral port range. The range varies depending on the client's operating system.

- Many Linux kernels (including the Amazon Linux kernel) use ports 32768-61000
- Requests originating from Elastic Load Balancing use ports 1024-65535
- Windows operating systems through Windows Server 2003 use ports 1025-5000
- Windows Server 2008 and later versions use ports 49152-65535
- A NAT gateway uses ports 1024-65535
- AWS Lambda functions use ports 1024-65535

For example, if a request comes into a web server in your VPC from a Windows XP client on the Internet, your network ACL must have an outbound rule to enable traffic destined for ports 1025-5000.

If an instance in your VPC is the client initiating a request, your network ACL must have an inbound rule to enable traffic destined for the ephemeral ports specific to the type of instance (Amazon Linux, Windows Server 2008, and so on).

In practice, to cover the different types of clients that might initiate traffic to public-facing instances in your VPC, you can open ephemeral ports 1024-65535. However, you can also add rules to the ACL to deny traffic on any malicious ports within that range. Ensure that you place the DENY rules earlier in the table than the ALLOW rules that open the wide range of ephemeral ports.

Working with Network ACLs

The following tasks show you how to work with network ACLs using the Amazon VPC console.

Tasks

- [Determining Network ACL Associations \(#ACLSubnet\)](#)
- [Creating a Network ACL \(#CreateACL\)](#)
- [Adding and Deleting Rules \(#Rules\)](#)
- [Associating a Subnet with a Network ACL \(#NetworkACL\)](#)
- [Disassociating a Network ACL from a Subnet \(#DisassociateNetworkACL\)](#)
- [Changing a Subnet's Network ACL \(#ChangeNetworkACL\)](#)
- [Deleting a Network ACL \(#DeleteNetworkACL\)](#)

Determining Network ACL Associations

You can use the Amazon VPC console to determine the network ACL that's associated with a subnet. Network ACLs can be associated with more than one subnet, so you can also determine the subnets that are associated with a network ACL.

To determine which network ACL is associated with a subnet

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Subnets**, and then select the subnet.

The network ACL associated with the subnet is included in the **Network ACL** tab, along with the network ACL's rules.

To determine which subnets are associated with a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**. The **Associated With** column indicates the number of associated subnets for each network ACL.
3. Select a network ACL.
4. In the details pane, choose **Subnet Associations** to display the subnets associated with the network ACL.

Creating a Network ACL

You can create a custom network ACL for your VPC. By default, a network ACL that you create blocks all inbound and outbound traffic until you add rules, and is not associated with a subnet until you explicitly associate it with one.

To create a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**.
3. Choose **Create Network ACL**.
4. In the **Create Network ACL** dialog box, optionally name your network ACL, and then select the ID of your VPC from the **VPC** list, and choose **Yes, Create**.

Adding and Deleting Rules

When you add or delete a rule from an ACL, any subnets associated with the ACL are subject to the change. You don't have to terminate and relaunch the instances in the subnet; the changes take effect after a short period.

If you're using the Amazon EC2 API or a command line tool, you can't modify rules; you can only add and delete rules. If you're using the Amazon VPC console, you can modify the entries for existing rules (the console removes the rule and adds a new rule for you). If you

need to change the order of a rule in the ACL, you must add a new rule with the new rule number, and then delete the original rule.

To add rules to a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**.
3. In the details pane, choose either the **Inbound Rules** or **Outbound Rules** tab, depending on the type of rule that you need to add, and then choose **Edit**.
4. In **Rule #**, enter a rule number (for example, 100). The rule number must not already be used in the network ACL. We process the rules in order, starting with the lowest number.

Tip

We recommend that you leave gaps between the rule numbers (such as 100, 200, 300), rather than using sequential numbers (101, 102, 103). This makes it easier add a new rule without having to renumber the existing rules.

5. Select a rule from the **Type** list. For example, to add a rule for HTTP, choose **HTTP**. To add a rule to allow all TCP traffic, choose **All TCP**. For some of these options (for example, HTTP), we fill in the port for you. To use a protocol that's not listed, choose **Custom Protocol Rule**.
6. (Optional) If you're creating a custom protocol rule, select the protocol's number and name from the **Protocol** list. For more information, see [IANA List of Protocol Numbers](http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml) (<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>) .
7. (Optional) If the protocol you've selected requires a port number, enter the port number or port range separated by a hyphen (for example, 49152-65535).
8. In the **Source** or **Destination** field (depending on whether this is an inbound or outbound rule), enter the CIDR range that the rule applies to.
9. From the **Allow/Deny** list, select **ALLOW** to allow the specified traffic or **DENY** to deny the specified traffic.
10. (Optional) To add another rule, choose **Add another rule**, and repeat steps 4 to 9 as required.
11. When you are done, choose **Save**.

To delete a rule from a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**, and then select the network ACL.
3. In the details pane, select either the **Inbound Rules** or **Outbound Rules** tab, and then choose **Edit**. Choose **Remove** for the rule you want to delete, and then choose **Save**.

Associating a Subnet with a Network ACL

To apply the rules of a network ACL to a particular subnet, you must associate the subnet with the network ACL. You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL. Any subnet not associated with a particular ACL is associated with the default network ACL by default.

To associate a subnet with a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**, and then select the network ACL.
3. In the details pane, on the **Subnet Associations** tab, choose **Edit**. Select the **Associate** check box for the subnet to associate with the network ACL, and then choose **Save**.

Disassociating a Network ACL from a Subnet

You can disassociate a custom network ACL from a subnet — by doing so, the subnet is then automatically associated with the default network ACL.

To disassociate a subnet from a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**, and then select the network ACL.
3. In the details pane, choose the **Subnet Associations** tab.
4. Choose **Edit**, and then deselect the **Associate** check box for the subnet. Choose **Save**.

Changing a Subnet's Network ACL

You can change the network ACL that's associated with a subnet. For example, when you create a subnet, it is initially associated with the default network ACL. You might want to instead associate it with a custom network ACL that you've created.

After changing a subnet's network ACL, you don't have to terminate and relaunch the instances in the subnet; the changes take effect after a short period.

To change a subnet's network ACL association

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Subnets**, and then select the subnet.
3. Choose the **Network ACL** tab, and then choose **Edit**.
4. Select the network ACL to associate the subnet with from the **Change to** list, and then choose **Save**.

Deleting a Network ACL

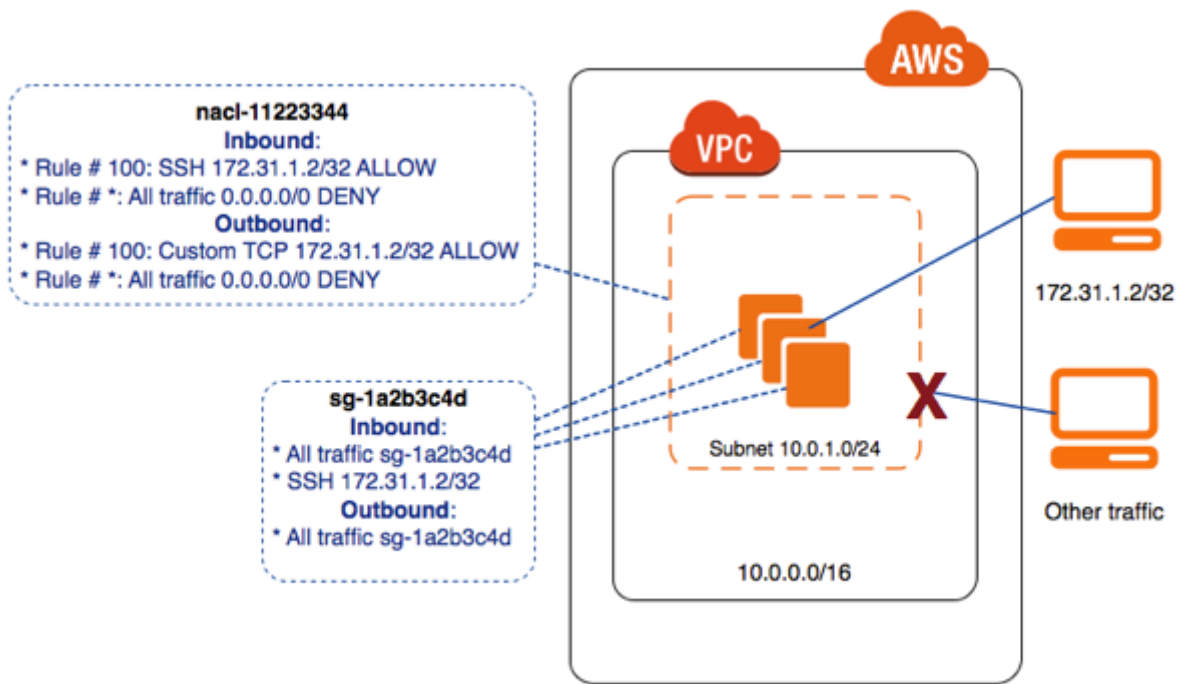
You can delete a network ACL only if there are no subnets associated with it. You can't delete the default network ACL.

To delete a network ACL

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/> (<https://console.aws.amazon.com/vpc/>) .
2. In the navigation pane, choose **Network ACLs**.
3. Select the network ACL, and then choose **Delete**.
4. In the confirmation dialog box, choose **Yes, Delete**.

Example: Controlling Access to Instances in a Subnet

In this example, instances in your subnet can communicate with each other, and are accessible from a trusted remote computer. The remote computer may be a computer in your local network or an instance in a different subnet or VPC that you use to connect to your instances to perform administrative tasks. Your security group rules and network ACL rules allow access from the IP address of your remote computer (172.31.1.2/32). All other traffic from the Internet or other networks is denied.



All instances use the same security group (sg-1a2b3c4d), with the following rules.

Inbound Rules				
Protocol Type	Protocol	Port Range	Source	Comments
All traffic	All	All	sg-1a2b3c4d	Enables instances associated with the same security group to communicate with each other.
SSH	TCP	22	172.31.1.2/32	Allows inbound SSH access from the remote computer. If the instance is a Windows computer, then this rule must use the RDP protocol for port 3389 instead.
Outbound Rules				
Protocol Type	Protocol	Port Range	Destination	Comments
All traffic	All	All	sg-1a2b3c4d	Enables instances associated with the same security group to communicate with each other. Security groups are stateful, therefore you don't need a rule that allows response traffic for inbound requests.

The subnet is associated with a network ACL that has the following rules.

Inbound Rules						
Rule #	Type	Protocol	Port Range	Source	Allow/Deny	Comments
100	SSH	TCP	22	172.31.1.2/32	ALLOW	Allows inbound traffic

						from the remote computer. If the instance is a Windows computer, then this rule must use the RDP protocol for port 3389 instead.
*	All traffic	All	All	0.0.0.0/0	DENY	Denies all other inbound traffic that does not match the previous rule.
Outbound Rules						
Rule #	Type	Protocol	Port Range	Destination	Allow/Deny	Comments
100	Custom TCP	TCP	1024-65535	172.31.1.2/32	ALLOW	Allows outbound responses to the remote computer. Network ACLs are stateless, therefore this rule is required to allow response traffic for inbound requests.
*	All traffic	All	All	0.0.0.0/0	DENY	Denies all other outbound traffic that does not match the previous rule.

This scenario gives you the flexibility to change the security groups or security group rules for your instances, and have the network ACL as the backup layer of defense. The network ACL rules apply to all instances in the subnet, so if you accidentally make your security group rules too permissive, the network ACL rules continue to permit access only from the single IP address. For example, the following rules are more permissive than the earlier rules — they allow inbound SSH access from any IP address.

Inbound Rules				
Type	Protocol	Port Range	Source	Comments
All traffic	All	All	sg-1a2b3c4d	Enables instances associated with the same security group to communicate with each other.
SSH	TCP	22	0.0.0.0/0	Allows SSH access from any IP address.
Outbound Rules				
Type	Protocol	Port Range	Destination	Comments
All traffic	All	All	0.0.0.0/0	Allows all outbound traffic.

However, only other instances within the subnet and your remote computer are able to access this instance. The network ACL rules still prevent all inbound traffic to the subnet except from your remote computer.

API and Command Overview

You can perform the tasks described on this page using the command line or an API. For more information about the command line interfaces and a list of available APIs, see [Accessing Amazon VPC \(what-is-amazon-vpc.html#VPCInterfaces\)](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html#VPCInterfaces).

Create a network ACL for your VPC

- [create-network-acl](https://docs.aws.amazon.com/cli/latest/reference/ec2/create-network-acl.html) (<https://docs.aws.amazon.com/cli/latest/reference/ec2/create-network-acl.html>) (AWS CLI)
- [New-EC2NetworkAcl](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2NetworkAcl.html) (<https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2NetworkAcl.html>) (AWS Tools for Windows PowerShell)

Describe one or more of your network ACLs

- [describe-network-acls](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-network-acls.html) (<https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-network-acls.html>) (AWS CLI)
- [Get-EC2NetworkAcl](https://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2NetworkAcl.html) (<https://docs.aws.amazon.com/powershell/latest/reference/items/Get-EC2NetworkAcl.html>) (AWS Tools for Windows PowerShell)

Add a rule to a network ACL

- [create-network-acl-entry](https://docs.aws.amazon.com/cli/latest/reference/ec2/create-network-acl-entry.html) (<https://docs.aws.amazon.com/cli/latest/reference/ec2/create-network-acl-entry.html>) (AWS CLI)
- [New-EC2NetworkAclEntry](https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2NetworkAclEntry.html) (<https://docs.aws.amazon.com/powershell/latest/reference/items/New-EC2NetworkAclEntry.html>) (AWS Tools for Windows PowerShell)

Delete a rule from a network ACL

- [delete-network-acl-entry](https://docs.aws.amazon.com/cli/latest/reference/ec2/delete-network-acl-entry.html) (<https://docs.aws.amazon.com/cli/latest/reference/ec2/delete-network-acl-entry.html>) (AWS CLI)
- [Remove-EC2NetworkAclEntry](https://docs.aws.amazon.com/powershell/latest/reference/items/Remove-EC2NetworkAclEntry.html) (<https://docs.aws.amazon.com/powershell/latest/reference/items/Remove-EC2NetworkAclEntry.html>) (AWS Tools for Windows PowerShell)

Replace an existing rule in a network ACL

- [replace-network-acl-entry](https://docs.aws.amazon.com/cli/latest/reference/ec2/replace-network-acl-entry.html) (<https://docs.aws.amazon.com/cli/latest/reference/ec2/replace-network-acl-entry.html>) (AWS CLI)

- [Set-EC2NetworkAclEntry](https://docs.aws.amazon.com/powershell/latest/reference/items/Set-EC2NetworkAclEntry.html)
(<https://docs.aws.amazon.com/powershell/latest/reference/items/Set-EC2NetworkAclEntry.html>) (AWS Tools for Windows PowerShell)

Replace a network ACL association

- [replace-network-acl-association](https://docs.aws.amazon.com/cli/latest/reference/ec2/replace-network-acl-association.html)
(<https://docs.aws.amazon.com/cli/latest/reference/ec2/replace-network-acl-association.html>) (AWS CLI)
- [Set-EC2NetworkAclAssociation](https://docs.aws.amazon.com/powershell/latest/reference/items/Set-EC2NetworkAclAssociation.html)
(<https://docs.aws.amazon.com/powershell/latest/reference/items/Set-EC2NetworkAclAssociation.html>) (AWS Tools for Windows PowerShell)


Delete a network ACL

- [delete-network-acl](https://docs.aws.amazon.com/cli/latest/reference/ec2/delete-network-acl.html) (<https://docs.aws.amazon.com/cli/latest/reference/ec2/delete-network-acl.html>) (AWS CLI)
- [Remove-EC2NetworkAcl](https://docs.aws.amazon.com/powershell/latest/reference/items/Remove-EC2NetworkAcl.html)
(<https://docs.aws.amazon.com/powershell/latest/reference/items/Remove-EC2NetworkAcl.html>) (AWS Tools for Windows PowerShell)

RECOMMENDED FOR YOU



Deep dive on Bring Your Own IP

Learn how to easily migrate legacy applications that use IP addresses with Bring Your Own IP. [Register now](https://pages.awscloud.com/Deep-Dive-on-Bring-Your-Own-IP_1024-NET_OD.html?sc_ichannel=ha&sc_icontent=console_aws-docs_vpc_docsvpcbyoip_awssm-1111&sc_icampaign=Adoption_Campaign_CSI_10_2019_webevent_BYOIP&trkCampaign=CSI_10_2019_xxx&sc_ioutcome=CSI_Digital_Marketing&sc_iplace=console_aws-docs_vpc_STANDARD)  (https://pages.awscloud.com/Deep-Dive-on-Bring-Your-Own-IP_1024-NET_OD.html?sc_ichannel=ha&sc_icontent=console_aws-docs_vpc_docsvpcbyoip_awssm-1111&sc_icampaign=Adoption_Campaign_CSI_10_2019_webevent_BYOIP&trkCampaign=CSI_10_2019_xxx&sc_ioutcome=CSI_Digital_Marketing&sc_iplace=console_aws-docs_vpc_STANDARD)

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.