Extend Your IT Infrastructure with Amazon Virtual Private Cloud

December 2018

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network you define. This paper provides an overview of how you can connect an Amazon VPC to your existing IT infrastructure while meeting security and compliance requirements. This allows you to access AWS resources as though they are a part of your existing network.

Introduction

With Amazon Virtual Private Cloud (Amazon VPC), you can provision a private, isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. With Amazon VPC, you can define a virtual network topology that closely resembles a traditional network that you might operate in your own data center. You have complete control over your virtual networking environment, including selection of your own IPv4 address range, creation of subnets, and configuration of route tables and network gateways. For example, with VPC you can:

- Expand the capacity of existing on-premises infrastructure.
- Launch a backup stack of your environment for disaster recovery purposes.
- Launch a Payment Card Industry Data Security Standard (PCI DSS) compliant website that accepts secure payments.
- Launch isolated development and testing environments.
- Serve virtual desktop applications within your corporate network.

In a traditional approach to these use cases, you would need a lot of upfront investment to build your own data center, provision the required hardware, acquire the necessary security certifications, hire system administrators, and keep everything running. With VPC on AWS, you have little upfront investment, and you can scale your infrastructure in or out as necessary. You get all of the benefits of a secure environment at no extra cost; AWS security controls, certifications, accreditations, and features meet the security criteria required by some of the most discerning and security-conscious customers in large enterprise as well as governmental agencies. For a full list of certifications and accreditations, see the AWS Compliance Center.

This paper highlights common use cases and best practices for Amazon VPC and related services.

Understanding Amazon Virtual Private Cloud

Amazon VPC is a secure, private, and isolated section of the AWS cloud where you can launch AWS resources in a virtual network topology that you define. When you create a VPC, you provide a set of private IPv4 addresses that you want instances in your VPC to use. You specify this set of addresses in the form of a Classless Inter-Domain

Routing (CIDR) block, for example 10.0.0.0/16. You can assign block sizes of between /28 (16 IPv4 addresses) and /16 (65,536 IPv4 addresses).

You can also add a set of IPv6 addresses to your VPC. IPv6 addresses are allocated from an Amazon-owned range of addresses, and the VPC receives a /56 (more than 10^{21} IPv6 addresses).

In Amazon VPC, each Amazon Elastic Compute Cloud (Amazon EC2) instance has a default network interface that is assigned a primary private IP address on your Amazon VPC network. You can create and attach additional elastic network interfaces (ENI) to any Amazon EC2 instance in your VPC. Each ENI has its own MAC address. It can have multiple IPv6 or private IPv4 addresses, and it can be assigned to a specific security group. The total number of supported ENIs and private IP addresses per instance depends on the instance type. The ENIs can be created in different subnets within the same Availability Zone and attached to a single instance to build, for example, a low-cost management network or network and security appliances. The secondary ENIs and private IP addresses can be moved within the same subnet to other instances for low-cost, high-availability solutions. To each private IPv4 address, you can associate a public elastic IPv4 address to make the instance reachable from the Internet. IPv6 addresses are the same whether inside the VPC or on the public Internet (if the subnet is public). You can also configure your Amazon EC2 instance to be assigned a public IPv4 address at launch. Public IPv4 addresses are assigned to your instances from the Amazon pool of public IPv4 addresses; they are not associated with your account. With support for multiple IPv6 addresses, private IPv4 addresses, and Elastic IP addresses, you can, among other things, use multiple SSL certificates on a single server and associate each certificate with a specific IP address.

There are some default limits on the number of components you can deploy in your VPC, as documented in Amazon VPC Limits. To request an increase in any of these limits, fill out the Amazon VPC Limits form.

Different Levels of Network Isolation

You can set up your VPC subnets as public, private, or VPN-only. In order to set up a public subnet, you have to configure its routing table so that traffic from that subnet to the Internet is routed through an Internet gateway associated with the VPC, as shown in Figure 1. By assigning EIP addresses to instances in that subnet, you can make them reachable from the Internet over IPv4 as well. It is a best practice to restrict both

ingress and egress traffic for these instances by leveraging stateful security group rules for your instances. You can also use network address translation (NAT) gateways (for IPv4 traffic) and egress-only gateways (for IPv6 traffic) on private subnets to enable them to reach Internet addresses without allowing inbound traffic. Stateless network filtering can also be applied for each subnet by setting up network access control lists (ACLs) for the subnet.

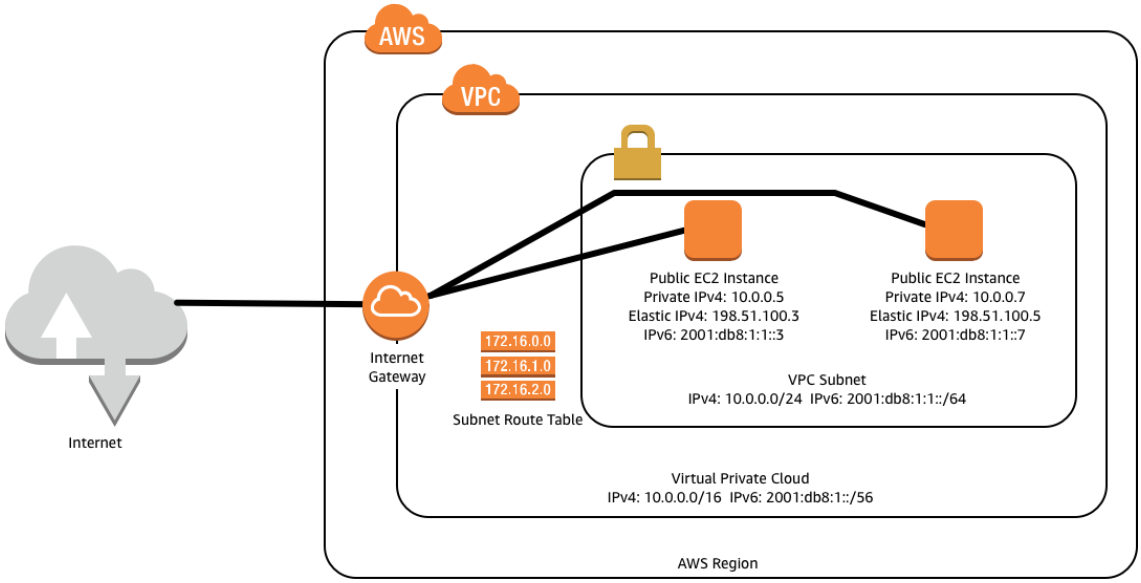


Figure 1: Example of a VPC with a public subnet only

For private subnets, traffic to the Internet can be routed through a NAT gateway or NAT instance with a public EIP that resides in a public subnet. This configuration allows your resources in the private subnet to connect outbound traffic to the Internet without allocating Elastic IP addresses or accepting direct inbound connections. AWS provides a managed NAT gateway, or you can use your own Amazon EC2 based NAT appliance. Figure 2 shows an example of a VPC with both public and private subnets using an AWS NAT gateway.

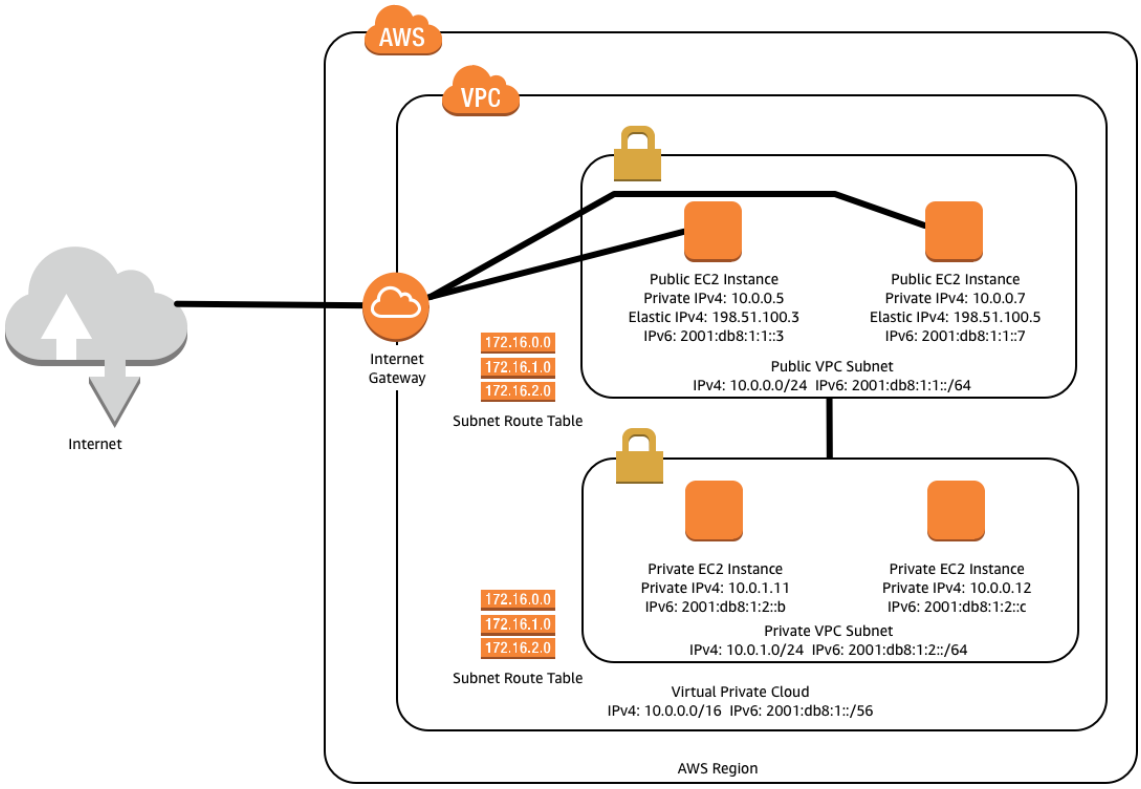


Figure 2: Example of a VPC with public and private subnets

By attaching a virtual private gateway to your VPC, you can create a VPN connection between your VPC and your own data center for IPv4 traffic, as shown in Figure 3. The VPN connection uses industry-standard IPsec tunnels (IKEv1-PSK, with encryption using AES-256 and HMAC-SHA-2 with various Diffie-Hellman groups) to mutually authenticate each gateway and to protect against eavesdropping or tampering while your data is in transit. For redundancy, each VPN connection has two tunnels, with each tunnel using a unique virtual private gateway public IPv4 address.

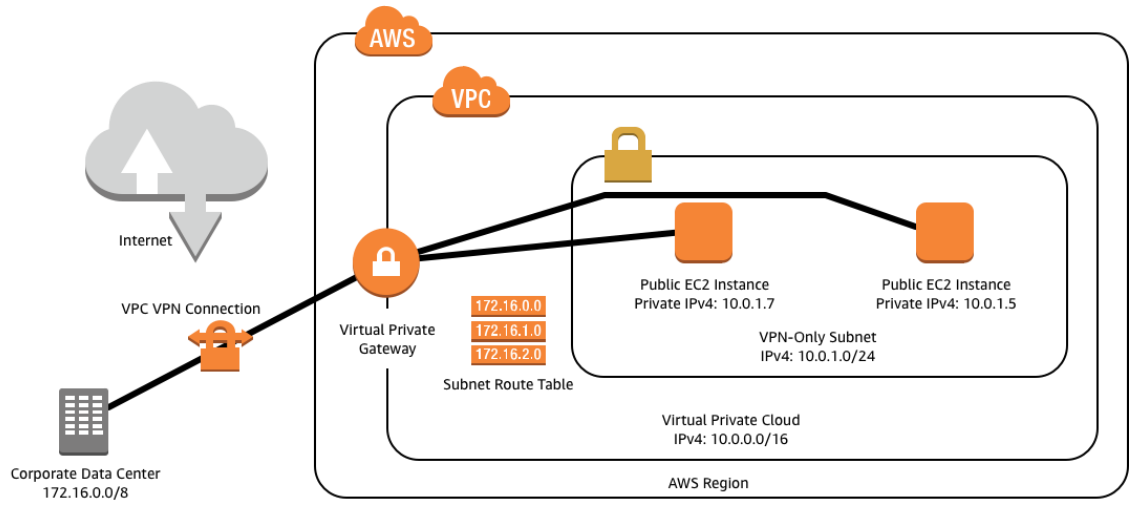


Figure 3: Example of a VPC isolated from the Internet and connected through VPN to a corporate data center

You have two routing options for setting up a VPN connection: dynamic routing using Border Gateway Protocol (BGP) or static routing. For BGP, you need the IPv4 address and the BGP autonomous system number (ASN) of the customer gateway before attaching it to a VPC. Once you have provided this information, you can download a configuration template for a number of different VPN devices and configure both VPN tunnels. For devices that do not support BGP, you may set up one or more static routes back to your on-premises network by providing the corresponding CIDR ranges when you configure your VPN connection. You then configure static routes on your VPN customer gateway and on other internal network devices to route traffic to your VPC via the IPsec tunnel.

If you choose to have only a virtual private gateway with a connection to your on-premises network, you can route your Internet-bound traffic over the VPN and control all egress traffic with your existing security policies and network controls.

You can also use AWS Direct Connect to establish a private logical connection from your on-premises network directly to your Amazon VPC. AWS Direct Connect provides a private, high-bandwidth network connection between your network and your VPC. You can use multiple logical connections to establish private connectivity to multiple VPCs while maintaining network isolation.

With AWS Direct Connect, you can establish 1 Gbps or 10 Gbps dedicated network connections between AWS and any of the AWS Direct Connect locations. A dedicated connection can be partitioned into multiple logical connections by using industry standard 802.1Q VLANs. In this way, you can use the same connection to access public

resources, such as objects stored in Amazon Simple Storage Service (Amazon S3) that use publicly accessible IPv4 and IPv6 addresses, and private resources, such as Amazon EC2 instances that are running within a VPC using Amazon owned IPv6 space or private IPv4 space—all while maintaining network separation between the public and private environments. You can choose a partner from the AWS Partner Network (APN) to integrate the AWS Direct Connect endpoint in an AWS Direct Connect location with your remote networks.

Figure 4 shows a typical AWS Direct Connect setup.

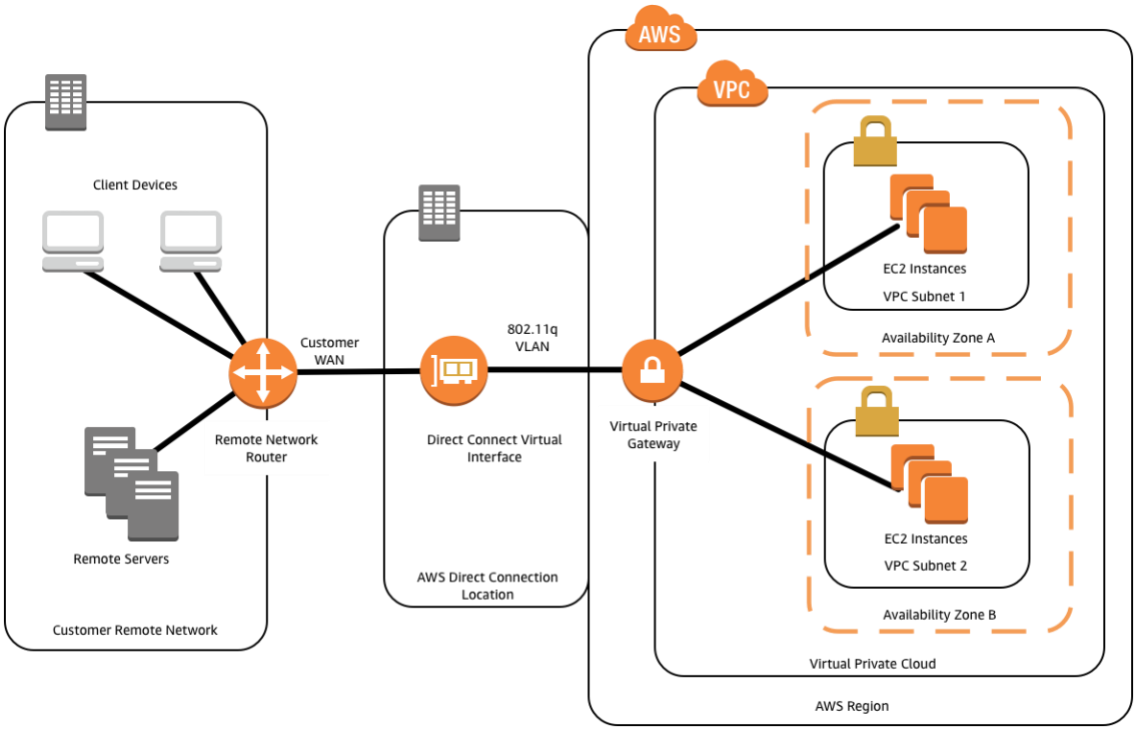


Figure 4: Example of using VPC and AWS Direct Connect with a customer remote network

Finally, you may combine all of these different options in any combination that make the most sense for your business and security policies. For example, you could attach a VPC to your existing data center with a virtual private gateway and set up an additional public subnet to connect to other AWS services that do not run within the VPC, such as Amazon S3, Amazon Simple Queue Service (Amazon SQS) or Amazon Simple Notification Service (Amazon SNS). In this situation, you could also leverage IAM Roles for Amazon EC2 for accessing these services and configure IAM policies to only allow access from the Elastic IP address of the NAT server.

Example Scenarios

Because of the inherent flexibility of Amazon VPC, you can design a virtual network topology that meets your business and IT security requirements for a variety of different use cases. To understand the true potential of Amazon VPC, let's take a few of the most common use cases:

- Host a PCI-compliant e-commerce website
- Build a development and test environment
- Plan for disaster recovery and business continuity
- Extend your data center into the cloud
- Create branch office and business unit networks

Host a PCI-Compliant E-Commerce Website

E-commerce websites often handle sensitive data, such as credit card information, user profiles, and purchase history. As such, they require a Payment Card Industry Data Security Standard (PCI DSS) compliant infrastructure in order to protect sensitive customer data.

Because AWS is accredited as a Level 1 service provider under PCI DSS, you can run your application on PCI-compliant technology infrastructure for storing, processing, and transmitting credit card information in the cloud. As a merchant, you still have to manage your own PCI certification, but by using an accredited infrastructure service provider, you don't need to put additional effort into PCI compliance at the infrastructure level. For more information about PCI compliance, see the AWS Compliance Center.

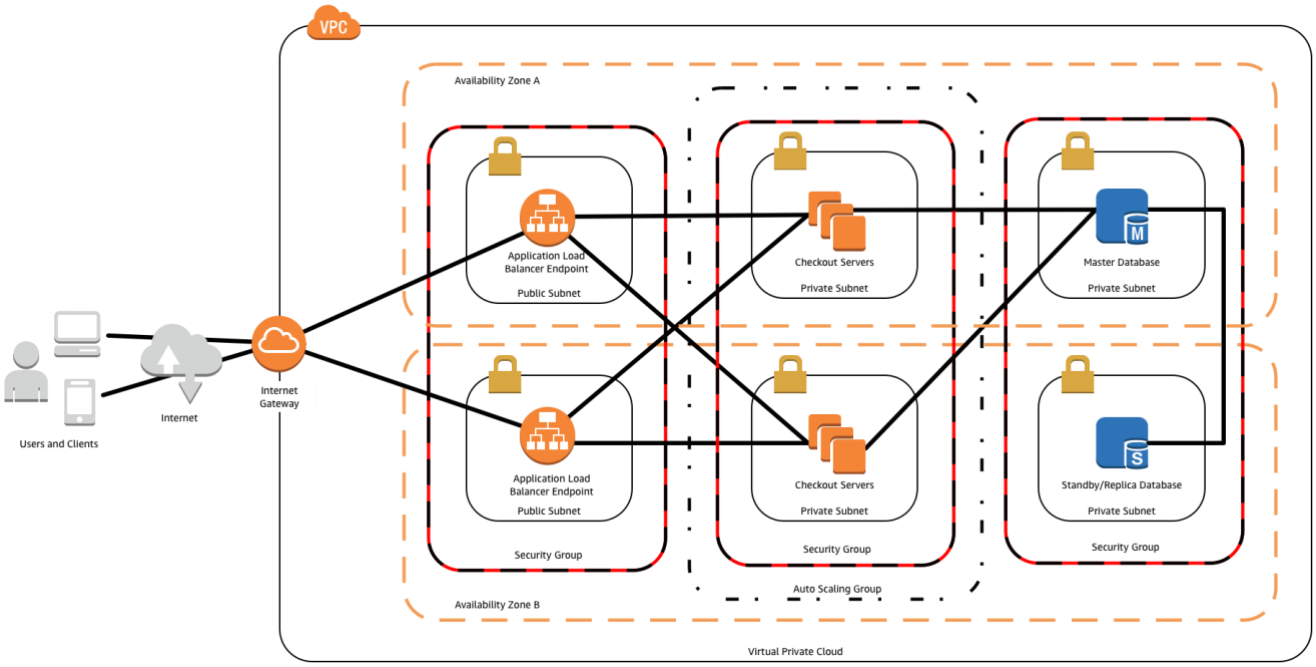
For example, you can create a VPC to host the customer database and manage the checkout process of your e-commerce website. To offer high availability, you set up private subnets in each Availability Zone within the same region and then deploy your customer and order management databases in each Availability Zone. Your checkout servers will be in an Auto Scaling group over several private subnets in different Availability Zones. Those servers will be behind an elastic load balancer that spans public subnets across all used Availability Zones, and the elastic load balancer can be protected by an AWS web application firewall (WAF). By combining VPC, subnets, network ACLs, and security groups, you have fine-grained control over access to your AWS infrastructure. You'll be prepared for the main challenges—scalability, security,

elasticity, and availability—for the most sensitive part of commerce websites. Figure 5 shows an example of an E-Commerce architecture.

Figure 5: Example of an E-Commerce architecture

Build a Development and Test Environment

Software environments are in constant flux, with new versions, features, patches, and updates. Software changes must often be deployed rapidly, with little time to carry out regression testing. Your ideal test environment would be an exact replica of your



production environment where you would apply your updates and then test them against a typical workload. When the update or new version passes all tests, you can roll it into production with greater confidence.

To build such a test environment in-house, you would have to provision a lot of hardware that would go unused most of the time. Sometimes this unused hardware is subsequently repurposed, leaving you without your test environment when you need it. Amazon VPC can help you build an economical, functional, and isolated test environment that simulates your live production environment that can be launched when you need it and shut down when you're finished testing. You don't have to buy expensive hardware; you are more flexible and agile when your environment changes; your test environment can transparently interact within your on-premises network by using LDAP, messaging, and monitoring; and you pay AWS only for what you actually

use. This process can even be fully automated and integrated into your software development process. Figure 6 shows an example of development, test, and production environments within different VPCs.

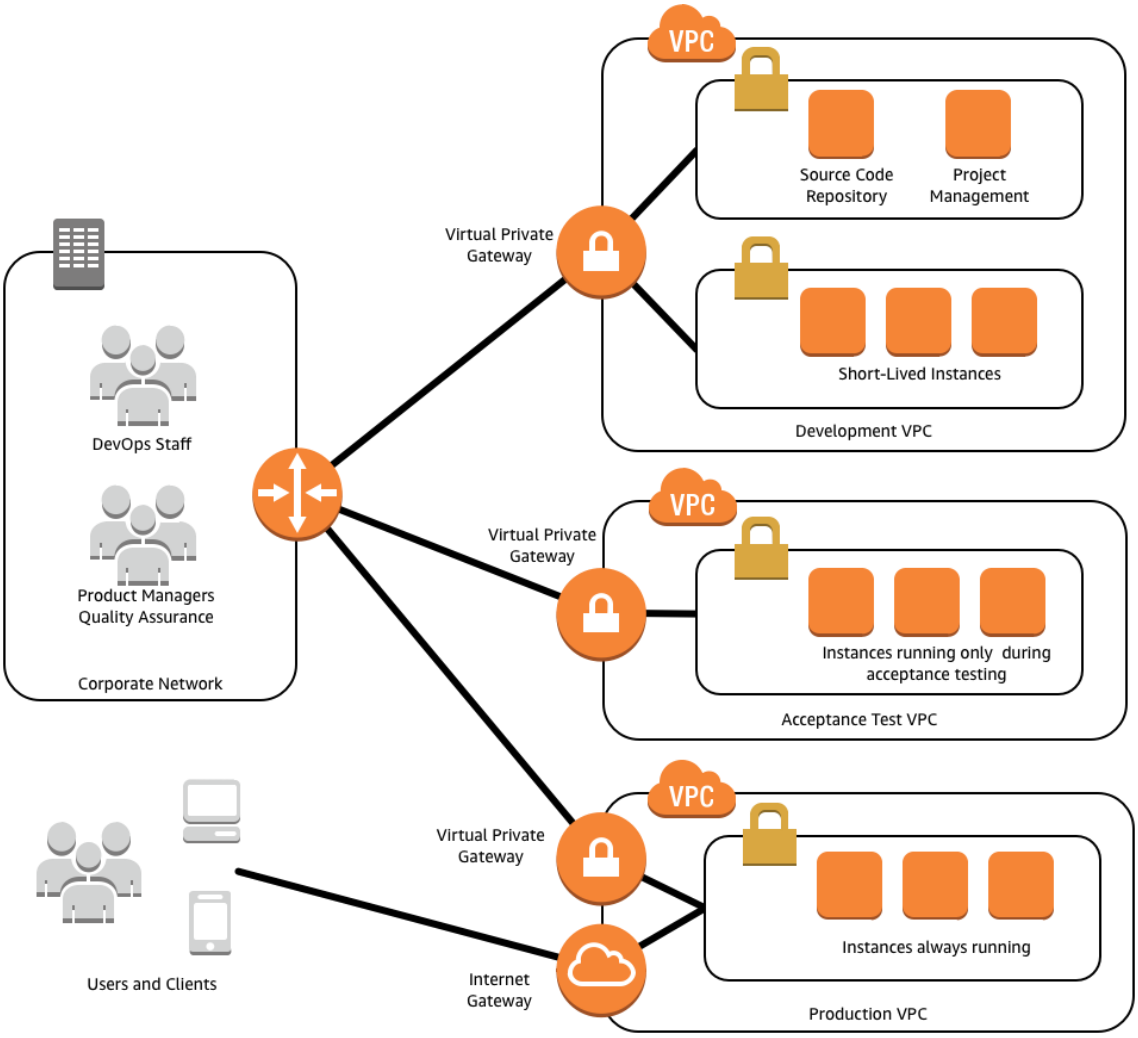


Figure 6: Example of development, test, and production environments

The same logic applies to experimental applications. When you are evaluating a new software package that you want to keep isolated from your production environment, you can install it on a few Amazon EC2 instances inside your test environment within a VPC and then give access to a selected set of internal users. If all goes well, you can transition these images into production and terminate unneeded resources.

Plan for Disaster Recovery and Business Continuity

The consequences of a disaster affecting your data center can be devastating for your business if you are not prepared for such an event. It is worth spending time devising a strategy to minimize the impact on your operations when these events happen. Traditional approaches to disaster recovery usually require labor-intensive backups and expensive standby equipment. Instead, consider including Amazon VPC in your disaster recovery plan. The elastic, dynamic nature of AWS is ideal for disaster scenarios where there are sudden spikes in resource requirements.

Start by identifying the IT assets that are most critical to your business. As in the test environment described previously in this paper, you can automate the replication of your production environment to duplicate the functionality of your critical assets. Using automated processes, you can back up your production data to Amazon Elastic Block Store (Amazon EBS) volumes or Amazon S3 buckets. Database contents can be continually replicated to your AWS infrastructure using AWS Database Migration Service (AWS DMS). You can write declarative AWS CloudFormation templates to describe your VPC infrastructure stack, which you can launch automatically in any AWS region or Availability Zone.

In the event of a disaster, you can quickly launch a replication of your environment in the VPC and then direct your business traffic to those servers. If a disaster involves only the loss of data from your in-house servers, you can recover it from the Amazon EBS data volumes that you've been using as backup storage.

For more information, read Using Amazon Web Services for Disaster Recovery, which is available at the AWS Architecture Center.

Extend Your Data Center into the Cloud

If you have invested in building your own data center, you may be facing challenges to keep up with constantly changing capacity requirements. Occasional spikes in demand may exceed your total capacity. If your enterprise is successful, even routine operations will eventually reach the capacity limits of your data center, and you'll have to decide how to extend that capacity. Building a new data center is one way, but it is expensive and slow, and the risk of underprovisioning or overprovisioning is high. In both of these cases, Amazon VPC can help you by serving as an extension of your own data center.

Amazon VPC allows you to specify your own IP address range so you can extend your network into AWS in much the same way you would extend an existing network into a new physical data center or branch office. VPN and AWS Direct Connect connectivity options allow these networks to be seamlessly and securely integrated to create a single corporate network capable of supporting your users and applications regardless of where they are physically located. And, just like a physical extension of a data center, IT resources hosted in VPC will be able to leverage existing centralized IT systems, like user authentication, monitoring, logging, change management, or deployment services, without the need to change how users or systems administrators access or manage your applications.

External connectivity from this extended, virtual data center is also completely up to you. You may choose to direct all VPC traffic to traverse your existing network infrastructure to control which existing internal and external networks your Amazon EC2 instances can access. This approach, for example, allows you to leverage all of your existing Internet-based network controls for your entire network. Figure 7 shows an example of a data center that has been extended into AWS.

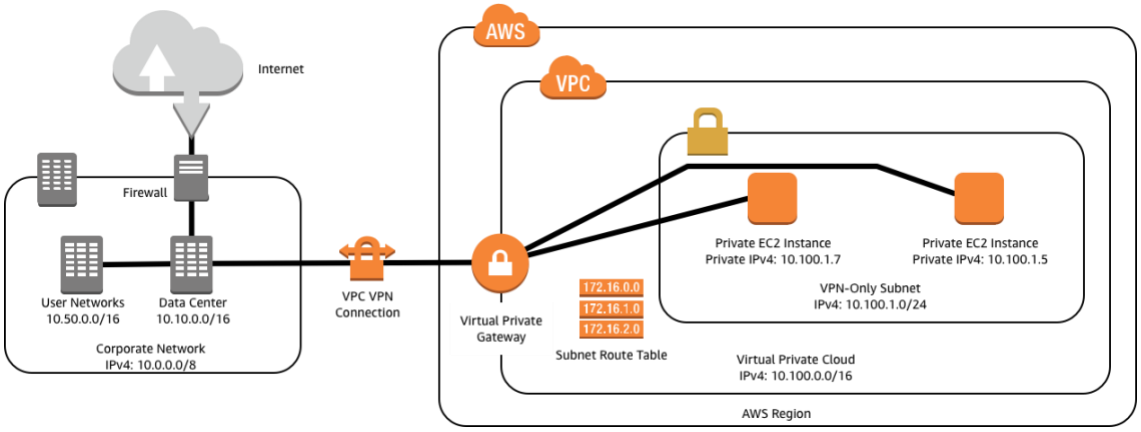


Figure 7: Example of a data center extended into AWS that leverages a customer's existing connection to the Internet

Additionally, you could also choose to leverage the extensive Internet connectivity of AWS to offload traffic from on-premises firewalls and load balancers and selectively present IPv6 endpoints even if your on-premises network only supports IPv4. You can deploy an AWS WAF to protect your infrastructure against attacks, leverage an application load balancer in your VPC to direct traffic to a mix of AWS based and on-premises resources using a VPN connection to provide a seamless end-user experience, as shown in Figure 8.

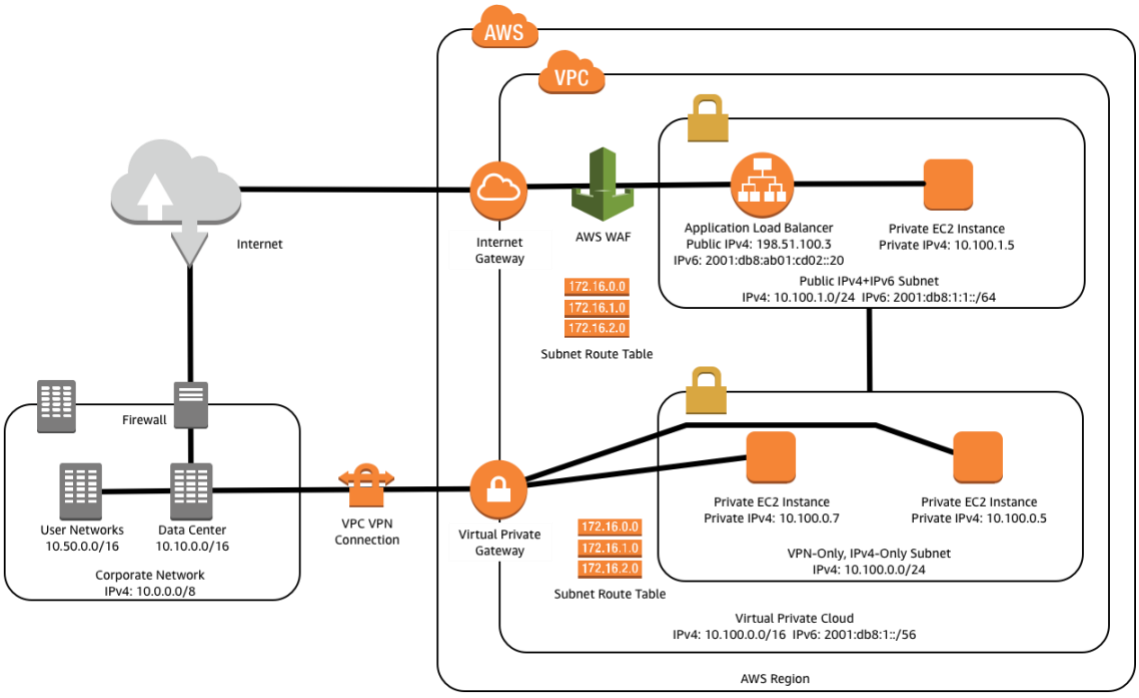


Figure 8: Example of a data center extended into AWS that leverages multiple connections to the Internet

Create Branch Office and Business Unit Networks

If you have branch offices that require separate but interconnected local networks, consider deploying separate VPCs for each branch office. Applications can easily communicate with each other using VPC peering, subject to VPC security group rules that you apply. The VPCs can even be in different AWS accounts and different regions, which can help reduce latency, enhance resource isolation, and enable cost allocation controls. If you need to limit network communication within or across subnets, you can configure security groups or network ACL rules to define which instances are permitted to communicate with each other. You could also use this same idea to group applications according to business unit functions. Applications specific to particular business units can be installed in separate VPCs, one for each unit. Figure 9 shows an example of using VPCs and VPNs for branch office scenarios.

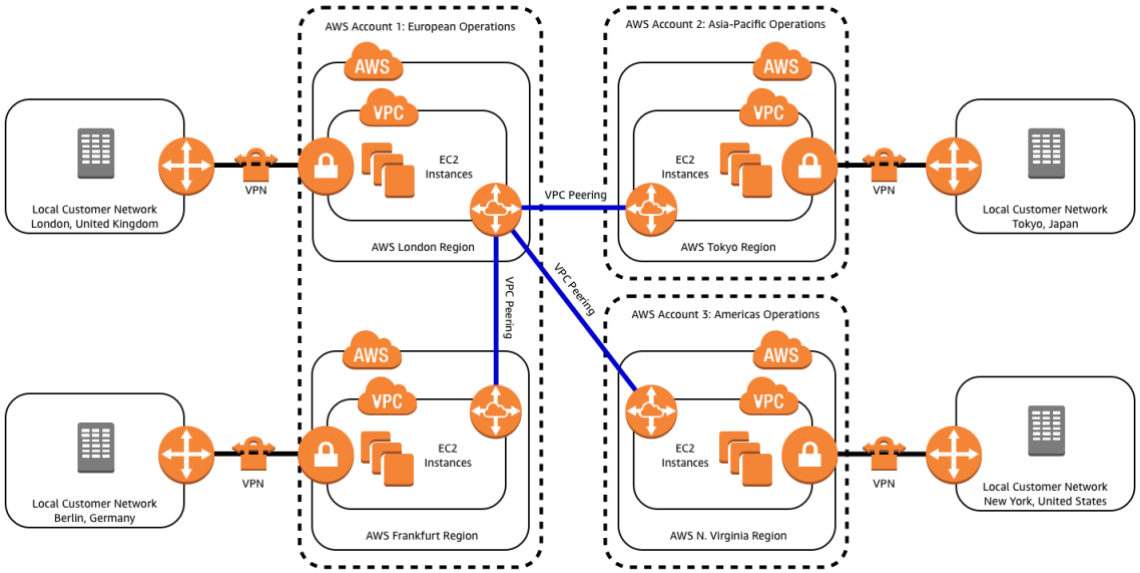


Figure 9: Example of using VPC and VPN for branch office scenarios

The main advantages of using Amazon VPC over provisioning dedicated on-premises hardware in a branch office are similar to those described elsewhere: you can elastically scale resources up, down, in, and out to meet demand, ensuring that you don't underprovision or overprovision. Adding capacity is easy: launch additional Amazon EC2 instances from your custom Amazon Machine Images (AMIs). When the time comes to decrease capacity, simply terminate the unneeded instances manually or automatically using Auto Scaling policies. Although the operational tasks may be the same to keep assets running properly, you won't need dedicated remote staff, and you'll save money with the AWS pay-as-you-use pricing model.

Best Practices for Using Amazon VPC

When using Amazon VPC, there are a few best practices you should follow:

- Automate the deployment of your infrastructure
- Use Multi-AZ deployments in VPC for high availability
- Use security groups and network ACLs
- Control access with IAM users and policies
- Use Amazon CloudWatch to monitor the health of your VPC instances and VPN link

Automate the Deployment of Your Infrastructure

Managing your infrastructure manually is tedious, error-prone, slow, and expensive. For example, in the case of a disaster recovery, your plan should include only a limited number of manual steps, because they slow down the process. Even in less critical use cases, such as development and test environments, we recommend that you ensure that your standby environment is an exact replica of the production environment. Manually replicating your production environment can be very challenging, and it increases the risk of introducing or not discovering bugs related to dependencies in your deployment.

By automating the deployment with AWS CloudFormation, you can describe your infrastructure in a declarative way by writing a template. You can use the template to deploy predefined stacks within a very short time in any AWS region. The template can fully automate creation of subnets, routing information, security groups, provisioning of AWS resources—whatever you need. By using AWS CloudFormation helper scripts, you can use standard Amazon Machine Images (AMIs) that will, upon startup of Amazon EC2 instances, install all of the software at the right version required for your deployment.

Automated infrastructure deployment should be fully integrated into your processes. You should treat your automation scripts like software that needs to be tested and maintained according to your standards and policies. A continuous deployment methodology, using services such as AWS CodePipeline to orchestrate the full process through build, test, and deploy phases, can help make infrastructure deployment a regular and well-tested business process. Thoroughly tested automated processes are often faster, cheaper, more reliable, and more secure than processes that rely on many manual steps.

Use Multi-AZ Deployments in VPC for High Availability

Architectures designed for high availability typically distribute AWS resources redundantly across multiple Availability Zones within the same region. If a service disruption occurs in one Availability Zone, you can redirect traffic to the other Availability Zone to limit the impact of the disruption. This general best practice also applies to architectures that include Amazon VPC.

Although a VPC can span multiple Availability Zones, each subnet within the VPC is restricted to a single Availability Zone. In order to deploy a multi-AZ Amazon Relational Database Service (Amazon RDS) instance, for example, you first have to configure VPC subnets in each Availability Zone within the region where the database instances will be launched. Likewise, Auto Scaling groups and elastic load balancers can span multiple Availability Zones by being deployed across VPC subnets that have been created for each zone.

Use Security Groups and Network ACLs

Amazon VPC security groups allow you to control both ingress and egress traffic, and you can define rules for all IP protocols and ports. For a full overview of the features available with Amazon VPC security groups, see Security Groups for Your VPC. Amazon VPC security groups are stateful firewalls, allowing return traffic for permitted TCP connections.

A network access control list (ACL) is an additional layer of security that acts as a firewall to control traffic into and out of a subnet. You can define access control rules for each of your subnets. Although a VPC security group operates at the instance level, a network ACL operates at the subnet level. For a network ACL, you can specify both allow and deny rules for both ingress and egress. Network ACLs are stateless firewalls; return traffic for TCP connections must be explicitly allowed on the TCP ephemeral ports (typically 32768-65535).

As a best practice, you should secure your infrastructure with multiple layers of defense. By running your infrastructure in a VPC, you can control which instances are exposed to the Internet in the first place, and you can define both security groups and network ACLs to further protect your infrastructure at the infrastructure and subnet levels. Additionally, you should secure your instances with a firewall at the operating system level and follow other security best practices as outlined in AWS Security Resources.

Control Access with IAM Users and Policies

With AWS Identity and Access Management (IAM), you can create and manage users in your AWS account. A user can be either a person or an application that needs to interact with AWS. With IAM, you can centrally manage your users, their security credentials, such as access credentials, and permissions that control which AWS

resources the users can access. You typically create IAM users for users and use IAM roles for applications.

We recommend that you use IAM to implement a least privilege security strategy. For example, you should not use a single AWS IAM user to manage all aspects of your AWS infrastructure. Instead, we recommend that you define user groups (or roles, if using federated logins) for the different tasks that have to be performed on AWS and restrict each user to exactly the functionality he or she requires to perform that role. For example, you can create a network admin group of users in IAM and then give only that group the rights to create and modify the VPC. For each user group, define restrictive policies that grant each user access only to those services he or she needs. Make sure that only authorized people in your organization have access to these users. Use services such as Amazon GuardDuty to detect anomalous access patterns. Implement strong authentication requirements such as minimum password length and complexity, and consider multifactor authentication to reduce the risk of compromising your infrastructure.

For more information on how to define IAM users and policies, see Controlling Access to Amazon VPC Resources.

Use Amazon CloudWatch to Monitor the Health of Your VPC Instances and VPN Link

Just as you do with public Amazon EC2 instances, you can use Amazon CloudWatch to monitor the performance of the instances running inside your VPC. Amazon CloudWatch provides visibility into resource utilization, operational performance, and overall demand patterns, including CPU utilization, disk reads and writes, and network traffic. The information is displayed on the AWS Management Console and is also available through the Amazon CloudWatch API so you can integrate into your existing management tools.

You can also view the status of your VPN connections by using either the AWS Management Console or making an API call. The status of each VPN tunnel will include the state (up/down) of each VPN tunnel and the amount of traffic seen across the VPN tunnels.

Conclusion

Amazon VPC offers a wide range of tools that give you more control over your AWS infrastructure. Within a VPC, you can define your own network topology by defining subnets and routing tables, and you can restrict access at the subnet level with network ACLs and at the resource level with VPC security groups. You can isolate your resources from the Internet and connect them to your own data center through a VPN. You can assign elastic IPv4 and public IPv6 addresses to some instances and connect them to the public Internet through an Internet gateway, while keeping the rest of your infrastructure in private subnets. Amazon VPC makes it easier to protect your AWS resources while you keep the benefits of AWS with regard to flexibility, scalability, elasticity, performance, availability, and the pay-as-you-use pricing model.

Further Reading

- Amazon VPC product page: https://aws.amazon.com/vpc/
- Amazon VPC documentation: https://aws.amazon.com/documentation/vpc/
- AWS Direct Connect product page: https://aws.amazon.com/directconnect/
- Getting started with AWS Direct Connect: https://aws.amazon.com/directconnect/getting-started/
- AWS Security Center: https://aws.amazon.com/security/
- Amazon VPC Connectivity Options: https://media.amazonwebservices.com/AWS_Amazon_VPC_Connectivity_Options.pdf
- AWS VPN CloudHub: https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPN_CloudHub.html
- AWS Security Best Practices: https://aws.amazon.com/whitepapers/aws-security-best-practices/
- Architecting for the Cloud: Best Practices: http://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf

Document Revisions

Date	Description
December 2018	Added IPv6 features. Removed references to EC2 classic. Added AWS DMS, AWS CodePipeline, Amazon GuardDuty Changed multiple-subnet strategy to multiple-VPC, VPC peering, CloudHub. Removed recommendation to change credentials regularly (no longer NIST recommended); added complexity and MFA.
December 2013	Major revision to reflect new functionality of Amazon VPC Added new use cases for Amazon VPC Added section “Understanding Amazon Virtual Private Cloud”. Added section “Best Practices for Using Amazon VPC”
January 2010	First publication