Amazon Virtual Private Cloud

Amazon VPC lets us provision a logically isolated section of the Amazon Web Services (AWS) cloud where we can launch AWS resources in a virtual network that we define. We can have complete control over our virtual networking environment, including selection of our own IP address ranges, creation of subnets and configuration of route tables and network gateways. We can also create a hardware Virtual Private Network (VPN) connection between our corporate datacenter and our VPC and leverage the AWS cloud as an extension of our corporate datacenter.

We can customize the network configuration for our Amazon VPC. For e.g. we can create a public-facing subnet for our web server that have access to the Internet and place our backend systems such as databases in a private-facing subnet with no Internet access. We can leverage multiple layers of security, including security groups and network access control lists to help control access to Amazon EC2 instances in each subnet.

Components of Amazon VPC

Amazon VPC comprises a variety of objects that will be familiar to customers with existing networks:

- **A Virtual Private Cloud:** A logically isolated virtual network in the AWS cloud. We define a VPC’s IP address space from ranges we can select.
- **Subnet:** A segment of a VPC’s IP address range where we can place groups of isolated resources.
- **Internet Gateway:** The Amazon VPC side of a connection to the public Internet.
- **NAT Gateway:** A highly available, managed Network Address Translation (NAT) service for our resources in a private subnet to access the Internet.
- **Hardware VPN Connection:** A hardware-based VPN connection between your Amazon VPC and our datacenter, home network or co-location facility.
- **Virtual Private Gateway:** The Amazon VPC side of a VPN connection.
- **Customer Gateway:** Our side of a VPN connection.
- **Router:** Routers interconnect subnets and direct traffic between Internet gateways, virtual private gateways, NAT gateways, and subnets.
- **Peering Connection:** A peering connection enables us to route traffic via private IP addresses between two peered VPCs.
- **VPC Endpoints:** Enables private connectivity to services hosted in AWS, from within our VPC without using an Internet Gateway, VPN, Network Address Translation (NAT) devices or firewall proxies.

Amazon Virtual Private Cloud (Amazon VPC) enables us to launch AWS resources into a virtual network that we have defined. This virtual network closely resembles a traditional network that we would operate in our own data center, with the benefits of using the scalable infrastructure of AWS.

Amazon VPC Concepts

As we get started with Amazon VPC, we should understand the key concepts of this virtual network and how it is similar to or different from our own networks. Amazon VPC is the networking layer for Amazon EC2. The key concepts for Amazon VPC are:

VPCs and Subnets

A *virtual private cloud* (VPC) is a virtual network dedicated to our AWS account. It is logically isolated from other virtual networks in the AWS Cloud. We can launch your AWS resources, such as Amazon EC2 instances, into our VPC. We can configure our VPC by modifying its IP address range, create subnets, and configure route tables, network gateways and security settings.

A *subnet* is a range of IP addresses in your VPC. We can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet and a private subnet for resources that will not be connected to the internet.

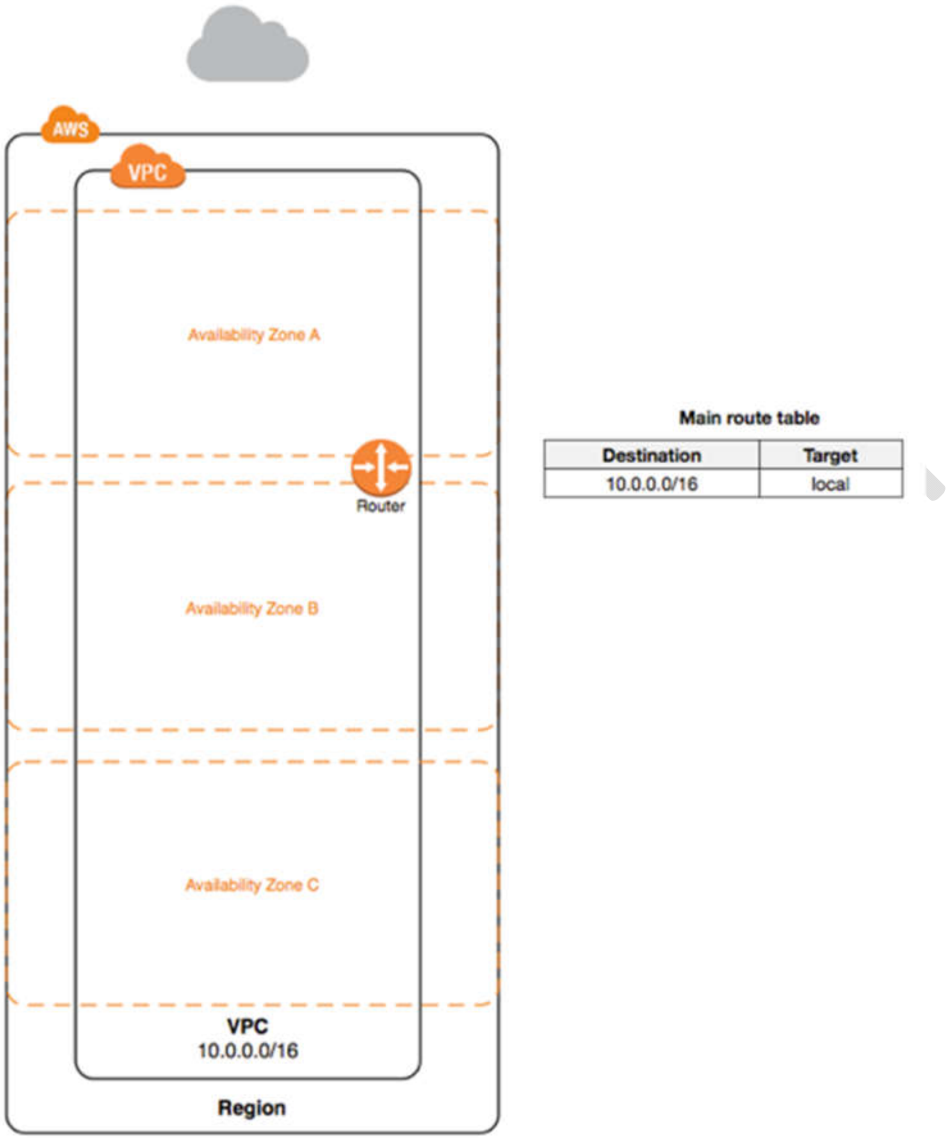
To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL).

VPC and Subnet

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

When we create a VPC, we must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. This is the primary CIDR block for your VPC.

The following diagram shows a new VPC with an IPv4 CIDR block, and the main route table.

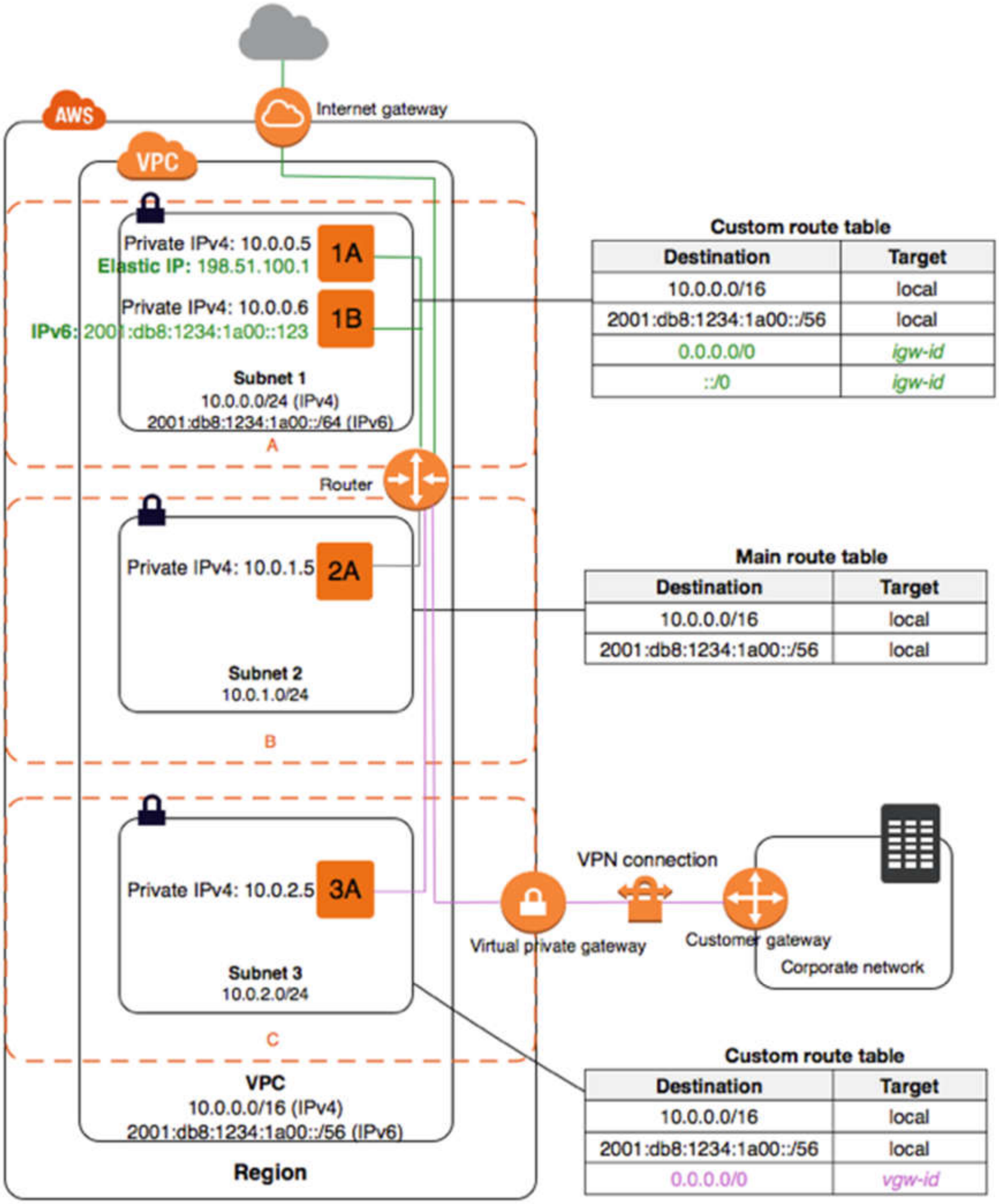


Main route table

Destination	Target
10.0.0.0/16	local

A VPC spans all the Availability Zones in the region. After creating a VPC, you can add one or more subnets in each Availability Zone. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location. We assign a unique ID to each subnet.

You can also optionally assign an IPv6 CIDR block to your VPC and assign IPv6 CIDR blocks to your subnets. The following diagram shows a VPC that has been configured with subnets in multiple Availability Zones. 1A, 1B, 2A, and 3A are instances in your VPC. An IPv6 CIDR block is associated with the VPC, and an IPv6 CIDR block is associated with subnet 1. An internet gateway enables communication over the internet, and a virtual private network (VPN) connection enables communication with your corporate network.



Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	igw-id
::/0	igw-id

Main route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local

Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	vgw-id

If a subnet's traffic is routed to an internet gateway, the subnet is known as a *public subnet*. In this diagram, subnet 1 is a public subnet. If you want your instance in a public subnet to communicate with the internet over IPv4, it must have a public IPv4 address or an Elastic IP address (IPv4). If we want our instance in the public subnet to communicate with the internet over IPv6, it must have an IPv6 address.

If a subnet doesn't have a route to the internet gateway, the subnet is known as a *private subnet*. In this diagram, subnet 2 is a private subnet.

If a subnet doesn't have a route to the internet gateway, but has its traffic routed to a virtual private gateway for a VPN connection, the subnet is known as a *VPN-only subnet*. In this diagram, subnet 3 is a VPN-only subnet. Currently, we do not support IPv6 traffic over a VPN connection.

Note

Regardless of the type of subnet, the internal IPv4 address range of the subnet is always private—we do not announce the address block to the internet.

You have a limit on the number of VPCs and subnets you can create in your account.

VPC and Subnet Sizing

Amazon VPC supports IPv4 and IPv6 addressing and has different CIDR block size limits for each. By default, all VPCs and subnets must have IPv4 CIDR blocks—you can't change this behavior. You can optionally associate an IPv6 CIDR block with your VPC.

VPC and Subnet Sizing for IPv4

When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses). After you've created your VPC, you can associate secondary CIDR blocks with the VPC.

When you create a VPC, we recommend that you specify a CIDR block (of /16 or smaller) from the private IPv4 address ranges as specified in RFC 1918:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

We can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918; however, for the purposes of this documentation, we refer to *private IP addresses* as the IPv4 addresses that are within the CIDR range of your VPC.

The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC), or a subset of the CIDR block for the VPC (for multiple subnets). The allowed block size is between a /28 netmask and /16 netmask. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap. For example, if you create a VPC with CIDR block 10.0.0.0/24, it supports 256 IP addresses. You can break this CIDR block into two subnets, each supporting 128 IP addresses. One subnet uses CIDR block 10.0.0.0/25 (for addresses 10.0.0.0 - 10.0.0.127) and the other uses CIDR block 10.0.0.128/25 (for addresses 10.0.0.128 - 10.0.0.255).

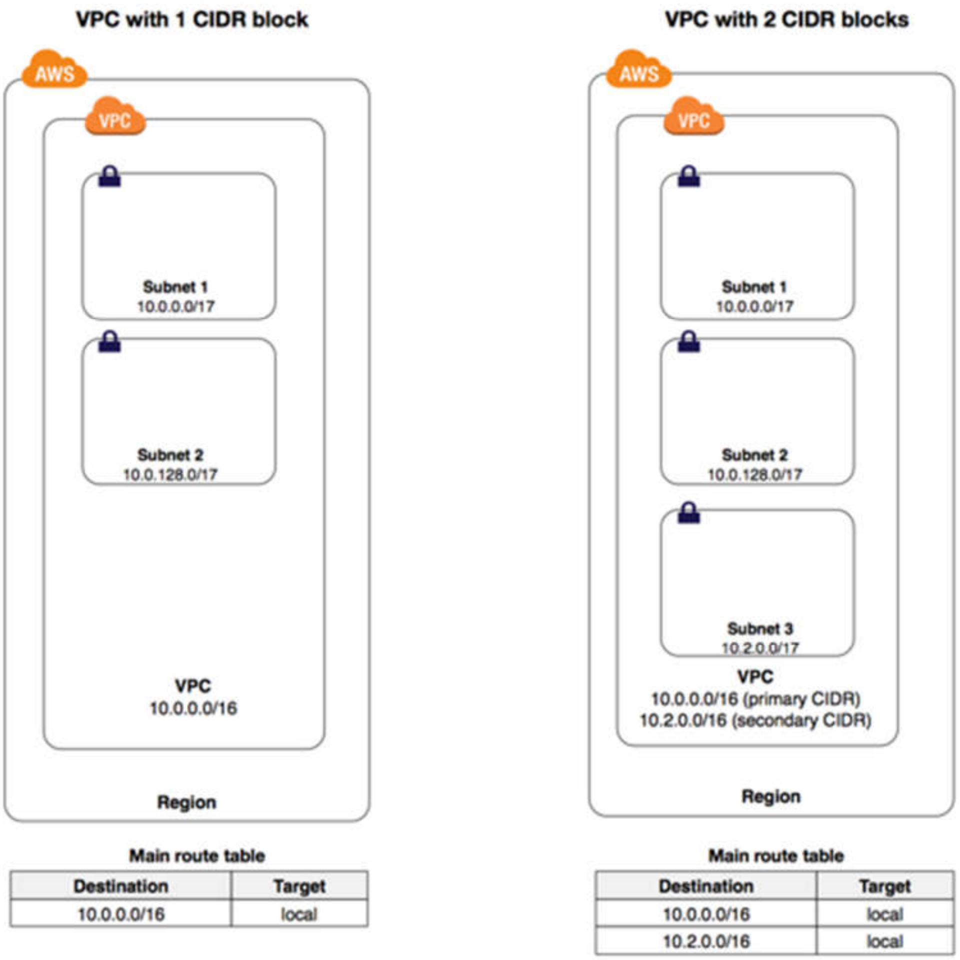
The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance. For example, in a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:

- 10.0.0.0: Network address.
- 10.0.0.1: Reserved by AWS for the VPC router.
- 10.0.0.2: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, we also reserve the base of each subnet range plus two. For VPCs with multiple CIDR blocks, the IP address of the DNS server is located in the primary CIDR.
- 10.0.0.3: Reserved by AWS for future use.
- 10.0.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

Adding IPv4 CIDR Blocks to a VPC

You can associate secondary IPv4 CIDR blocks with your VPC. When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is local).

In the following example, the VPC on the left has a single CIDR block (10.0.0.0/16) and two subnets. The VPC on the right represents the architecture of the same VPC after you've added a second CIDR block (10.2.0.0/16) and created a new subnet from the range of the second CIDR.



Main route table

Destination	Target
10.0.0.0/16	local

Main route table

Destination	Target
10.0.0.0/16	local
10.2.0.0/16	local

To add a CIDR block to our VPC, the following rules apply:

- The allowed block size is between a /28 netmask and /16 netmask.
- The CIDR block must not overlap with any existing CIDR block that's associated with the VPC.
- We cannot increase or decrease the size of an existing CIDR block.
- We have a limit on the number of CIDR blocks you can associate with a VPC and the number of routes we can add to a route table. We cannot associate a CIDR block if this results in we exceeding our limits.
- The CIDR block must not be the same or larger than the CIDR range of a route in any of the VPC route tables. For example, if we have a route with a destination of 10.0.0.0/24 to a virtual private gateway, we cannot associate a CIDR block of the same range or larger. However, we can associate a CIDR block of 10.0.0.0/25 or smaller.
- The following rules apply when you add IPv4 CIDR blocks to a VPC that's part of a VPC peering connection:
    - If the VPC peering connection is active, you can add CIDR blocks to a VPC provided they do not overlap with a CIDR block of the peer VPC.

  - If the VPC peering connection is pending-acceptance, the owner of the requester VPC cannot add any CIDR block to the VPC, regardless of whether it overlaps with the CIDR block of the accepter VPC. Either the owner of the accepter VPC must accept the peering connection, or the owner of the requester VPC must delete the VPC peering connection request, add the CIDR block, and then request a new VPC peering connection.
  - If the VPC peering connection is pending-acceptance, the owner of the accepter VPC can add CIDR blocks to the VPC. If a secondary CIDR block overlaps with a CIDR block of the requester VPC, the VPC peering connection request fails and cannot be accepted.
- When you add or remove a CIDR block, it can go through various states: associating | associated | disassociating | disassociated | failing | failed. The CIDR block is ready for you to use when it's in the associated state.

We can disassociate a CIDR block that we have associated with our VPC; however, we cannot disassociate the CIDR block with which we originally created the VPC (the primary CIDR block). To view the primary CIDR for our VPC in the Amazon VPC console, choose **Your VPCs**, select our VPC, and take note of the first entry under **CIDR blocks**.

Subnet Security

AWS provides two features that you can use to increase security in your VPC: *security groups* and *network ACLs*. Security groups control inbound and outbound traffic for your instances, and network ACLs control inbound and outbound traffic for your subnets. In most cases, security groups can meet your needs; however, you can also use network ACLs if you want an additional layer of security for our VPC.

By design, each subnet must be associated with a network ACL. Every subnet that we create is automatically associated with the VPC's default network ACL. We can change the association, and we can change the contents of the default network ACL.

You can create a flow log on your VPC or subnet to capture the traffic that flows to and from the network interfaces in your VPC or subnet. You can also create a flow log on an individual network interface. Flow logs are published to CloudWatch Logs.

Default and Nondefault VPCs

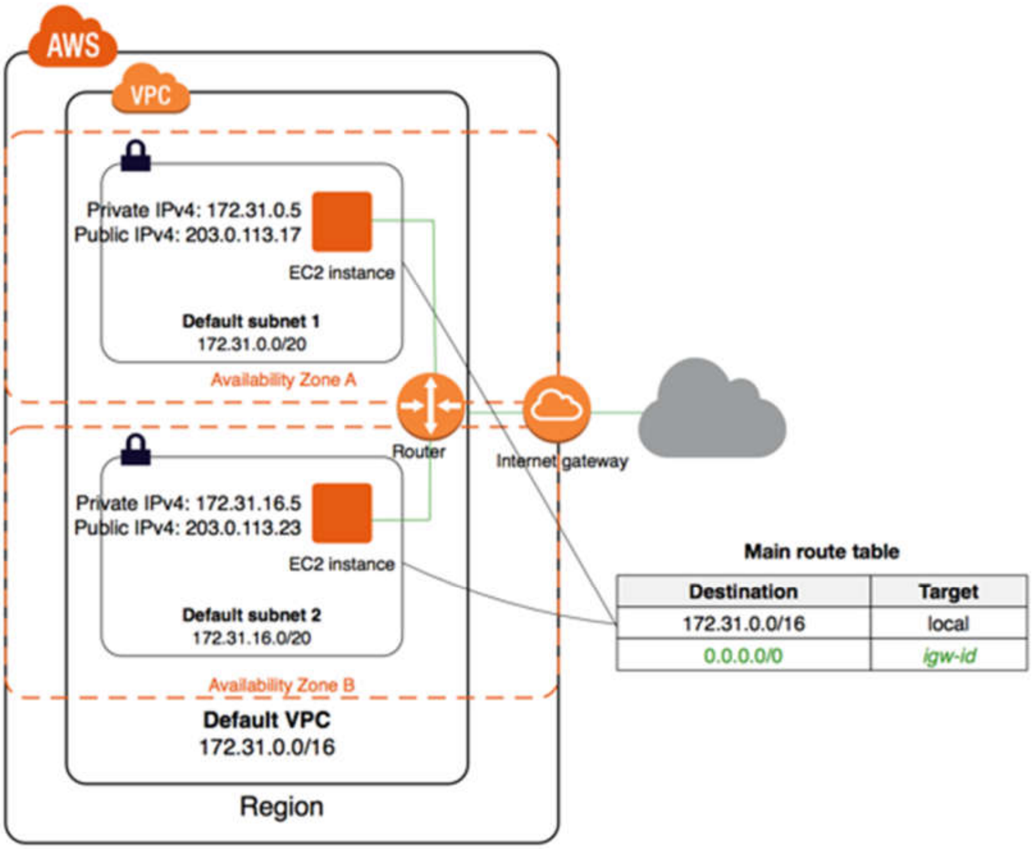
If your account supports the EC2-VPC platform only, it comes with a *default VPC* that has a *default subnet* in each Availability Zone. A default VPC has the benefits of the advanced features provided by EC2-VPC and is ready for us to use. If we have a default VPC and don't specify a subnet when we launch an instance, the instance is launched into our default VPC. You can launch instances into your default VPC without needing to know anything about Amazon VPC.

Regardless of which platforms your account supports, we can create your own VPC, and configure it as we need. This is known as a *nondefault VPC*. Subnets that we create in our nondefault VPC and additional subnets that we create in our default VPC are called *nondefault subnets*.

Accessing the Internet

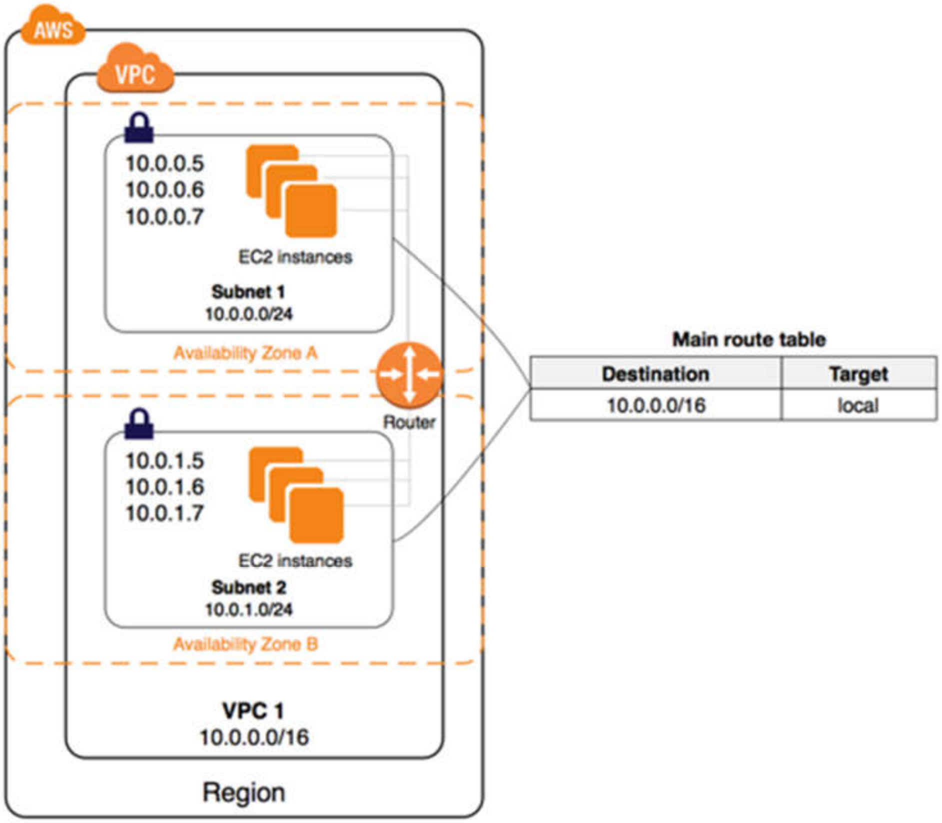
You control how the instances that you launch into a VPC access resources outside the VPC.

Your default VPC includes an internet gateway, and each default subnet is a public subnet. Each instance that you launch into a default subnet has a private IPv4 address and a public IPv4 address. These instances can communicate with the internet through the internet gateway. An internet gateway enables our instances to connect to the internet through the Amazon EC2 network edge.



Destination	Target
172.31.0.0/16	local
0.0.0.0/0	igw-id

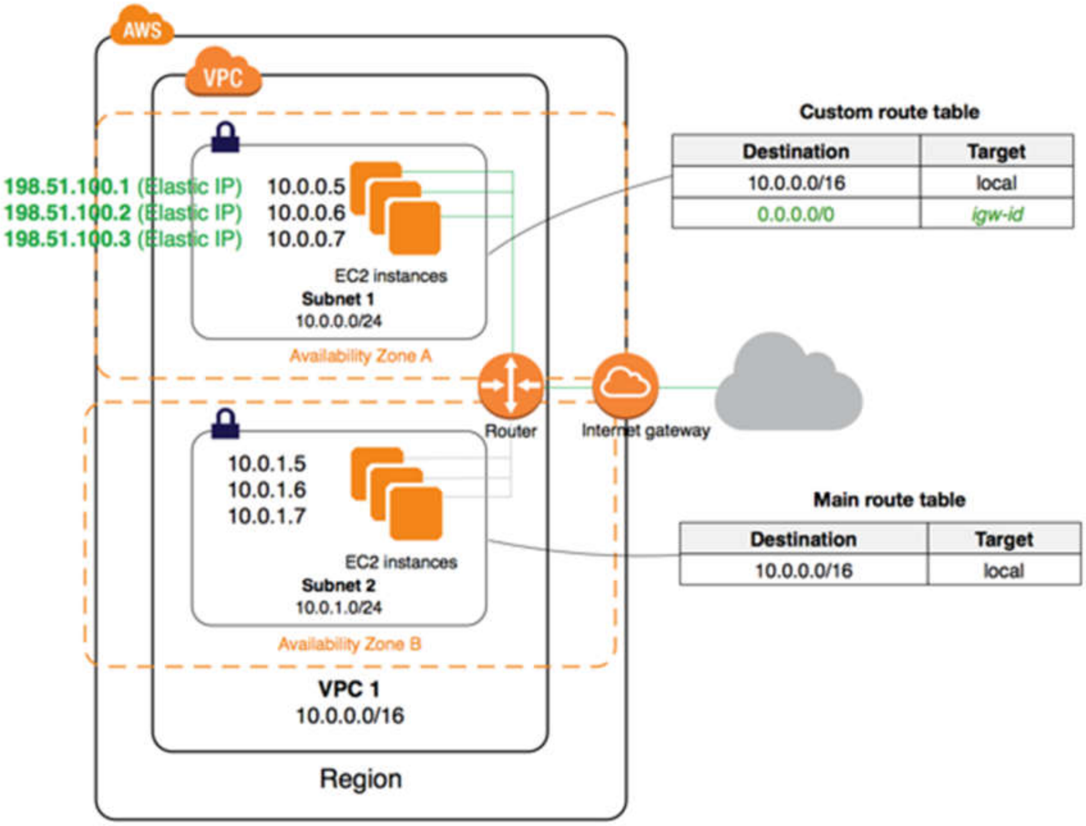
By default, each instance that you launch into a non-default subnet has a private IPv4 address, but no public IPv4 address, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute. These instances can communicate with each other but can't access the internet.



Main route table

Destination	Target
10.0.0.0/16	local

We can enable internet access for an instance launched into a nondefault subnet by attaching an internet gateway to its VPC (if its VPC is not a default VPC) and associating an Elastic IP address with the instance.



Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main route table

Destination	Target
10.0.0.0/16	local

Alternatively, to allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a network address translation (NAT) device for IPv4 traffic. NAT maps multiple private IPv4 addresses to a single public IPv4 address. A NAT device has an Elastic IP address and is connected to the internet through an internet gateway. We can connect an instance in a private subnet to the internet through the NAT device, which routes traffic from the instance to the internet gateway and routes any responses to the instance.