# AMAZON VPC WITH PUBLIC AND PRIVATE SUBNET

Table of Contents

[AMAZON VPC WITH PUBLIC AND PRIVATE SUBNET 1](#_Toc405458527)

[NAT Instances 2](#_Toc405458528)

[Create a VPC with two subnets. 2](#_Toc405458529)

[Creating a Custom Route Table 3](#_Toc405458530)

[To create a custom route table 4](#_Toc405458531)

[Create the NATSG security group You'll specify this security group when you launch the NAT instance. 4](#_Toc405458532)

[Creating the NATSG Security Group 4](#_Toc405458533)

[To create the NATSG security group 5](#_Toc405458534)

[ACL Rules for the Public Subnet[OPTIONAL in case you have sec group] 6](#_Toc405458535)

[Disable the SrcDestCheck attribute for the NAT instance 8](#_Toc405458536)

[Disabling Source/Destination Checks 8](#_Toc405458537)

[Testing Your NAT Instance Configuration 8](#_Toc405458538)

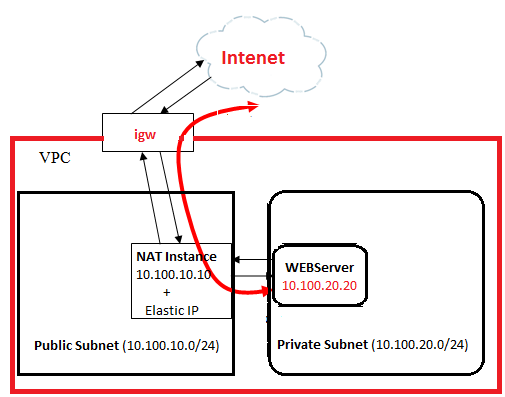
[To launch an instance into your private subnet 9](#_Toc405458539)

[Amazon VPC Limits 11](#_Toc405458540)

AnilKumar AN

# NAT Instances

Instances that you launch into a private subnet in a virtual private cloud (VPC) can't communicate with the Internet. You can optionally use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound traffic to the Internet, but prevent the instances from receiving inbound traffic initiated by someone on the Internet.



Note : Change the subnet/IP address as per your requirement.

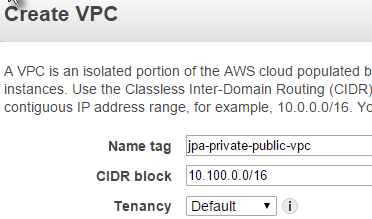
igw- internet gateway

In the above scenario, we’ll create a micro instance inside the public subnet with an IP 10.100.10.0/24, which will act as the gateway for all the instance(s) inside the private subnet (10.100.20.0/24).

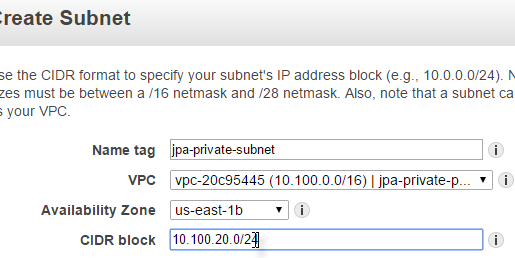
You can launch a pre configured NAT instance using the AMI (ami-4c9e4b24)

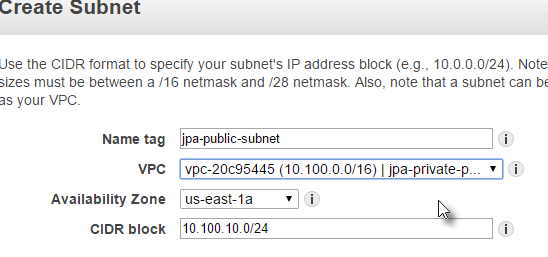
## Create a VPC with two subnets.

* 1. Create a VPC

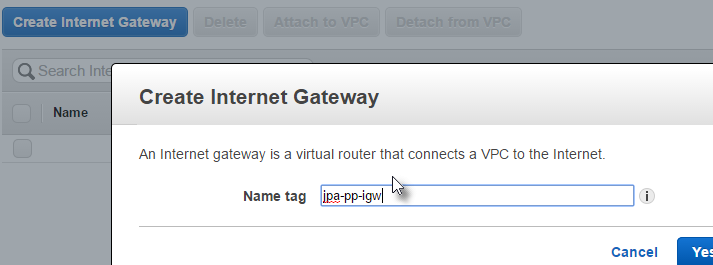


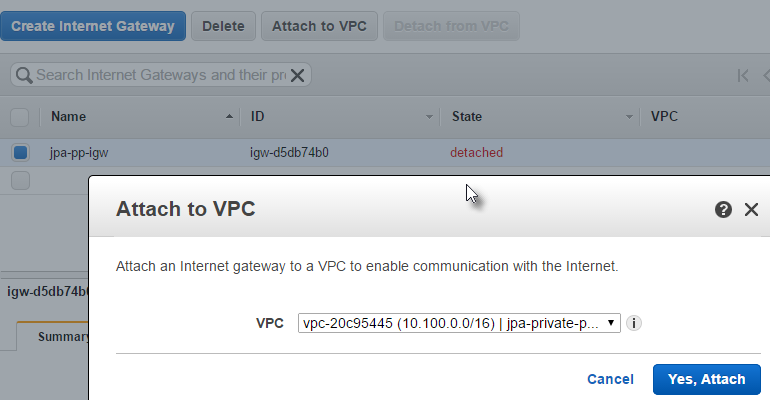
* 1. Create two subnets





* 1. Attach an Internet gateway to the VPC





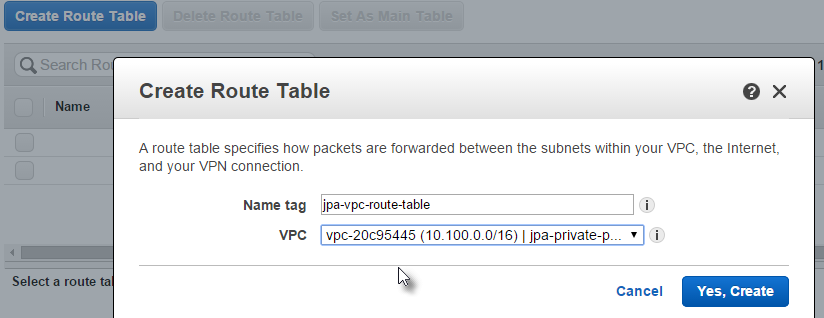
* 1. Create a custom route table that sends traffic destined outside the VPC to the Internet gateway, and then associate it with one subnet, making it a public

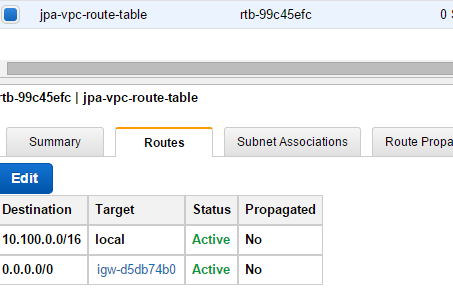
## Creating a Custom Route Table FoR PUBLIC and PRIVATE SUBNETs

When you create a subnet, we automatically associate it with the main route table for the VPC. By default, the main route table doesn't contain a route to an Internet gateway. The following procedure creates a custom route table with a route that sends traffic destined outside the VPC to the Internet gateway, and then associates it with your subnet.

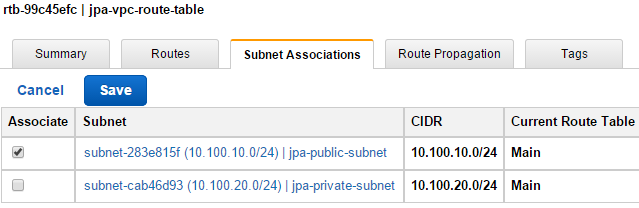
## To create a custom route table

1. Open the Amazon VPC console .
2. In the navigation pane, click **Route Tables**, and then click **Create Route Table**.
3. In the **Create Route Table** dialog box, optionally name your route table, then select your VPC, and then click **Yes, Create**.
4. Select the custom route table that you just created. The details pane displays tabs for working with its routes, associations, and route propagation.
5. On the **Routes** tab, click **Edit**, specify 0.0.0.0/0 in the **Destination** box, select the Internet gateway ID in the **Target** list, and then click**Save**.
6. On the **Subnet Associations** tab, click **Edit**, select the **Associate** check box for the subnet, and then click **Save**





Select Public subnet’s checkbox alone and save



## Create the NATSG security group You'll specify this security group when you launch the NAT instance.

## Creating the NATSG Security Group

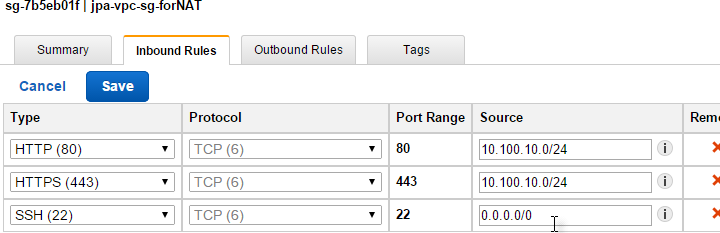
Define the NATSG security group as described in the following table to enable your NAT instance to receive Internet-bound traffic from instances in a private subnet, as well as SSH traffic from your network. The NAT instance can also send traffic to the Internet, which enables the instances in the private subnet to get software updates.

**NATSG: Recommended Rules**

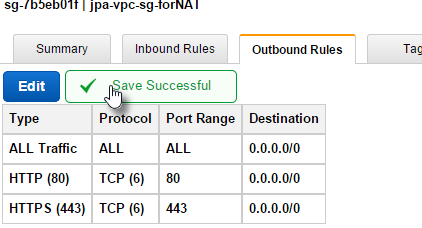
|  |  |  |  |
| --- | --- | --- | --- |
| **Inbound** | | | |
| **Source** | **Protocol** | **Port Range** | **Comments** |
| 10.100.20.0/24 | TCP | 80 | Allow inbound HTTP traffic from servers in the private subnet |
| 10.100.20.0/24 | TCP | 443 | Allow inbound HTTPS traffic from servers in the private subnet |
| Public IP address range of your home network | TCP | 22 | Allow inbound SSH access to the NAT instance from your home network (over the Internet gateway) |
| **Outbound** | | | |
| **Destination** | **Protocol** | **Port Range** | **Comments** |
| 0.0.0.0/0 | TCP | 80 | Allow outbound HTTP access to the Internet |
| 0.0.0.0/0 | TCP | 443 | Allow outbound HTTPS access to the Internet |

## To create the NATSG security group

1. Open the Amazon VPC console
2. Click **Security Groups** in the navigation pane.
3. Click the **Create Security Group** button.
4. In the **Create Security Group** dialog box, specify jpa-vpc-sg-forNAT  as the name of the security group, and provide a description. Select the ID of your VPC from the **VPC** list, and then click **Yes, Create**.
5. Select the NATSG security group that you just created. The details pane displays the details for the security group, plus tabs for working with its inbound and outbound rules.
6. Add rules for inbound traffic using the **Inbound Rules** tab as follows:
   1. Click **Edit**.
   2. Click **Add another rule**, and select **HTTP** from the **Type** list. In the **Source** field, specify the IP address range of your private subnet.
   3. Click **Add another rule**, and select **HTTPS** from the **Type** list. In the **Source** field, specify the IP address range of your private subnet.
   4. Click **Add another rule**, and select **SSH** from the **Type** list. In the **Source** field, specify the public IP address range of your network.
   5. Click **Save**.



1. Add rules for outbound traffic using the **Outbound Rules** tab as follows:
   1. Click **Edit**.
   2. Click **Add another rule**, and select **HTTP** from the **Type** list. In the **Destination** field, specify 0.0.0.0/0
   3. Click **Add another rule**, and select **HTTPS** from the **Type** list. In the **Destination** field, specify 0.0.0.0/0
   4. Click **Save**.



IF YOU DO NOT WANT TO USE A SECURITY GROUP, WHICH IS EFFECTIVE AT INSTANCE LEVEL, You MAY CREATE a NETWOK ACL , WHICH IS EFFECTIVE AT SUBNET LEVEL

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ACL Rules for the Public Subnet[OPTIONAL in case you have sec group]  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Inbound** | | | | | | | **Rule #** | **Source IP** | **Protocol** | **Port** | **Allow/Deny** | **Comments** | | 100 | 0.0.0.0/0 | TCP | 80 | ALLOW | Allows inbound HTTP traffic from anywhere | | 110 | 0.0.0.0/0 | TCP | 443 | ALLOW | Allows inbound HTTPS traffic from anywhere | | 120 | Public IP address range of your home network | TCP | 22 | ALLOW | Allows inbound SSH traffic from your home network (over the Internet gateway) | | 130 | Public IP address range of your home network | TCP | 3389 | ALLOW | Allows inbound RDP traffic from your home network (over the Internet gateway) | | 140 | 0.0.0.0/0 | TCP | 49152-65535 | ALLOW | Allows inbound return traffic from requests originating in the subnet  See the important note at the beginning of this topic about specifying the correct ephemeral ports. | | \* | 0.0.0.0/0 | all | all | DENY | Denies all inbound traffic not already handled by a preceding rule (not modifiable) | | **Outbound** | | | | | | | **Rule #** | **Dest IP** | **Protocol** | **Port** | **Allow/Deny** | **Comments** | | 100 | 0.0.0.0/0 | TCP | 80 | ALLOW | Allows outbound HTTP traffic from the subnet to the Internet | | 110 | 0.0.0.0/0 | TCP | 443 | ALLOW | Allows outbound HTTPS traffic from the subnet to the Internet | | 120 | 10.100.20.0/24 | TCP | 1433 | ALLOW | Allows outbound MS SQL access to database servers in the private subnet | | 130 | 10.100.20/24 | TCP | 3306 | ALLOW | Allows outbound MYSQL access to database servers in the private subnet | | 140 | 0.0.0.0/0 | TCP | 49152-65535 | ALLOW | Allows outbound responses to clients on the Internet (for example, serving web pages to people visiting the web servers in the subnet)  See the important note at the beginning of this topic about specifying the correct ephemeral ports. | | \* | 0.0.0.0/0 | all | all | DENY | Denies all outbound traffic not already handled by a preceding rule (not modifiable) |   **ACL Rules for the Private Subnet**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Inbound** | | | | | | | **Rule #** | **Source IP** | **Protocol** | **Port** | **Allow/Deny** | **Comments** | | 100 | 10.100.0.0/24 | TCP | 1433 | ALLOW | Allows web servers in the public subnet to read and write to MS SQL servers in the private subnet | | 110 | 10.100.0.0/24 | TCP | 3306 | ALLOW | Allows web servers in the public subnet to read and write to MySQL servers in the private subnet | | 120 | 10.100.0.0/24 | TCP | 22 | ALLOW | Allows inbound SSH traffic from the SSH bastion in the public subnet | | 130 | 10.100.0.0/24 | TCP | 3389 | ALLOW | Allows inbound RDP traffic from the Microsoft Terminal Services gateway in the public subnet | | 140 | 0.0.0.0/0 | TCP | 49152-65535 | ALLOW | Allows inbound return traffic from NAT instance in the public subnet for requests originating in the private subnet  See the important note at the beginning of this topic about specifying the correct ephemeral ports. | | \* | 0.0.0.0/0 | all | all | DENY | Denies all inbound traffic not already handled by a preceding rule (not modifiable) | | **Outbound** | | | | | | | **Rule #** | **Dest IP** | **Protocol** | **Port** | **Allow/Deny** | **Comments** | | 100 | 0.0.0.0/0 | TCP | 80 | ALLOW | Allows outbound HTTP traffic from the subnet to the Internet | | 110 | 0.0.0.0/0 | TCP | 443 | ALLOW | Allows outbound HTTPS traffic from the subnet to the Internet | | 120 | 10.0.0.0/24 | TCP | 49152-65535 | ALLOW | Allows outbound responses to the public subnet (for example, responses to web servers in the public subnet that are communicating with DB Servers in the private subnet)  See the important note at the beginning of this topic about specifying the correct ephemeral ports. | | \* | 0.0.0.0/0 | all | all | DENY | Denies all outbound traffic not already handled by a preceding rule (not modifiable) | |

1. Launch an instance into your public subnet from an AMI that's been configured to run as a NAT instance. Amazon provides Amazon Linux AMIs that are configured to run as NAT instances. These AMIs include the string amzn-ami-vpc-nat in their names, so you can search for them in the Amazon EC2 console.

Open the Amazon EC2 console.

On the dashboard, click the **Launch Instance** button, and complete the wizard as follows:

* 1. On the **Choose an Amazon Machine Image (AMI)** page, select the **Community AMIs** category, and search for amzn-ami-vpc-nat. In the results list, each AMI's name includes the version to enable you to select the most recent AMI, for example,2013.09. Click **Select**.

**e.g.amzn-ami-vpc-nat-hvm-2014.03.2.x86\_64-ebs** - ami-4c9e4b24

* 1. On the **Choose an Instance Type** page, select the instance type, then click **Next: Configure Instance Details**.
  2. On the **Configure Instance Details** page, select the VPC you created from the **Network** list, and select **your public subnet** from the **Subnet** list.
  3. (Optional) Select the **Public IP** check box to request that your NAT instance receives a public IP address. If you choose not to assign a public IP address now, you can allocate an Elastic IP address and assign it to your instance after it's launched. For more information about assigning a public IP at launch, Click **Next: Add Storage**.
  4. You can choose to add storage to your instance, and on the next page, you can add tags. Click **Next: Configure Security Group** when you are done.
  5. On the **Configure Security Group** page, select the **Select an existing security group** option, and select the jpa-vpc-sg-forNAT   security group that you created. Click **Review and Launch**.
  6. Review the setting and Launch
  7. Check if proper iptable rules in place with iptables -t nat -v -L -n

an implemented rule, effect similar to below iptables rule create line:

iptables -t nat -A POSTROUTING -o eth0 -s 10.100.20.0/24 -j MASQUERADE

* 1. Check in /etc/stsctl.conf

# Controls IP packet forwarding

net.ipv4.ip\_forward = 1

else change it as above and run sysctl –p to effect it.

## Disable the SrcDestCheck attribute for the NAT instance

## Disabling Source/Destination Checks

Each EC2 instance performs source/destination checks by default. This means that the instance must be the source or destination of any traffic it sends or receives. However, a NAT instance must be able to send and receive traffic when the source or destination is not itself. Therefore, you must disable source/destination checks on the NAT instance.

You can disable the SrcDestCheck attribute for a NAT instance that's either running or stopped using the console or the command line.

**To disable source/destination checking using the console**

1. Open the Amazon EC2 console.
2. Click **Instances** in the navigation pane.
3. Select the NAT instance, click **Actions**-**Networking** and then click **Change Source/Dest. Check**.
4. For a NAT instance, verify that this attribute is disabled. Otherwise, click **Yes, Disable**

## Testing Your NAT Instance Configuration

After you have launched a NAT instance and completed the configuration steps above, you can perform a test to check if an instance in your private subnet can access the Internet through the NAT instance. To do this, update your NAT instance's security group rules to accept inbound ICMP traffic, launch an instance into your private subnet, configure SSH agent forwarding to access instances in your private subnet, connect to your instance, and then test the Internet connectivity.

**To update your NAT instance's security group**

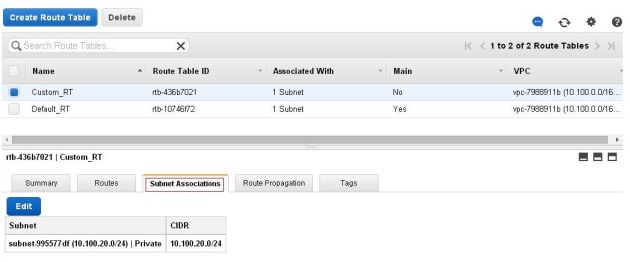
1. Open the Amazon EC2 console.
2. In the navigation pane, click **Security Groups**.
3. Find the security group associated with your NAT instance, and click **Edit** in the **Inbound** tab.
4. Click **Add Rule**, select **All ICMP** from the **Type** list, and select **Custom IP** from the **Source** list. Enter the IP address range of your private subnet, for example, 10.100.20.0/24. Click **Save**.

| **Type** | **Protocol** | **Port Range** | **Source** |
| --- | --- | --- | --- |
|  | | | | |
| SSH | TCP | 22 | 0.0.0.0/0 |  |
| HTTP | TCP | 80 | 10.100.10.0/24 |  |
| HTTPS | TCP | 443 | 10.100.10.0/24 |  |
| All ICMP | All | N/A | 10.100.20.0/24 |  |

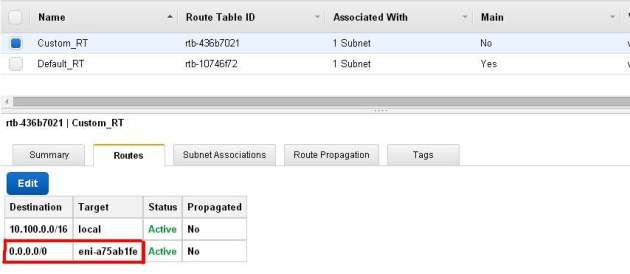
## To launch an instance into your private subnet

## ROUTE TABLE

1. Create a custom route, associate your private subnet(s) to it and make a default route to use the NAT instance as a gateway:

[](http://rbgeek.files.wordpress.com/2014/04/rt-1.jpg)

1. Create a route rule, Select the NAT INSTANCE' ID as Target, with 0.0.0.0/0 as Destination.

[](http://rbgeek.files.wordpress.com/2014/04/rt-2.jpg)

## LAUNCH AN INSTANCE

1. Open the Amazon EC2 console.
2. In the navigation pane, click **Instances**.
3. Launch an instance into your private subnet. For more information, see [Launching an Instance into Your Subnet](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#VPC_Launch_Instance). Ensure that you configure the following options in the launch wizard, and then click **Launch**:
   * On the **Choose an Amazon Machine Image (AMI)** page, select an Amazon Linux AMI from the **Quick Start** category.
   * On the **Configure Instance Details** page, select your private subnet from the **Subnet** list, **and do not assign a public IP address to your instance.**
   * On the **Configure Security Group** page, ensure that your security group includes a rule that allows SSH access from your NAT instance's private IP address, or from the IP address range of your public subnet. 10.100.10.0/24

|  |  |
| --- | --- |
| Inbound | Outbound |
|  |  |

* + In the **Select an existing key pair or create a new key pair** dialog box, select the same key pair you used to launch the NAT instance.

**To configure SSH agent forwarding for Linux**

From your local machine, add your private key to the authentication agent. (copy the contents of your PEM file to mykeypair.pem

Connect to your NAT instance using the -A option to enable SSH agent forwarding, for example:

ssh –i mykeypair.pem ec2-user@<PRIVATE-SUBNET –INSTANCE-IP>

[ec2-user@ip-10-100-10-137 ~]$ chmod 400 mykeypair.pem

[ec2-user@ip-10-100-10-137 ~]$ ssh -i mykeypair.pem ec2-user@10.100.20.35

Try to PING internet from the instance running in Private Subnet

[ec2-user@**ip-10-100-20-35** ~]$ ping www.google.com

PING www.google.com (74.125.131.99) 56(84) bytes of data.

64 bytes from vc-in-f99.1e100.net (74.125.131.99): icmp\_seq=1 ttl=46 time=14.7 ms

64 bytes from vc-in-f99.1e100.net (74.125.131.99): icmp\_seq=2 ttl=46 time=14.8 ms

FOR ADDITIONAL KNOWLEDGE:

# Amazon VPC Limits

The following table lists the limits related to Amazon VPC. Unless indicated otherwise, you can request an increase for any of these limits by using the support link : <https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-vpc>

| **Resource** | **Limit** | **Comments** |
| --- | --- | --- |
| VPCs per region | 5 | This limit can be increased upon request. The limit for Internet gateways per region is directly correlated to this one. Increasing this limit will increase the limit on Internet gateways per region by the same amount. |
| Subnets per VPC | 200 | This limit can be increased upon request. |
| Internet gateways per region | 5 | This limit is directly correlated with the limit on VPCs per region. You cannot increase this limit individually; the only way to increase this limit is to increase the limit on VPCs per region. Only one Internet gateway can be attached to a VPC at a time. |
| Virtual private gateways per region | 5 | Only one virtual private gateway can be attached to a VPC at a time. This limit can be increased upon request. |
| Customer gateways per region | 50 | This limit can be increased upon request. |
| VPN connections per region | 50 | This limit can be increased upon request. |
| VPN connections per VPC (per virtual private gateway) | 10 | This limit can be increased upon request. |
| Route tables per VPC | 200 | Including the main route table. You can associate one route table to one or more subnets in a VPC. |
| Entries per route table | 50 | This is the limit for the number of non-propagated entries per route table. This limit can be increased upon request; however, network performance may be impacted. |
| Elastic IP addresses per region for each AWS account | 5 | This is the limit for the number of VPC Elastic IPs you can allocate within a region. This is a separate limit from the EC2 Elastic IP address limit. This limit can be increased upon request. |
| Security groups per VPC | 100 | This limit can be increased upon request; however, network performance may be impacted, depending on the way the security groups are configured. |
| Rules per security group | 50 | This limit can be increased or decreased upon request, however, the multiple of the limit for rules per security group and the limit for security groups per network interface cannot exceed 250. For example, if you want 100 rules per security group, we decrease your number of security groups per network interface to 2. |
| Security groups per network interface | 5 | This limit can be increased or decreased upon request; up to a maximum of 16. The multiple of the limit for security groups per network interface and the limit for rules per security group cannot exceed 250. For example, if you want 10 security groups per network interface, we decrease your number of rules per security group to 25. |
| Network ACLs per VPC | 200 | You can associate one network ACL to one or more subnets in a VPC. This limit is not the same as the number of rules per network ACL. |
| Rules per network ACL | 20 | This is the one-way limit for a single network ACL, where the limit for ingress rules is 20, and the limit for egress rules is 20. |
| BGP Advertised Routes per VPN Connection | 100 | This limit cannot be increased. If you require more than 100 prefixes, advertise a default route. |
| Active VPC peering connections per VPC | 50 | This limit can be increased via special request to AWS Support. The maximum limit is 125 peering connections per VPC. The number of entries per route table should be increased accordingly; however, network performance may be impacted. |
| Outstanding VPC peering connection requests | 25 | This is the limit for the number of outstanding VPC peering connection requests that you've requested from your account. This limit can be increased via special request to AWS Support. |
| Expiry time for an unaccepted VPC peering connection request | 1 week (168 hours) | This limit can be increased via special request to AWS Support. |