AMAZON VPC & NETWORKING

YOUR LOGICALLY ISOLATED NETWORK IN THE CLOUD

INTRODUCTION 👋

Amazon **V**irtual **P**rivate **C**loud (**VPC**) enables you to define and launch AWS resources in a logically isolated virtual network. It can imitate your local data center, but with all the benefits of the cloud's scalable infrastructure.

INTERNET GATEWAYS 🌎

An **I**nternet **G**ate**w**ay (**IGW**) is an AWS-managed highly-available VPC component that allows resources that reside in **public subnets** to communicate with the internet.

Private subnets do not have a routing connection to the IGW.

NAT DEVICES 🔀

If you need resources in your private subnets to access the internet, you need a **N**etwork **A**ddress **T**ranslations (**NAT**) device that maps multiple of your private IPv4 addresses to a single public IPv4.

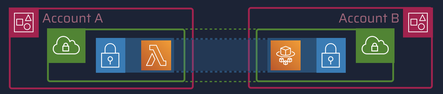
There are two different types of NAT devices at AWS:

- **NAT Gateway** - an AWS-managed gateway
- **NAT Instance** - your own NAT device, running on EC2

The AWS-managed version can result in huge costs, as you're billed for each **running hour** & **GB of traffic that is processed**.

VPC PEERING 🔒

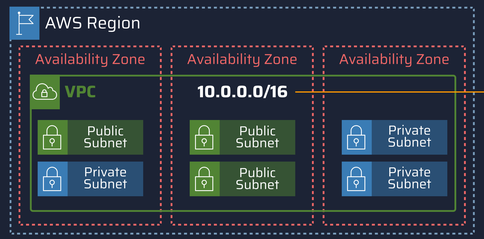
Peering connections allow you to route traffic between two VPCs as if they were in the same VPC. It also allows you to not only connect to VPCs in other regions but also in other AWS accounts.



💡 CIDR blocks for your VPCs **can't overlap**.

VIRTUAL PRIVATE CLOUDS ☁️

Each VPC is created for a region and always spans across all availability zones.



Each of the availability zones can contain subnets that are another break down of your VPC.

SECURITY GROUPS 🔒

Security **G**roups (**SG**) define allow rules for your traffic - **inbound** or **outbound**. They enable traffic filtering based on protocols and port numbers.

Inbound Rules ↓

Source	Protocol	Port Range
0.0.0.0/0	TCP	80
::/0	TCP	80
0.0.0.0/0	TCP	22

Outbound Rules ↑

Destination	Protocol	Port Range
0.0.0.0/0	TCP	1433
::/0	TCP	1433
0.0.0.0/0	TCP	3306

SGs operate on **instance level** and are **stateful**.

VPC SHARING 💞

Share a VPC with other accounts that are part of the same AWS Organization, so that multiple accounts can launch resources into the centrally-managed subnets but still be in full control of their resources. Participating accounts can't modify resources in shared subnets that they do not own.

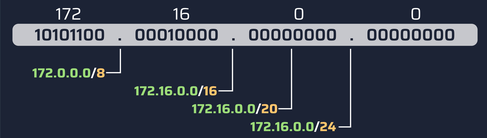
This allows for a fine-grained separation of accounts for billing and access control, but still having components with high interconnectivity.

CIDR 📝

Your VPC needs a range of IPv4 addresses that can be used by attached network interfaces. They are defined as **C**lassless **I**nter-**D**omain **R**outing (**CIDR**) blocks.

They are made up of two number sets:

- **prefix**: the binary representation of the address
- **suffix**: the total number of bits in the entire address

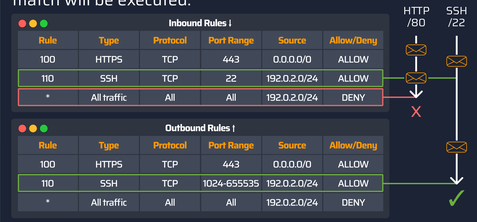


The allowed block size for a VPC is between 65,536 (netmask **/16**) and 16 IP (netmask **/24**) addresses.

NETWORK ACLS 🔐

Network **A**ccess **C**ontrol **L**ists (**NACL**s) act as a firewall on network level. They can hold one or multiple **allow** and/or **deny** rules which are ordered via priority numbers.

Evaluation starts with the lowest rule number and the first match will be executed.



Each subnet **must be assigned** to a network ACL and return traffic **must be explicitly allowed** (NACLs are stateless).

PREFIX LISTS 📒

You're able to bind one or several CIDR blocks into a prefix list that can be later used within your security groups or route tables. This reduces the efforts of referencing each of the CIDR blocks individually.

SUBNETS ⚛️

We separate between **public** and **private** subnets:

- **public**: for publicy available resources.
- **private**: for resources that only need to be access internally and therefore do not need a public IP address.

Resources in each subnet can be protected with **multiple layers of security**, e.g. with **S**ecurity **G**roups (**SG**s) or **N**etwork **A**ccess **C**ontrol **L**ists (**ACL**s).

💡 There are a lot of services that can only be launched into a VPC like EC2 instances - some even **require** a private subnet like an ElastiCache cluster.

VPC FLOW LOGS 🌊

Some requests are not reaching your instance? Do you want to get insights into how traffic is flowing within your VPC?

You can monitor your VPC via **Flow Logs**. Those logs capture details about how IP traffic is going to and from network interfaces in your VPC. The logs can be shipped to either CloudWatch, S3, or Kinesis Data Firehose.

```
[...] eni-5123b7ac012345678 219.42.22.48  172.16.0.101  [...] ACCEPT OK
[...] eni-5123b7ac012345678 172.31.16.139 219.42.22.48  [...] REJECT OK
```

Looking at the **example flow logs** above, an incoming request was accepted, but the response rejected. This could happen even if you've defined allow rules for inbound traffic in your security group and network ACLs. As security groups are stateful, responses are allowed. ACLs are not stateful, so a missing outbound allow rule does result in a rejection.

IAM INTEGRATION 🔑

Amazon VPC is fully integrated with IAM and there are no additional costs. Create roles and policies to define which principal can perform actions on what resources, and under what conditions.

💡 VPC shares its API namespace with Amazon EC2.

ROUTE TABLES 📚

Traffic inside your VPC needs directions. That's why you can create route tables, which are sets of rules that you can associate with a subnet (custom route tables).

Each route table entry needs a **destination** and **target** which defines how traffic is routed.

- **destination**: a range of IP addresses where traffic should go to defined as a CIDR block. e.g.an external corporate defined as 172.16.0.0/12.
- **target**: the gateway, network interface, or connection through which to send the destination traffic, e.g. an internet gateway.

Each of your VPCs comes with a **default route table** (= **main route table**) that controls traffic for subnets which do not have a custom route table attached.

DEFAULT VPC 🌤️

Each AWS account created after the end of 2013 comes with a default VPC per region.

Each of those default VPCs also has a public subnet in each availability zone, an internet gateway and settings to enable DNS resolution.

DHCP OPTION SETS 📝

Each device in a VPC requires an IP address to communicate over the network. You don't need to manually assign them but rely on DHCP servers that are using the **D**ynamic **H**ost **C**onfiguration **P**rotocol.

Amazon VPC allows you to further control information returned by the AWS-managed DHCP servers via DHCP option sets. This for example allows you to use your own domain name server that should be used for domain name resolution in your network.

aws