I host a website on an EC2 instance. How do I allow my users to connect on HTTP (80) or HTTPS (443)?

Issue

I host my website on an EC2 instance, and I want users to connect to my website on HTTP (port 80) or HTTPS (port 443). How can I do that?

Resolution

To allow traffic on port 80 and 443, you must configure the associated security group and network access control list (network ACL).

Security group rules

For HTTP traffic, add an inbound rule on port 80 from the source address 0.0.0.0/0. For HTTPS traffic, add an inbound rule on port 443 from the source address 0.0.0.0/0. These inbound rules allow traffic from IPv4 addresses. To allow IPv6 traffic, add inbound rules on the same ports from the source address ::/0. For more information on creating or modifying security groups, see [Working with Security Groups](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#WorkingWithSecurityGroups).

Because security groups are stateful, the return traffic from the instance to users is allowed automatically, so you don't need to modify the security group's outbound rules.

The following example shows the security group rules for allowing both IPv4 and IPv6 traffic on port 80 and 443:

Inbound rules

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Protocol | Port Range | Source |
| HTTP (80) | TCP (6) | 80 | 0.0.0.0/0 |
| HTTP (80) | TCP (6) | 80 | ::/0 |
| HTTPS (443) | TCP (6) | 443 | 0.0.0.0/0 |
| HTTPS (443) | TCP (6) | 443 | ::/0 |

Network ACL

The default network ACL allows all inbound and outbound traffic. If you use a custom network ACL with more restrictive rules, then explicitly allow traffic on port 80 and 443. Network ACLs are stateless, so add both inbound and outbound rules to enable the connection to your website. For more information on modifying network ACL rules, see [Network ACLs](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html).

Note: If your users connect over IPv6 and your Amazon Virtual Private Cloud (Amazon VPC) has an associated IPv6 CIDR block, then your default network ACL automatically adds rules allowing all inbound and outbound IPv6 traffic.

The following example shows a custom network ACL that allows traffic on port 80 and 443:

Inbound rules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rule # | Type | Protocol | Port Range | Source | Allow/Deny |
| 100 | HTTP (80) | TCP (6) | 80 | 0.0.0.0/0 | ALLOW |
| 101 | HTTPS (443) | TCP (6) | 443 | 0.0.0.0/0 | ALLOW |
| 102 | HTTP (80) | TCP (6) | 80 | ::/0 | ALLOW |
| 103 | HTTPS (443) | TCP (6) | 443 | ::/0 | ALLOW |
| \* | ALL Traffic | ALL | ALL | ::/0 | DENY |
| \* | ALL Traffic | ALL | ALL | 0.0.0.0/0 | DENY |

Outbound rules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rule # | Type | Protocol | Port Range | Destination | Allow/Deny |
| 100 | Custom TCP Rule | TCP (6) | 1024-65535 | 0.0.0.0/0 | ALLOW |
| 101 | Custom TCP Rule | TCP (6) | 1024-65535 | ::/0 | ALLOW |
| \* | ALL Traffic | ALL | ALL | ::/0 | DENY |
| \* | ALL Traffic | ALL | ALL | 0.0.0.0/0 | DENY |

Note: When the previous security group and network ACL example configurations are used together, all internet users can connect to the website. If the website owner or administrator wants to access other websites from the EC2 instance, then the following configurations must be allowed:

* Network ACL outbound rules allowing traffic on port 80 or port 443 to the destination IP address
* Network ACL inbound rules allowing traffic on ephemeral ports (1024-65535)
* Security group rules allowing outbound traffic

# Elastic Network Interfaces

An elastic network interface (referred to as a network interface in this documentation) is a logical networking component in a VPC that represents a virtual network card.

A network interface can include the following attributes:

* A primary private IPv4 address from the IPv4 address range of your VPC
* One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
* One Elastic IP address (IPv4) per private IPv4 address
* One public IPv4 address
* One or more IPv6 addresses
* One or more security groups
* A MAC address
* A source/destination check flag
* A description…

I’ve reached the limit for elastic network interfaces in an AWS region

Issue

I've reached the limit for elastic network interfaces in an AWS region. How do I get more elastic network interfaces?

Resolution

Your limit for elastic network interfaces increases automatically as new instances are launched on the account. If you need more elastic network interfaces than are automatically provisioned to you, do one or both of the following:

Delete unused elastic network interfaces

To free up more elastic network interfaces, start by [deleting any unused elastic network interfaces](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#delete_eni) in that AWS region:

1. Sign in to your account and open the [Network Interfaces pane of the EC2 console](https://console.aws.amazon.com/ec2/v2/home?#NIC).
2. (Optional) [Detach any elastic network interfaces](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html#detach_eni) that you don't need from their resources.
3. Select the elastic network interfaces that you want to delete.
4. Choose Delete.

Request a service limit for elastic network interfaces in a specific AWS region

1. Sign in to your account and open the [AWS Support console](https://console.aws.amazon.com/support/v1#/case/create). For Regarding, choose Service Limit Increase. For Limit type, choose VPC. For Region, choose the AWS region you need the elastic network interfaces in. For Limit, choose Network Interfaces per Region. For New limit value, type the number of elastic network interfaces you need.
2. Complete the remaining fields.
3. (Optional) If you need more elastic network interfaces in more than one AWS region, choose Add another request and repeat the process with another AWS region.
4. Choose Submit.

# Why can't I connect to a service when the security group and network ACL allow inbound traffic?

### Issue

I enabled inbound traffic to the necessary ports in both the security group and the network access control list (network ACL), but I still can't connect to a service running on an EC2 instance. How do I fix this?

### Short Description

Security groups are stateful, so allowing inbound traffic to the necessary ports enables the connection. Network ACLs are stateless, so you must allow both inbound and outbound traffic.

### Resolution

To enable the connection to a service running on an instance, the associated network ACL must allow both inbound traffic on the port that the service is listening on as well as allow outbound traffic from ephemeral ports. When a client connects to a server, a random port from the ephemeral port range (1024-65535) becomes the client's source port.

The designated ephemeral port then becomes the destination port for return traffic from the service, so outbound traffic from the ephemeral port must be allowed in the network ACL. For more information on modifying network ACL rules, see [Adding and Deleting Rules](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#Rules).

By default, network ACLs allow all inbound and outbound traffic. If your network ACL is more restrictive, then you need to explicitly allow traffic from the ephemeral port range.

Note: If you accept traffic from the internet, then you also must establish a route through an [internet gateway](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html). If you accept traffic over VPN or AWS Direct Connect, then you must establish a route through a [virtual private gateway](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/SetUpVPNConnections.html#vpn-create-vpg).

How do I resolve intermittent connection issues when using a NAT instance?

Issue

I'm using a NAT instance so that instances in a private VPC subnet can connect to the internet, but the instances have intermittent connection issues. How can I fix this?

Short Description

The connection problems might be related to these issues:

* Instance operating system-level connection limits
* Network access control list (network ACL) rules
* Network issues

Resolution

Instance operating system-level connection limits

Check if the NAT instance and the instances in the private subnet have reached their operating system-level connection limits. To get the number of active connections, run the netstat command:

Linux

netstat -ano | grep ESTABLISHED | wc –l

netstat -ano | grep TIME\_WAIT | wc –l

Windows

netstat -ano | find /i "estab" /c

netstat -ano | find /i "TIME\_WAIT" /c

If the command returns a value near the allowed local port range (source port for client connections), then you might be running into port exhaustion. To reduce port exhaustion, try one of these solutions:

* Increase the operating system local (ephemeral) port range by running this command:

net.ipv4.ip\_local\_port\_range = 1025 61000

* Add ephemeral ports for new connections by [allocating more elastic IPs](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-addressing-eips-allocating) to the NAT instance, or by increasing the number of NAT instances for internet-bound traffic.
* Resolve any application-level issues that drain the available connections.

Network ACL rules

Confirm that the network ACL allows inbound traffic from the ephemeral port range (1024-65535). If the network ACL allows only a subset of the ephemeral port range, and the instances in the private subnet use a source port outside of that range, then traffic is dropped.

Note: If you're using a NAT gateway instead of a NAT instance, use the CloudWatch ErrorPortAllocation metric to verify if source ports are exhausted. For more information on this metric, see [Amazon VPC NAT Gateway Metrics and Dimensions](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/nat-gateway-metricscollected.html).

Network issues

Connectivity issues can be related to network problems, such as packet loss or destination host issues. For troubleshooting steps, see [How do I troubleshoot network performance issues between Amazon EC2 Linux instances in a VPC and an on-premises host over the internet gateway?](https://aws.amazon.com/premiumsupport/knowledge-center/network-issue-vpc-onprem-ig/)

Note: Packet captures on the source, destination, and NAT instance can provide more information on network issues.

### How do I troubleshoot network performance issues between Amazon EC2 Linux instances in a VPC and an on-premises host over the internet gateway?(VIDEO) Issue

I am experiencing packet loss or latency issues between my Amazon EC2 Linux instances and my on-premises host over the internet gateway. How can I troubleshoot these issues with network performance?

Short Description

To diagnose network issues such as packet loss or latency, first test the network to isolate the source of the issue. This following steps can help determine if the source of the issue is a network or an application.

Before you begin troubleshooting, check the following:

* Ensure that both endpoints (the EC2 instance and the on-premises host) have the network utilities installed.
* Use an EC2 instance that supports enhanced networking, and ensure that the drivers are up to date. Enhanced networking provides higher I/O with low CPU utilization, which helps avoids instance-level issues when executing performance tests. If Enhanced networking is not enabled, see [Enabling Enhanced Networking on Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html#enabling_enhanced_networking).
* [Connect to Your Linux Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html) to access the instances and ensure that there is end-to-end connectivity between your EC2 instance and your on-premises host.

Resolution

Install the following tools to help troubleshoot and test your network:

* MTR to check for ICMP or TCP packet loss and latency problems.
* Traceroute to determine latency or routing problems.
* Hping3 to determine end-to-end TCP packet loss and latency problems.
* Tcpdump to analyze packet capture samples.

Review hops on traceroute or MTR reports using a bottom-up approach. For example, check for loss on the last hop/destination and then review the following hops. If the packet loss or latency issues continue through the last hop, there might be a network or routing issue. If you observe packet loss/latency on only one hop in the path, there might be an issue with the control plane rate limiting on that node. Check if the last hop reported is the destination noted in the command. If it isn't, there might be an issue caused by a restrictive security group.

Test performance using MTR

The Linux MTR command provides continual, updated output, allowing you to analyze network performance. It's a diagnostic tool that combines the functionality of traceroute and ping utilities. Most Linux distributions come with traceroute and MTR already installed. However, you can also download it from your distribution’s software package manager. To install MTR, run the following commands:

Amazon Linux

sudo yum install mtr

Ubuntu

sudo apt-get install mtr

To test your network's performance using MTR, run this test bidirectionally between the public IP address of your EC2 instances and your on-premises host. The path between nodes on a TCP/IP network can change if the direction is reversed. Therefore, it's important to obtain MTR results for both directions. You can use a TCP-based trace instead of ICMP because most internet devices deprioritize ICMP-based trace requests. Lastly, review your packet loss. Packet loss on a single hop usually doesn't indicate an issue. It can be the result of a control plane policy causing "ICMP time exceeded" messages to be dropped. If you notice sustained packet loss until the destination hop, or packet loss over several hops, this might indicate a problem.

Note: A few timed-out requests are common.

ICMP-based MTR

mtr -n -c 200 <Public IP EC2 instance/on-premises host> --report

TCP-based MTR

mtr -n -T -c 200 <Public IP EC2 instance/on-premises host> --report

The argument -T performs a TCP-based MTR, and the --report option puts MTR into report mode. MTR runs for the number of cycles specified by the -c option. Print the statistics, and then exit.

Note: There is a known issue with some versions of MTR where the final hop reports an incorrect value if TCP is used.

Test performance using traceroute

The Linux traceroute utility identifies the path taken from a client node to the destination node, and it records the time in milliseconds for each router to respond to the request. It also calculates the amount of time each hop takes before reaching its destination. To install traceroute, run the following commands:

Amazon Linux

sudo yum install traceroute

Ubuntu

sudo apt-get install traceroute

Note: Traceroute is not necessary if an MTR report has been executed. MTR provides latency and packet loss statistics to a destination.

Ensure that you have port 22 or the port that you are testing open in both directions. If you are troubleshooting network connectivity using traceroute, run the command in both directions—from the client to the server and then from the server back to the client. The path between nodes on a TCP/IP network can change if the direction is reversed. Use a TCP-based trace instead of ICMP because most internet devices deprioritize ICMP-based trace requests.

ICMP-based traceroute

sudo traceroute <Public IP of EC2 instance/on-premises host>

TCP-based traceroute

sudo traceroute -T -p 22 –n <Public IP of EC2 instance/on-premises host>

The argument -T -p 22 -n performs a TCP-based trace on port 22.

Test performance using hping3

Hping3 is a command-line oriented TCP/IP packet assembler and analyzer that measures end-to-end packet loss and latency over a TCP connection. In addition to ICMP echo requests, hping3 supports TCP, UDP, and RAW-IP protocols, and it includes a traceroute mode that can send files between a covered channel. Hping3 is designed to scan hosts, assist with penetration testing, test intrusion detection systems, and send files between hosts.

MTRs and traceroute capture per-hop latency. However, hping3 yields results that show end-to-end min/avg/max latency over TCP in addition to packet loss. To install hping3, run the following commands:

Amazon Linux

sudo yum --enablerepo=epel install hping3

Ubuntu

sudo apt-get install hping3

The following command sends 50 TCP SYN packets over port 0. By default, hping3 sends TCP headers to the target host's port 0 with a window size of 64 without any TCP flag:

hping3 -S -c 50 -V <Public IP of EC2 instance/on-premises host>

The following command sends 50 TCP SYN packets over port 22:

hping3 -S -c 50 -V <Public IP of EC2 instance/on-premises host> -p 22

Note: Ensure that you have port 22 or the port that you are testing open.

Test packet capture samples using tcpdump

It's best to perform simultaneous packet captures on your EC2 instance and on-premises host when diagnosing packet loss/latency issues because it can help isolate the issue at the networking and application layers. To install tcpdump, run the following commands:

Amazon Linux

sudo yum install tcpdump

Ubuntu

sudo apt-get install tcpdump

If you find evidence of packet loss in your network, refer to your vendor documentation for instructions on how to check network devices for analysis and troubleshooting. If you are multi-homed, perform these tests using a different Internet Service Provider (ISP), and then compare the results.

# I tried to delete my Amazon VPC, and I received a dependency error. How can I delete my Amazon VPC?

Issue

I received a dependency error when I tried to delete my Amazon Virtual Private Cloud (Amazon VPC). How do I delete my Amazon VPC?

Short Description

Before you can delete your Amazon VPC, you must delete its dependent resources. Otherwise, you might receive a dependency error.

If you [delete your Amazon VPC using the Amazon VPC console](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/working-with-vpcs.html#VPC_Deleting), all its components--such as subnets, security groups, network ACLs, route tables, internet gateways, VPC peering connections, and DHCP options--are also deleted. If you use the [AWS Command Line Interface (AWS CLI) to delete the Amazon VPC](https://docs.aws.amazon.com/cli/latest/reference/ec2/delete-vpc.html), you must terminate all instances, delete all subnets, delete custom security groups and custom route tables, and detach any internet gateway in the Amazon VPC before you can delete the Amazon VPC.

Resolution

To delete an Amazon VPC, first [terminate your instances](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/terminating-instances.html) in the Amazon VPC. If the Amazon VPC has running Amazon EC2 instances, you will receive the following error: "VPC contains one or more instances, and cannot be deleted until those instances have been terminated."

If the Amazon VPC has network interfaces for other AWS services, such as Amazon Relational Database Service (Amazon RDS), Amazon WorkSpaces, or Amazon ElastiCache, you will receive one of the following errors:

* Network vpc-xxxxxxxx has some mapped public address(es). Please unmap those public address(es) before detaching the gateway.
* Network interface 'eni-xxxxxxxx' is currently in use.

To resolve these errors, determine which service is dependent by following these steps:

1. Open the [Amazon EC2 console](https://console.aws.amazon.com/ec2).
2. Select the region that the Amazon VPC is in.
3. In the navigation pane, under Network Interfaces, search for the VPC ID that you want to delete.
4. Choose the elastic network interface, choose the Details view, and check the Description to see which resource the interface is attached to.
5. Delete the associated resource.  
   Note: If it is a Primary network interface, it will be deleted when you will delete the instance.
6. [Delete the Amazon VPC](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/working-with-vpcs.html#VPC_Deleting).

You must also delete any security group references from the default security group. If the default security group of the Amazon VPC references other security groups from the Amazon VPC, you will receive the following error: "Resource sg-xxxxxxxx has a dependent object". For more information about deleting security groups, see [Deleting a Security Group](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#DeleteSecurityGroup).

Why can't I delete a security group for my Amazon VPC?

Issue

I'm getting errors when I try to delete a security group for my Amazon Virtual Private Cloud (Amazon VPC). What can I do to delete it?

Short Description

You might not be able to delete the security group because it's:

* A default security group
* Associated with an instance that is in the running or stopped state
* Referenced by its own rule or a rule in another security group

You can delete the security group only when it does not meet any of these scenarios. For steps on how to delete a security group, see [Deleting a Security Group](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#DeleteSecurityGroup).

Note: To delete two or more security groups at the same time, use the [AWS Management Console](https://console.aws.amazon.com/). You can delete only one security group at a time when you use the AWS Command Line Interface (AWS CLI) or API.

Resolution

Default security group

Default security groups can't be deleted. To verify if a security group is a default resource, follow these steps:

1. Open the [Amazon VPC console](https://console.aws.amazon.com/vpc/).
2. In the navigation pane, under Security, choose Security Groups.
3. In the security group list, the default security groups are those with the Group Name shows as default.

Associated with an instance in the running or stopped state

You can't delete a security group if it's assigned to a running or stopped instance. To verify if the security group is assigned to an instance, follow these steps:

1. Open the [Amazon EC2 console](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Instances.
3. Choose the search bar in the content pane.
4. In the Resource Attributes filter list, select either Security Group ID or Security Group Name, and then select the security group ID or security group name that you want to delete.
5. Any instances assigned to the security group will appear in the filtered instance list.

Note: To change the security group assigned to an instance, see [Changing an Instance's Security Groups](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#SG_Changing_Group_Membership).

Referenced by its own rule or a rule in another security group

You can't delete a security group if it's referenced by a security group rule. If the security group is referenced in one of its own rules, then you must remove the rule before deleting the security group. If the security group is referenced in another security group's rules, you must remove the reference to delete the security group. To modify security group rules, see [Adding, Removing, and Updating Rules](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#AddRemoveRules).

The security group might also be referenced in a security group within another Amazon VPC where a [peering connection](https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/Welcome.html) is established. To delete the security group, you can either remove the reference or [delete the VPC peering connection](https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/delete-vpc-peering-connection.html).

# Why aren't the configuration parameters of my DHCP options set passing to instances in the VPC?

Issue

I set custom configuration parameters to the [Dynamic Host Configuration Protocol (DHCP)](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#DHCPOptionSets) options set for my VPC, but those options aren't passing to the EC2 instances in the VPC. How can I fix this?

Short Description

Check these resources and configurations to diagnose and troubleshoot the issue:

* Parameters of the DHCP options set
* Network configuration and operating system kernel parameters of the EC2 instances
* Settings of Simple Active Directory (Simple AD) and private hosted zones

Resolution

Parameters of the DHCP options set

1. Open the [Amazon VPC console](https://console.aws.amazon.com/vpc/).
2. In the navigation pane, under Virtual Private Cloud, choose DHCP Options Sets.
3. In the resource list, choose the DHCP options set with your custom configuration parameters.
4. In the Summary view, verify that the Options follow the guidelines described in [Overview of DHCP Options Sets](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#DHCPOptionSets).  
   Note: After a DHCP options set is created, it can't be modified. To modify your DHCP options set, [create a new DHCP options set](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#CreatingaDHCPOptionSet)with the correct parameters and [associate it with your VPC](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#ChangingDHCPOptionsofaVPC).

Network configuration and operating system kernel parameters of the EC2 instances

* Search for any customization (PEERDNS, timeouts, etc.) in the network configuration files that were set either manually or using scripts. For more information, see [User Data and Shell Scripts](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html#user-data-shell-scripts).
* Verify that the configuration files used by the operating system are mutable. If the files are immutable, then the instance won't receive the configuration parameters from the DHCP options set correctly. When using Linux, configuration files are typically made immutable with the chattr command.
* Check the operating systems of the EC2 instances and search for known bugs. If there's a bug related to the issue, follow the guidelines provided by the operating system.

Settings of Simple AD and private hosted zones

* You can resolve DNS requests to your Amazon Route 53 private hosted zone by pointing your on-premises computers to Simple AD, which forwards DNS requests to your VPC's Amazon-provided DNS servers. For more information, see [Using DNS with Simple AD and Microsoft AD](https://docs.aws.amazon.com/directoryservice/latest/admin-guide/dns_with_simple_ad.html).
* If you're using a custom hostname, confirm that the DNS server in the DHCP options set has proper zone definitions and DNS records for your custom domain. Additionally, be sure that you have manually configured the custom hostname on the EC2 instance. For more information, see [Changing the Hostname of Your Linux Instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/set-hostname.html) and [Working with Private Hosted Zones](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted-zones-private.html).

Note: To configure your VPC for Simple AD, [you can customize your DHCP options set to use your directory's DNS servers](http://docs.aws.amazon.com/directoryservice/latest/admin-guide/dhcp_options_set.html). Or, you can use the default DHCP options set, but you must then [manually join the instances to the Simple AD directory](http://docs.aws.amazon.com/directoryservice/latest/admin-guide/join_windows_instance.html).