AWS SINGLE VPC DESIGN

“How should I design my VPC?”

Overview

Amazon Virtual Private Cloud (Amazon VPC) offers a comprehensive set of virtual networking capabilities that provide AWS customers with many options for designing and implementing their AWS network. However, it can be difficult to decide on an ideal network design for your organization from among the various network strategies and configuration permutations, especially for customers who do not have dedicated network professionals to design, size, and manage their networks. This document provides AWS customers with high-level VPC design best practices and guidance, and outlines the most commonly used individual VPC configurations.

The following sections address key considerations and recommendations for designing and sizing individual VPCs, and assume some basic knowledge of network addressing, subnetting, and routing.[1]

General Best Practices

When configuring any computer network, there are some universal network design principles to consider. For example, it’s best to implement non-overlapping network ranges for your private networks to simplify the ability to route between remote networks. Also, it’s a good idea to set aside some IP address space for future use, as network requirements change over time. Keeping general networking best practices in mind, VPCs should adhere to the following high-level design best practices:

- Ensure that your VPC network range (CIDR block) does not overlap with your organization’s other private network ranges.
- Do not allocate all network addresses at once; instead ensure that you reserve some address space for future use.
- Divide your VPC network range evenly across all available Availability Zones (AZs) in a region.
- Create one subnet per available AZ for each group of hosts that have unique routing requirements (e.g., public vs. private subnets).
- Size your VPC CIDR and subnets to support significant growth for the expected workload(s).

Application on the AWS Platform

As you decide on an appropriate VPC design, you will need to consider both how you plan to leverage AWS from a user, backend system, and routing perspective, as well as estimate your current and future network sizing needs. Keep in mind that you can easily change your VPC design from one configuration to another, especially if you follow the best practice of reserving some address space for future use. For example, you can add private subnets to the *Internet-Accessible VPC* design described below to create a *Public and Privately Routed VPC* design. However, keep in mind that you cannot resize a VPC or individual subnet once you create it.

Common VPC Configurations

In order to select the appropriate configuration for your use, you must identify *who* will need to access AWS-hosted resources, *what* additional systems will be required to support the AWS-hosted resources, and *where* these entities are located from a network-routing perspective. Each of the following design patterns is based on these three criteria:

User Access: *Who needs to access the resources in the network (internal users, external users, or both)?*

Systems Access: *What additional systems (internal, external, or shared) does this workload need to interact with?*

Routing: *Do you need different hosts to route traffic in different ways (e.g., internal-only vs. public-facing hosts)?*

[1] Please see the Wikipedia article on CIDR address allocation for background information: https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

Based on the answers to these questions, individual VPCs are typically deployed in one of these common configurations.

Internet-Accessible VPC

VPC Wizard Scenario VPC with a Single Public Subnet

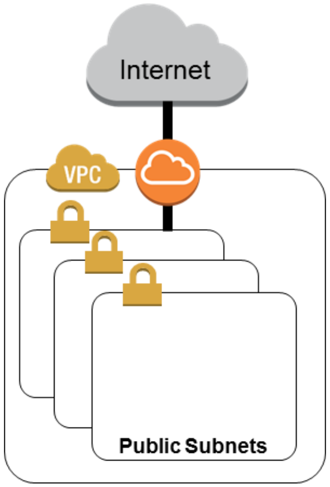
Use Case This design pattern is primarily used for test, R&D, sales demo, production, and other environments that require a network environment that is completely isolated from a customer's internal network. This design is appropriate for customers who want to ensure that AWS resources within this network never have private network connectivity to internal systems.

User Access Any user (internal or external) accessing resources in this VPC must do so over the Internet.

Systems Access AWS resources within the VPC network can be granted access to other publicly addressable systems on the Internet.

Routing AWS resources in this environment all share the same routing requirements and leverage an Internet Gateway and AWS-provided Internet connections.

Considerations AWS resources within the VPC network will not be able to connect directly to the customer's internal systems and networks using private IP addresses.



Public and Privately Routed VPC

VPC Wizard Scenario VPC with Public and Private Subnets (NAT)

Use Case This design pattern is used to create a network environment that has the ability to communicate with both internal (privately routed) and external (publicly routed) resources using a combination of public and private connections. This design is ideal for workloads that need to accommodate a combination of public and private routing needs, such as all-in Internet-facing, multi-tier web applications supported by databases or other privately routed backend systems.

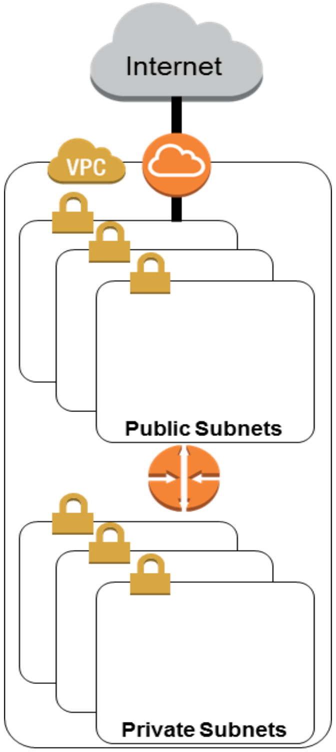
User Access This design pattern supports both internal and external users and can leverage either private network connections or AWS-provided Internet connections, depending on what makes the most sense for the given workload.

Systems Access AWS resources within the VPC network can be granted access to internal and external systems and can leverage either private network connections or AWS-provided Internet connections, depending on what makes the most sense for the application and workload.

Routing AWS resources in this environment should be grouped by unique routing requirements and deployed to appropriate public or private subnets to ensure both internal and Internet traffic will be routed properly.

Considerations Designs that support both internal and external network communication might require additional network controls to protect both AWS and internal systems. See the Controlling VPC Egress Traffic Solution Brief for additional information.

Additionally, this design is often incorporated in a larger, all-in network design involving multiple VPCs.



On-Premises and Internet-Accessible VPC

VPC Wizard Scenario VPC with Public and Private Subnets and Hardware VPN Access

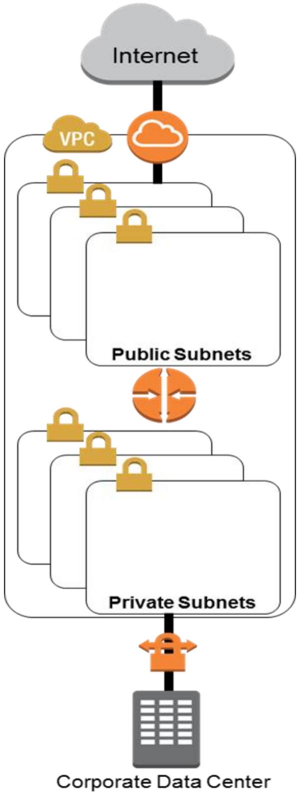
Use Case This design pattern is used to create a network environment that has the ability to communicate with both on-premises (privately routed) and external (publicly routed) resources using a combination of private connections and AWS-provided public routing. This design is ideal for Internet-facing workloads that need to interact with a customer's internal network and on-premises systems, internally facing systems that can benefit from AWS-provided Internet connections, or any other workload that requires a combination of public and private routing.

User Access This design pattern supports both internal and external users and can leverage either private network connections or AWS provided Internet connections, depending on what makes the most sense for the user's location and network connectivity.

Systems Access AWS resources within the VPC network can be granted access to internal and external systems and can leverage either private network connections or AWS provided Internet connections, depending on what makes the most sense for the application and workload.

Routing AWS resources in this environment should be grouped by unique routing requirements and deployed to appropriate public or private subnets to ensure both internal and external traffic will be routed properly.

Considerations Designs that support both internal and external network communication might require additional network controls to protect both AWS and internal systems. See the Controlling VPC Egress Traffic Solution Brief for additional information.



Internal-Only VPC

VPC Wizard Scenario VPC with a Private Subnet Only and Hardware VPN Access

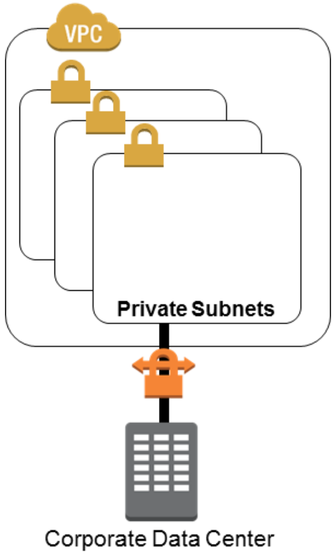
Use Case This design pattern is used to create a network environment that is only accessible from an existing, internal network, such as internally facing or back-office systems. This VPC design is appropriate for customers who want to utilize their own Internet Service Providers to control all Internet-based traffic, and forgo AWS-provided public IP addressing capabilities.

User Access Any user (internal or external) accessing resources in this VPC must do so over the customer's existing, internal network and customer-managed Internet connections.

Systems Access AWS resources within the VPC network can be granted access to a customer's internal systems and will leverage the customer's Internet providers for all publicly routed traffic.

Routing AWS resources in this environment all share the same routing requirements and leverage VPN or Direct Connect connections to communicate with the customer's internal network.

Considerations Amazon Elastic Compute Cloud (Amazon EC2) instances in this VPC will be able to access Amazon Simple Storage Service (Amazon S3) through VPC *endpoints*. However, access to other AWS publicly addressable services must traverse the customer's internal network and Internet connections.



VPC Sizing

VPCs can vary in size from 16 addresses (/28 netmask) to 65,536 addresses (/16 netmask). In order to size a VPC correctly, it is important to understand the number, types, and sizes of workloads expected to run in it, as well as workload elasticity and load balancing[2] requirements. Keep in mind that there is no charge for using Amazon VPC (aside from EC2 charges), therefore cost should not be a factor when determining the appropriate size for your VPC, so make sure you size your VPC for growth.

Moving workloads or AWS resources between networks is not a trivial task, so be generous in your IP address estimates to give yourself plenty of room to grow, deploy new workloads, or change your VPC design configuration from one to another. The majority of AWS customers use VPCs with a /16 netmask and subnets with /24 netmasks. The primary reason AWS customers select smaller VPC and subnet sizes is to avoid overlapping network addresses with existing networks.

The following table gives examples of potential VPC and subnet sizing:

VPC Size	Netmask	Subnet Size	Hosts/Subnet*	Subnets/VPC	Total IPs*
Micro	/24	/27	27	8	216
Small	/21	/24	251	8	2008
Medium	/19	/22	1019	8	8152
Large	/18	/21	2043	8	16344
Extra Large	/16	/20	4091	16	65456

* Note that the *Hosts/Subnet* and *Total IPs* columns in the above table exclude the five (5) IP addresses from each subnet that are unavailable for use.[3]

Resources

VPC Security Capabilities	https://d0.awsstatic.com/aws-answers/VPC_Security_Capabilities.pdf AWS Solution Brief on network security features for Amazon VPC
Controlling VPC Egress Traffic	https://d0.awsstatic.com/aws-answers/Controlling_VPC_Egress_Traffic.pdf AWS Solution Brief on approaches to control network traffic out of a VPC
Amazon VPC Quick Start Reference Deployment	https://aws.amazon.com/quickstart/architecture/vpc/ AWS Quick Start Reference Deployment that builds and Amazon VPC environment with public and private subnets.
Amazon VPC Documentation	https://aws.amazon.com/documentation/vpc/ AWS webpage with links to VPC technical documentation, including introductory material (*Getting Started Guide*), component and strategy overviews (*User Guide*), and more robust technical documentation (*Network Administrator Guide*)

[2] http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/setting-up-elb.html#set-up-ec2

[3] http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#SubnetSize