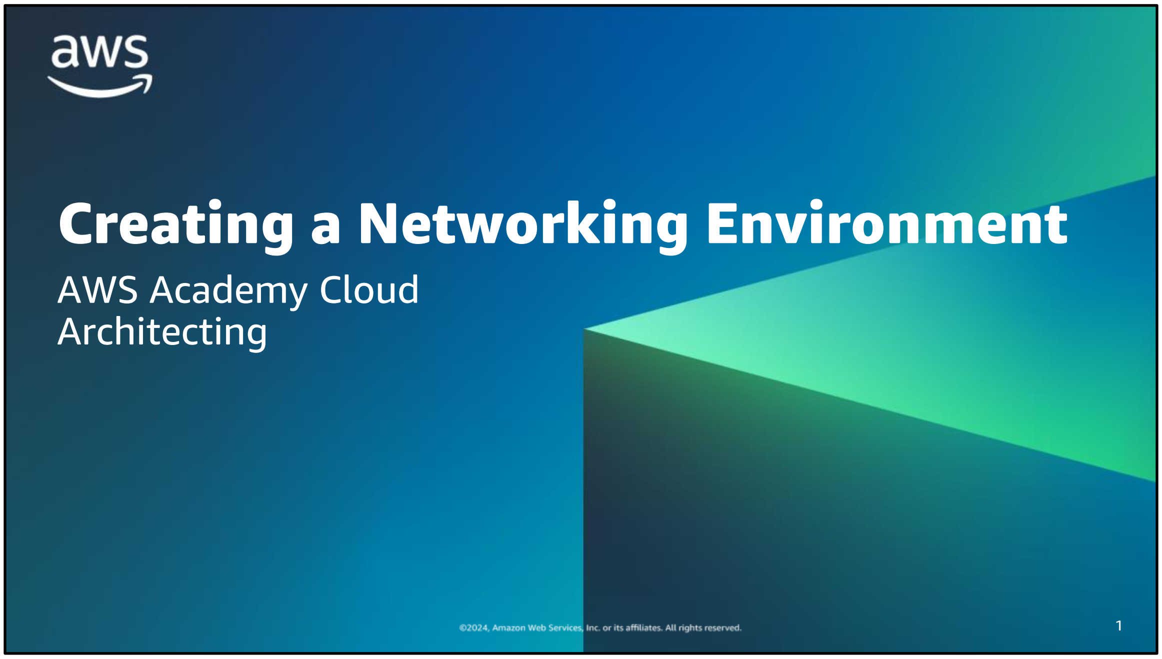
AWS Academy Cloud Architecting
Module 07 Student Guide
Version 3.0.0

200-ACACAD-30-EN-SG

Contents

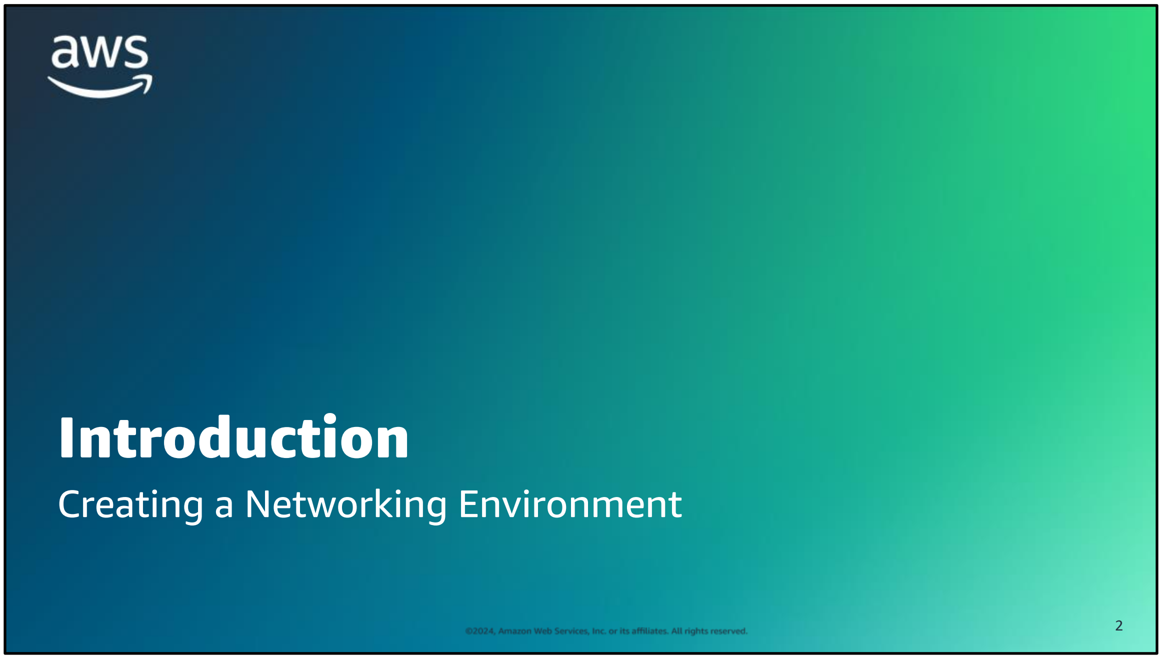


Creating a Networking Environment

AWS Academy Cloud
Architecting

1

Welcome to the Creating a Networking Environment module. This module describes the customer usage of the AWS networking environment with the Amazon Virtual Private Cloud (VPC) service.

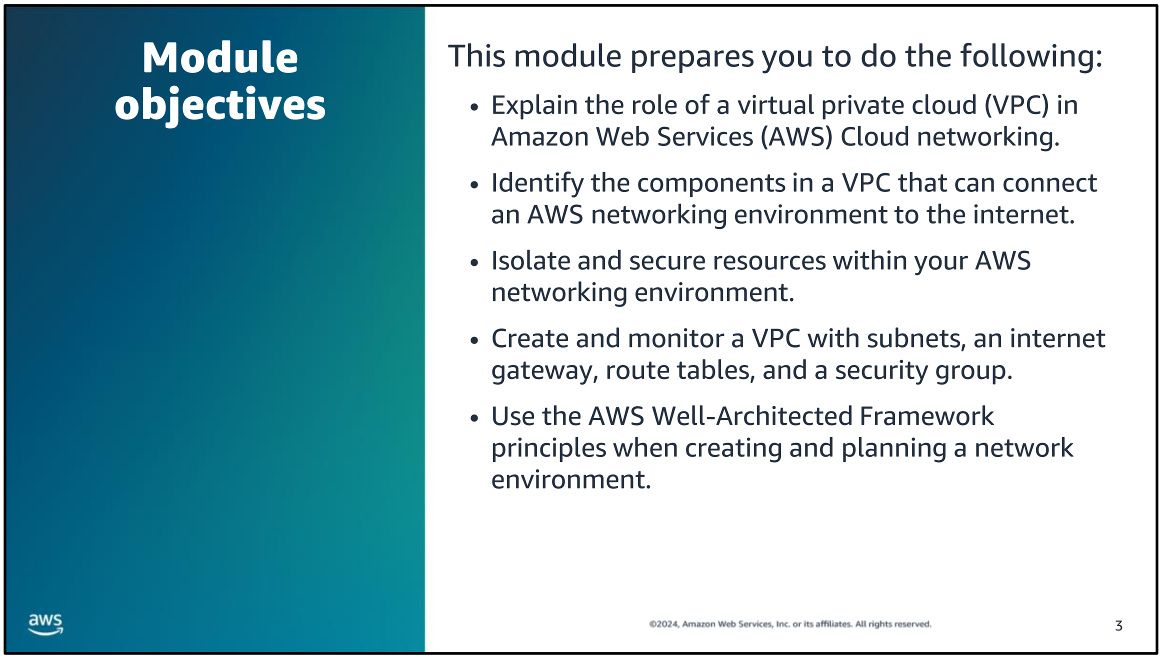


Introduction

Creating a Networking Environment

2

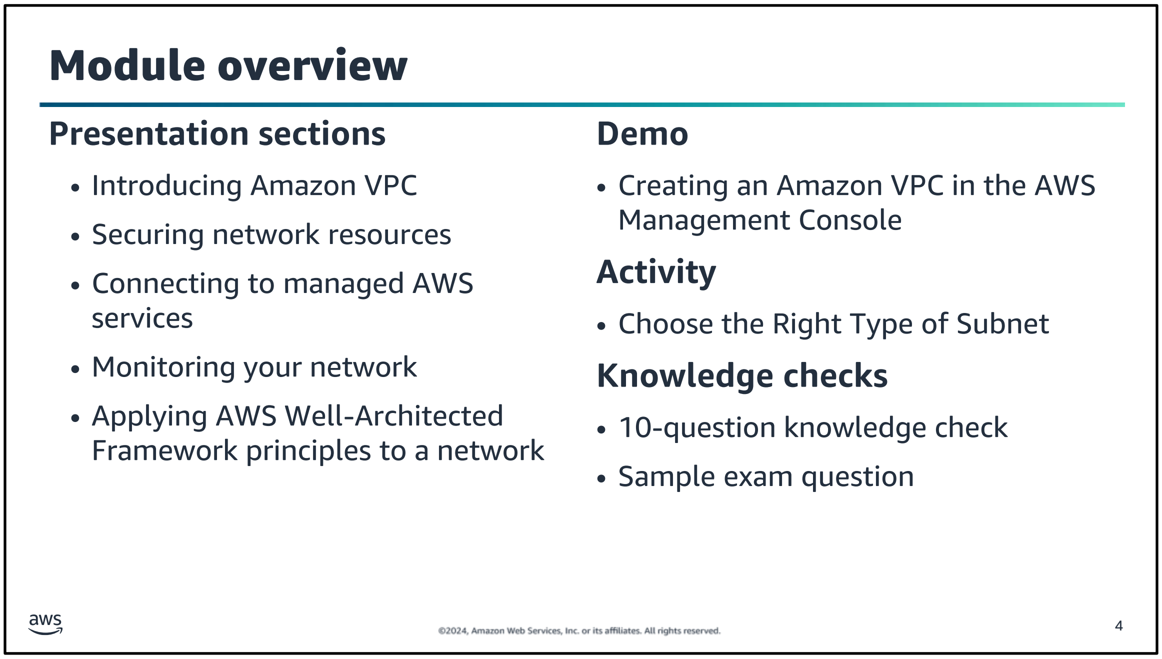
This introduction section describes the content of this module.



Module objectives

This module prepares you to do the following:

- Explain the role of a virtual private cloud (VPC) in Amazon Web Services (AWS) Cloud networking.
- Identify the components in a VPC that can connect an AWS networking environment to the internet.
- Isolate and secure resources within your AWS networking environment.
- Create and monitor a VPC with subnets, an internet gateway, route tables, and a security group.
- Use the AWS Well-Architected Framework principles when creating and planning a network environment.



Module overview

Presentation sections

- Introducing Amazon VPC
- Securing network resources
- Connecting to managed AWS services
- Monitoring your network
- Applying AWS Well-Architected Framework principles to a network

Demo

- Creating an Amazon VPC in the AWS Management Console

Activity

- Choose the Right Type of Subnet

Knowledge checks

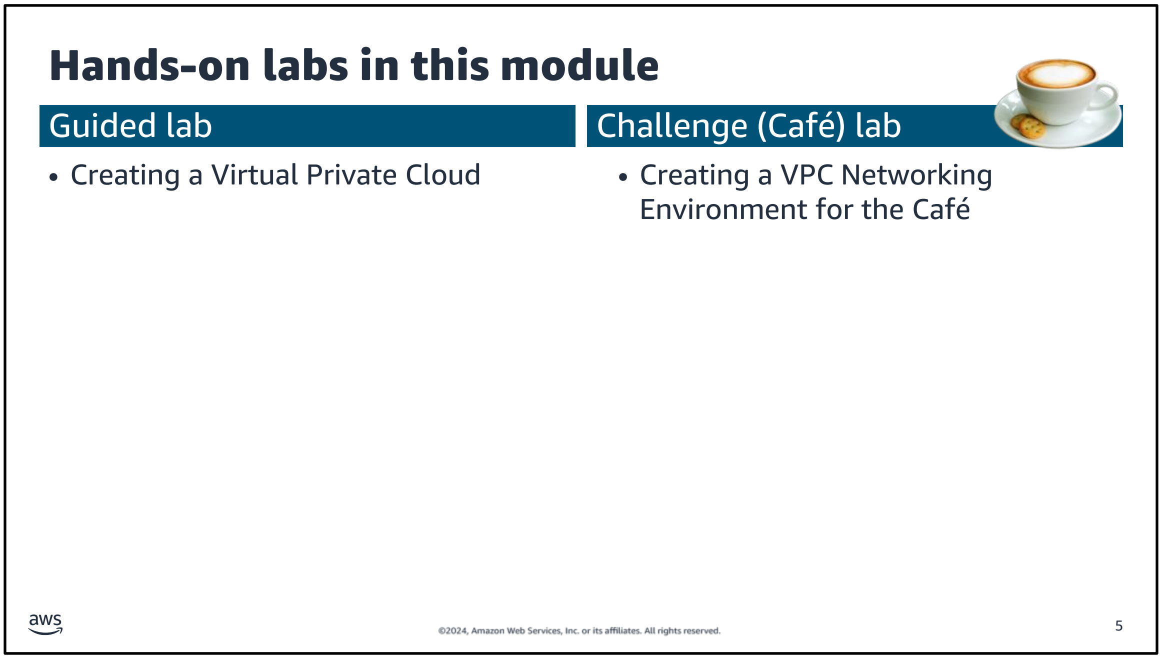
- 10-question knowledge check
- Sample exam question

4

The objectives of this module are presented across multiple sections.

You will also view a demonstration that shows how to create a VPC in the AWS Management Console and work on an activity to choose the right type of subnet for a given use case.

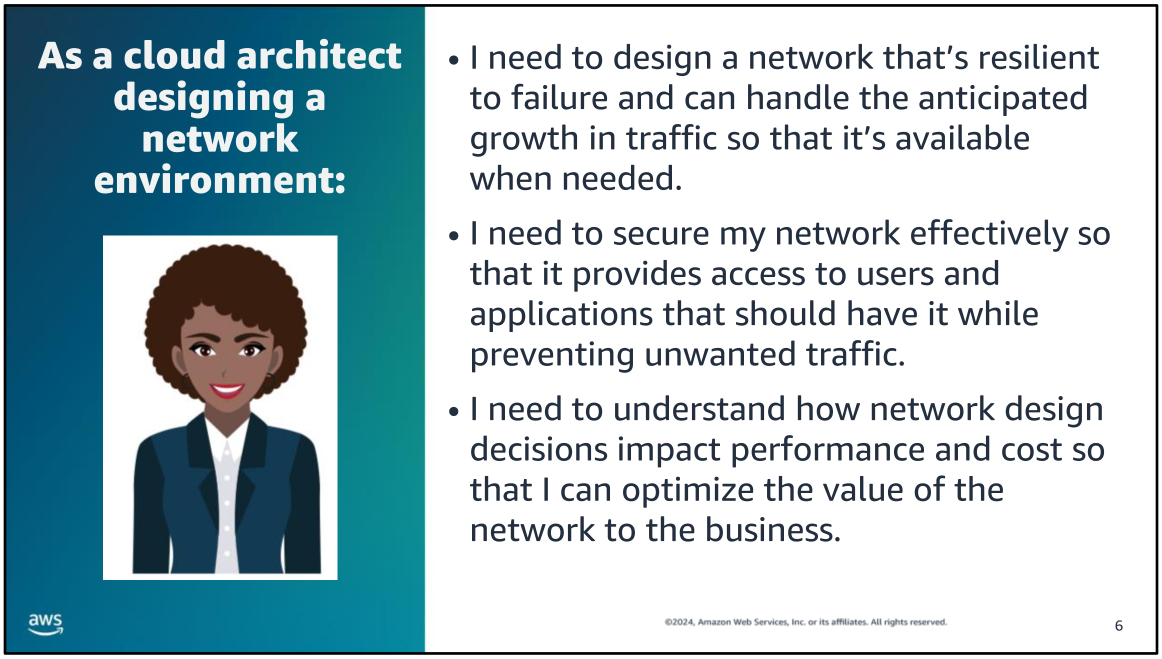
The module wraps up with a 10-question knowledge check delivered in the online course and a sample exam question to discuss in class.
The labs in this module are described on the next slide.



Hands-on labs in this module

Guided lab	Challenge (Café) lab
• Creating a Virtual Private Cloud	• Creating a VPC Networking Environment for the Café

This module includes the hands-on labs that are listed. There’s a guided lab where you are provided step-by-step instructions and a café lab where you work on updating the architecture for the café. Additional information about each lab is included in the student guide where the lab takes place, and detailed instructions are provided in the lab environment.



As a cloud architect designing a network environment:

- I need to design a network that's resilient to failure and can handle the anticipated growth in traffic so that it's available when needed.
- I need to secure my network effectively so that it provides access to users and applications that should have it while preventing unwanted traffic.
- I need to understand how network design decisions impact performance and cost so that I can optimize the value of the network to the business.

6

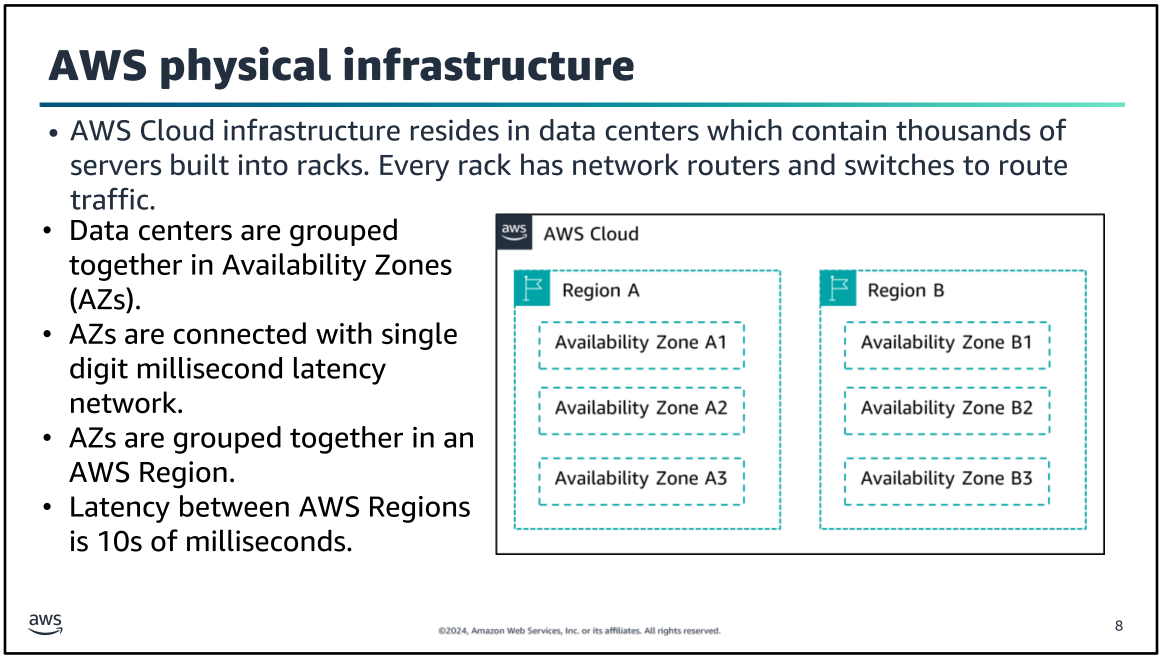
This slide asks you to take the perspective of a cloud architect as you think about how to approach cloud network design. Keep these considerations in mind as you progress through this module, remembering that the cloud architect should work backwards from the business need to design the best architecture for a specific use case. As you progress through the module, consider the café scenario presented in the course as an example business need and think about how you would address these needs for the fictional café business.

Introducing Amazon VPC

Creating a Networking Environment

7

This section introduces the Amazon Virtual Private Cloud (VPC) service and its components.



AWS physical infrastructure

- AWS Cloud infrastructure resides in data centers which contain thousands of servers built into racks. Every rack has network routers and switches to route traffic.
- Data centers are grouped together in Availability Zones (AZs).
- AZs are connected with single digit millisecond latency network.
- AZs are grouped together in an AWS Region.
- Latency between AWS Regions is 10s of milliseconds.

AWS Cloud

Region A

Availability Zone A1

Availability Zone A2

Availability Zone A3

Region B

Availability Zone B1

Availability Zone B2

Availability Zone B3

8

When you design your network on AWS, you need to understand the AWS physical infrastructure components and hierarchies. The AWS physical infrastructure is made up of multiple AWS Regions. Each AWS Region is a separate geographic area located in a country.

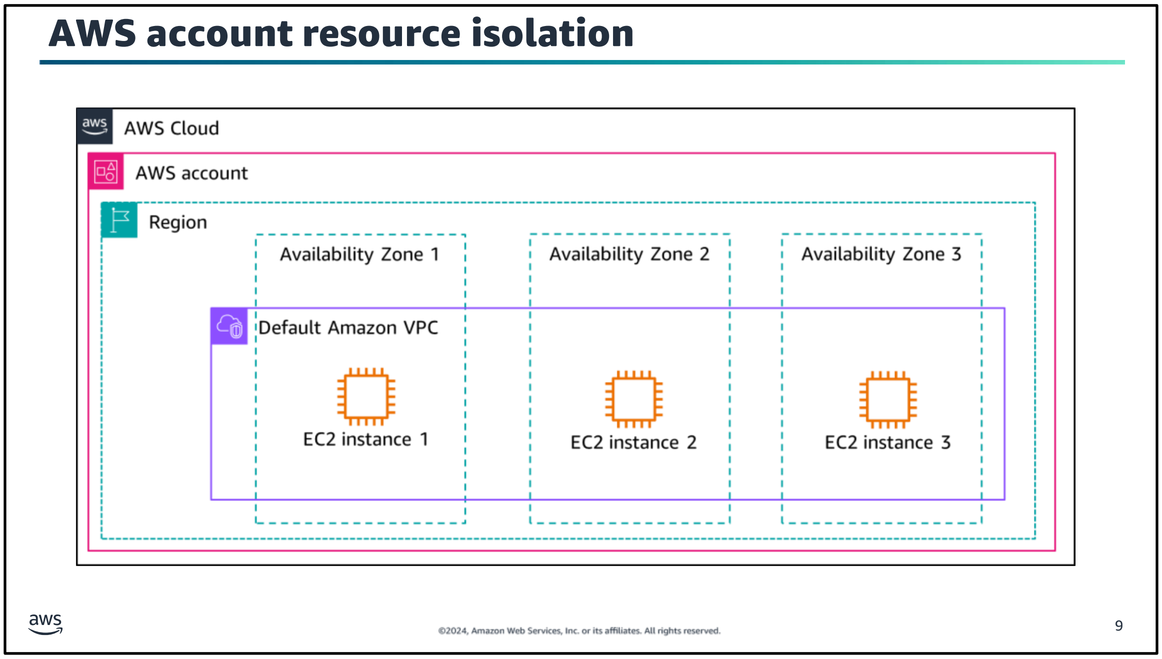
An AWS Region is made up of at least three isolated and physically separate locations called Availability Zones (AZs). The code for an AZ is its Region code followed by a letter identifier. For example, one of the AZs in the Northern Virginia Region **us-east-1** in the US is called **us-east-1a**.

An AZ is made up of one or more data centers with redundant power, networking, and connectivity in an AWS Region. AZs are connected with a wired, physical low latency network that allows synchronous data replication between AZs. Inside a data center, there are thousands of servers and hosts built into racks. Physical network devices, such as hardware routers, switches, firewalls, and load balancers, are connected to each rack. Each device in the network has an IP address that routers and switches can use to reach the device.

A virtual network (sometimes called an overlay network) is created on top of a physical network. To deploy a resource in AWS, you select either a Region or an AZ depending on the service that you use. Data centers are not logically identifiable.

A virtual network emulates a physical network with software defined network components such as switches, routers, firewalls, and load balancers. These components are created and managed programmatically. This means that all devices on the virtual network are logically isolated through software definitions.

See course source reference list for more information about AWS global infrastructure.



AWS account resource isolation

9

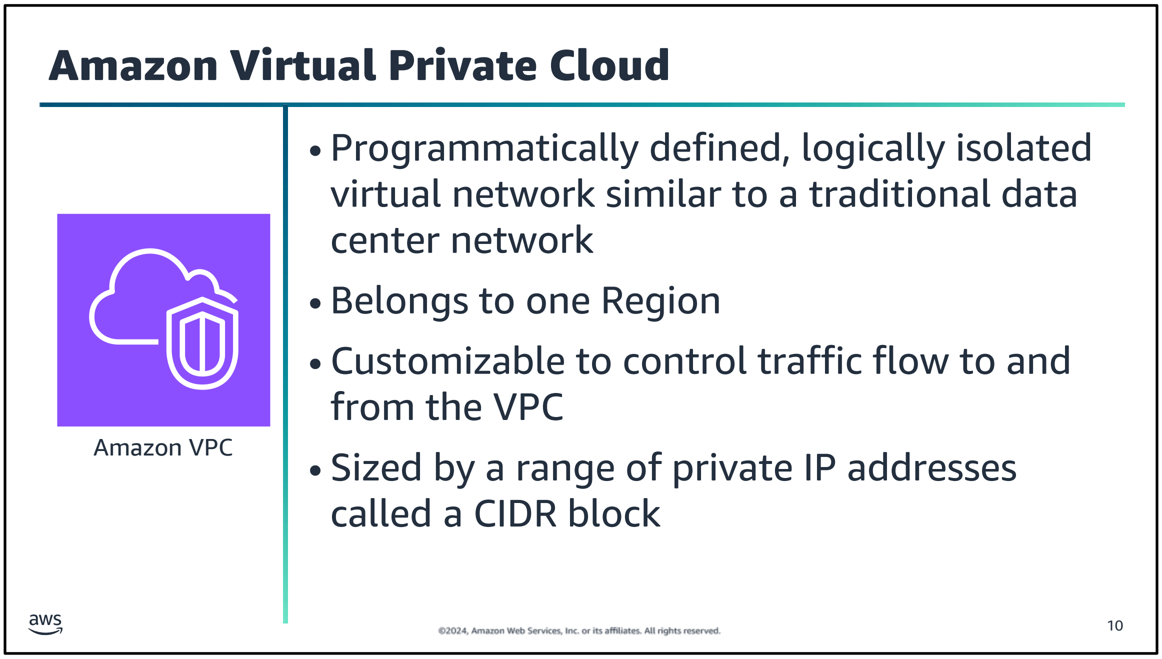
Now that you understand the physical AWS network supporting the virtual network, you can start to build your network programmatically.

When you request a new AWS account, network resources are dedicated to the account. An AWS account spans across AWS Regions. It contains a default virtual, software defined network in each public accessible Region. This is called an Amazon virtual private cloud (Amazon VPC). A VPC belongs to one Region and can span multiple AZs.

When you create a new VPC, it's logically isolated from other virtual networks in the AWS cloud and the internet. This means that it can't be accessed unless you add configuration to allow access.

AWS services such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) are designed to operate inside a VPC. These resources can be connected to AWS serverless services, such as AWS Lambda and Amazon CloudWatch, which operate outside of a customer VPC.

Because the default VPC already has connectivity configurations built in, you shouldn't use the default VPC for production workloads. Instead, define a new VPC with appropriate production workload security and connectivity settings.



Amazon Virtual Private Cloud

Amazon VPC

- Programmatically defined, logically isolated virtual network similar to a traditional data center network
- Belongs to one Region
- Customizable to control traffic flow to and from the VPC
- Sized by a range of private IP addresses called a CIDR block

10

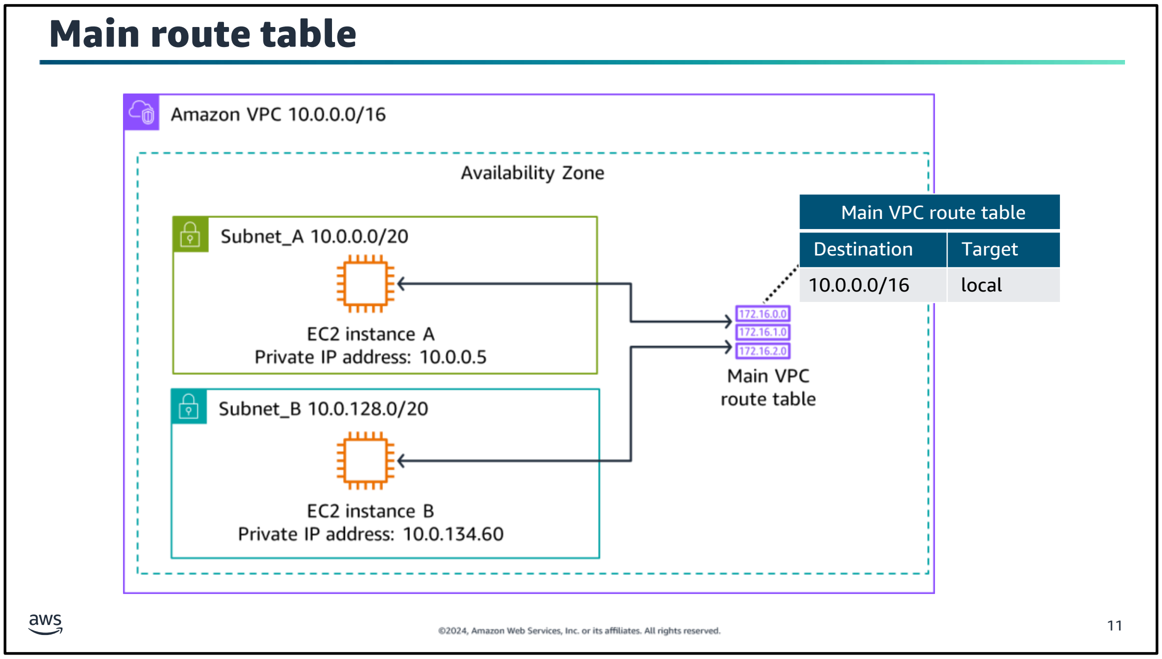
When you want to define your AWS Cloud virtual network in an AWS Region, you will use the Amazon VPC service. You can programmatically define how your network is exposed to the internet and corporate environments.

To size your VPC, you allocate a range of private IP addresses to your VPC. The range of IP addresses determines the size of the VPC. A range of IP addresses is called a Classless Inter-Domain Routing (CIDR) block. It’s important to allocate an adequate number of IP addresses for your VPC because it’s time consuming and difficult to add more IP addresses later on. The maximum IPv4 VPC size is a /16 netmask with 65,536 IP addresses and the smallest a /28 netmask of 16 IP addresses. For more information about CIDR blocks and netmasks, see “What are CIDR blocks” in the Content Resources page of your course.

When you create a VPC, you can specify it's IP addresses as IPv4 only or a dual stack. A dual stack VPC has both an IPv4 CIDR block and an IPv6 CIDR block. To plan, track, and monitor your IP addresses efficiently, you can use a VPC feature Amazon VPC IP Address Manager (IPAM).

A good reason to opt for IPv6 is that IPv6 has a large address space with each IP address sized at 128 bits. IPv4 IP addresses are only 32 bit. IPv6 also generally has better speed performance than IPv4 because it does not do NAT. Please note that effective early 2024 there will be a per hour cost for all public IPv4 addresses, whether attached to a service or not.

If you deploy your VPC over multiple AZs, the number of IP addresses should be spread evenly across each AZ.



Main route table

Main VPC route table	
Destination	Target
10.0.0.0/16	local

After you create the VPC and make sure it has enough IP addresses, the next step is to decide how isolated resources should be in the VPC. Should a resource be reachable from the internet, corporate environment, other VPCs, or not?

A VPC automatically comes with a router which uses a VPC main route table containing a set of routing rules, called routes. In the diagram, to route IPv4 traffic inside the VPC, the main route table has a default route rule with a destination of 10.0.0.0/16 and a target of local. The destination of 10.0.0.0/16 is the CIDR block of the VPC. This means every resource in the VPC is reachable by every other resource in the VPC. In the diagram, EC2 instance A can access EC2 instance B through the main VPC route table. This default route can’t be deleted.

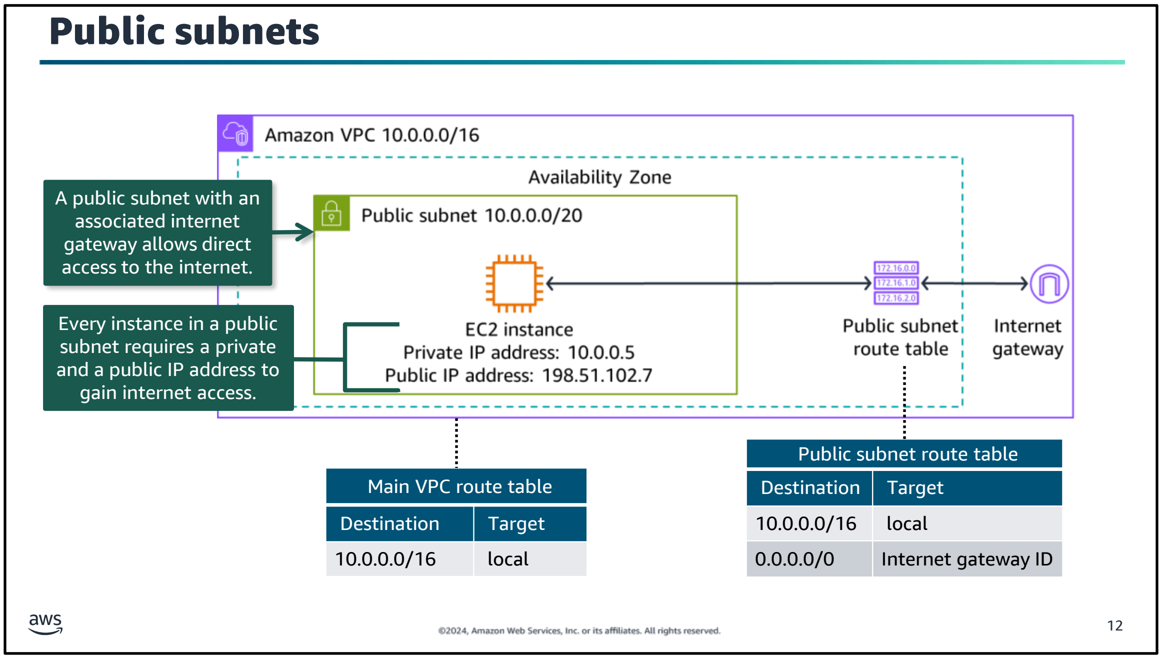
A VPC can be divided into subnets. A subnet is a container for routing policies and belongs to one AZ. Each subnet is a segment of the range of IP addresses of the VPC and should be significantly smaller than the VPC. Subnet CIDR blocks can’t overlap.

The subnets in the example has a netmask of /20 allowing for 4096 IP addresses.

AWS reserves the first four IP addresses and the last IP address in each subnet CIDR block. For example, in a subnet with CIDR block 10.0.0.0/20, AWS reserves the following five IP addresses for:

- 10.0.0.0: Network address
- 10.0.0.1: VPC local router
- 10.0.0.2: DNS resolution
- 10.0.0.3: Future use
- 10.0.15.254: Network broadcast address

It’s important to allocate enough IP addresses in a subnet.



Public subnets

Amazon VPC 10.0.0.0/16

Availability Zone

Public subnet 10.0.0.0/20

A public subnet with an associated internet gateway allows direct access to the internet.

Every instance in a public subnet requires a private and a public IP address to gain internet access.

EC2 instance
Private IP address: 10.0.0.5
Public IP address: 198.51.102.7

Public subnet route table

Internet gateway

Main VPC route table

Destination	Target
10.0.0.0/16	local

Public subnet route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	Internet gateway ID

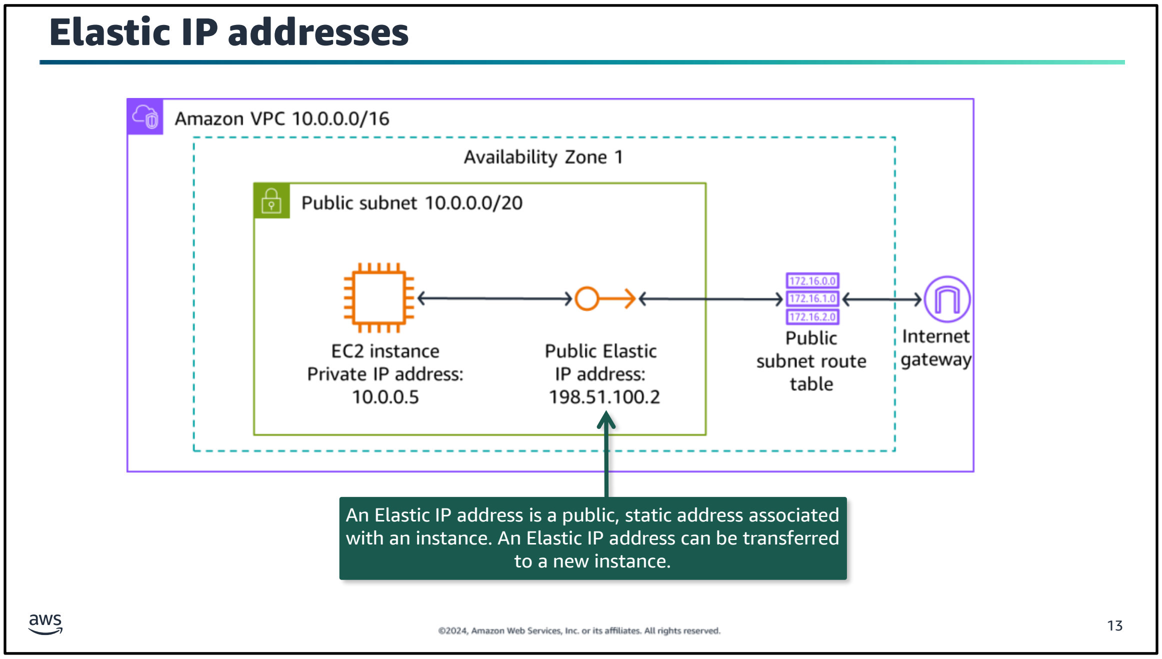
12

To isolate resources that share the same routing requirements, you can place them in public or private subnets. For resources that should be reached from the internet, create a subnet and an internet gateway.

An internet gateway is a VPC component that allows communication between resources in your VPC and the internet. It’s horizontally scaled, redundant, and highly available. An internet gateway supports IPv4 and IPv6 traffic. An internet gateway serves two purposes. First, it provides a target in your VPC route tables for internet-routable traffic. Second, the internet gateway performs NAT for instances that were assigned public IPv4 addresses.

To make your public subnet reachable from the internet, create an internet gateway and attach the internet gateway to the VPC. The internet gateway is a Regional resource and can only be attached to one VPC. Create an EC2 instance with a public IP address in the public subnet. To send traffic through the internet gateway to the internet from the subnet, create a public subnet route table. Add a route to the public subnet route table with destination 0.0.0.0/0 to indicate all traffic and a target with the internet gateway ID. The main VPC route table is overridden with the routes of the public subnet route table.

In the diagram, note that the EC2 instance has to have a public IP address to allow reachability from the internet. Public IP addresses are not associated with an AWS account and are released back to AWS when not in use.



Elastic IP addresses

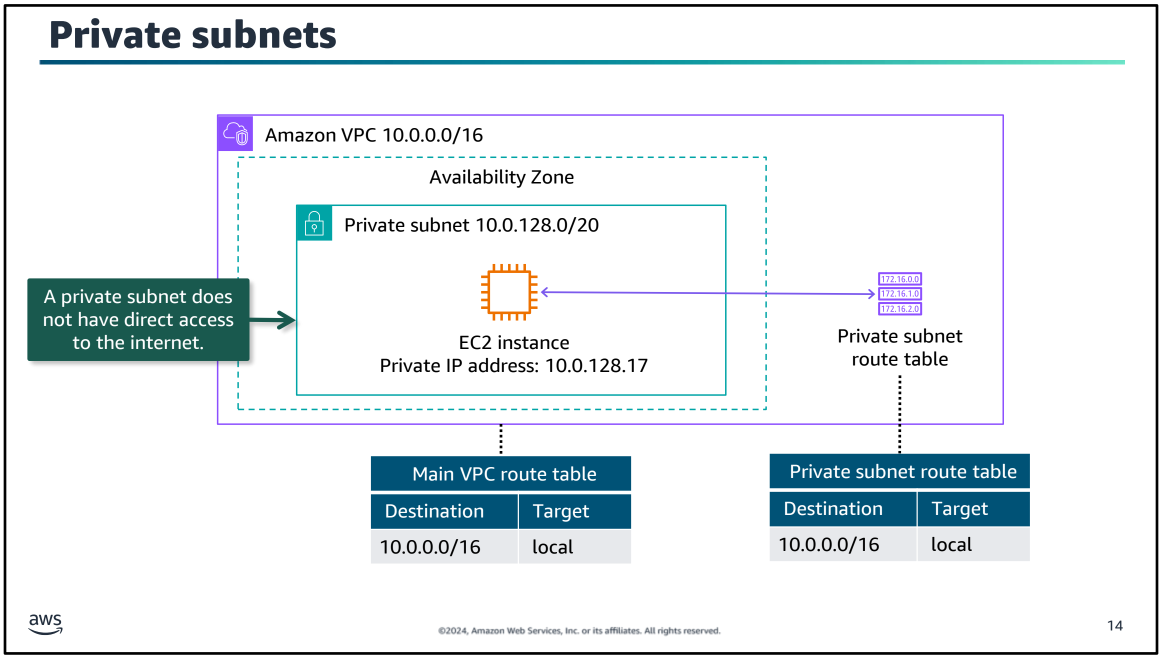
An Elastic IP address is a public, static address associated with an instance. An Elastic IP address can be transferred to a new instance.

13

An Elastic IP address is a static and public IPv4 address that you can associate with an EC2 instance. If the EC2 instance's health deteriorates, you can transfer the Elastic IP address to a new, healthy EC2 instance. An EC2 instance private IP address is dynamic as it is released when the instance is terminated.

There's no cost for the first Elastic IP address that's associated with a running EC2 instance. You will incur cost for any additional Elastic IP addresses associated with the instance or the instance is stopped. If you detach the Elastic IP address from the instance and don't associate it with another instance, you will incur an hourly cost.

Elastic IP addresses can also be associated with a load balancer or a VPC network interface.



Private subnets

A private subnet does not have direct access to the internet.

Main VPC route table	
Destination	Target
10.0.0.0/16	local

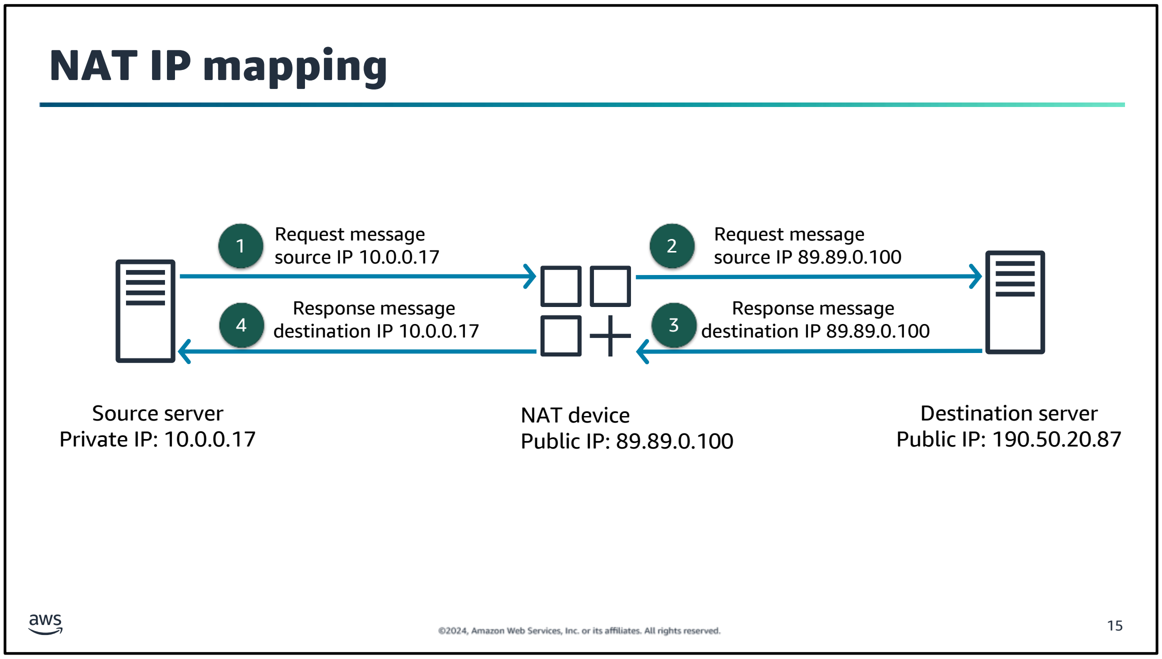
Private subnet route table	
Destination	Target
10.0.0.0/16	local

14

For resources that shouldn’t be accessible from the internet, you can define a VPC private subnet. All resources in the private subnet aren’t reachable from the internet and don’t have direct access to the internet.

It’s a best practice to define a custom route table for every subnet containing its own routes. In the diagram, the private subnet route table is identical to the main VPC route table. The EC2 instance can reach any resource in the VPC.

When a subnet doesn’t have an explicit routing table associated with it, the main VPC route table is used by default. A subnet can only be associated with one route table at a time, but you can associate multiple subnets in a subnet route table.



NAT IP mapping

1 Request message source IP 10.0.0.17

2 Request message source IP 89.89.0.100

3 Response message destination IP 89.89.0.100

4 Response message destination IP 10.0.0.17

Source server
Private IP: 10.0.0.17

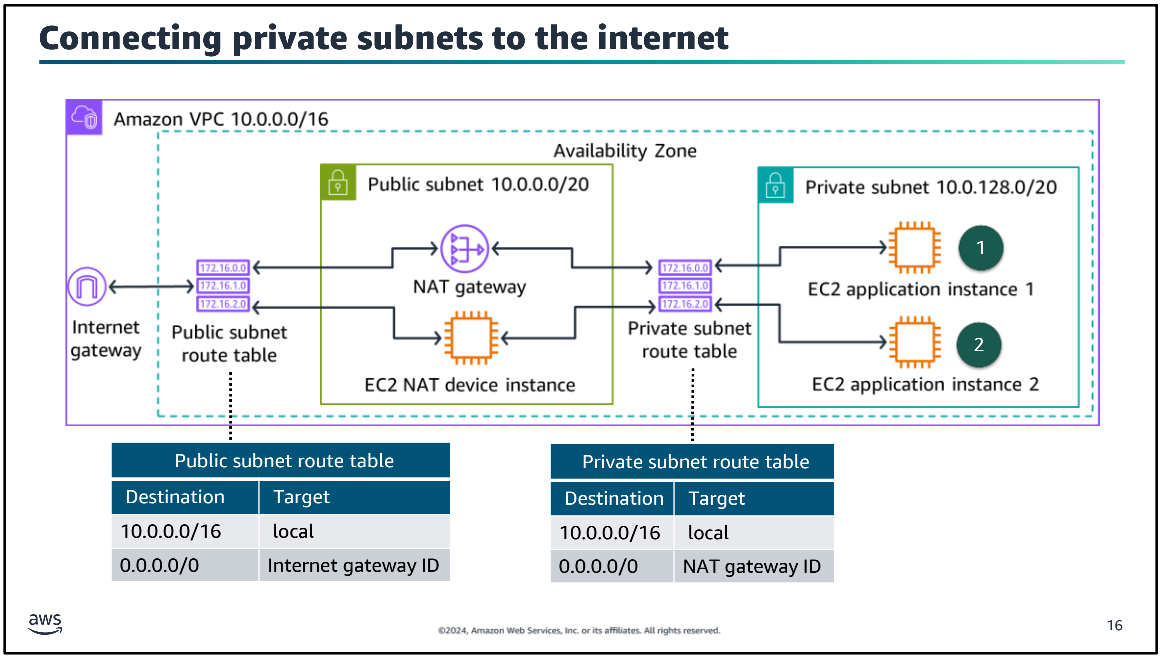
NAT device
Public IP: 89.89.0.100

Destination server
Public IP: 190.50.20.87

When you don’t want to expose a server IP address to other networks or the internet, it’s useful to use a NAT device. The NAT device replaces the private IP of the request initiator with it’s own public IP.

In the example on the slide, the following sequence of events take place:

1. The source server initiates a request message to the NAT device with it’s own private IP of 10.0.0.17 as the message source IP.
2. The NAT device maps the message source IP to it’s own public IP of 89.89.0.100 and forwards the message to the destination server.
3. The destination server responds with the destination IP set as 89.89.0.100 to the NAT device.
4. The NAT device maps the response message destination IP to the source server’s private IP of 10.0.0.17 and forwards the message to the source server.



Connecting private subnets to the internet

Amazon VPC 10.0.0.0/16
Availability Zone
Public subnet 10.0.0.0/20
Private subnet 10.0.128.0/20
Internet gateway
172.16.0.0
172.16.1.0
172.16.2.0
Public subnet route table
NAT gateway
EC2 NAT device instance
Private subnet route table
1
EC2 application instance 1
2
EC2 application instance 2

Public subnet route table	
Destination	**Target**
10.0.0.0/16	local
0.0.0.0/0	Internet gateway ID

Private subnet route table	
Destination	**Target**
10.0.0.0/16	local
0.0.0.0/0	NAT gateway ID

There are scenarios where there’s a requirement to have resources in a private subnet connect to the internet. The resources must stay private and not discoverable by the internet. For example, requesting patches from the internet to download on an EC2 instance.

You can use a NAT device to allow resources in private subnets to connect to the internet, other VPCs, or on-premises networks. These instances can communicate with services outside the VPC, but they can’t receive unsolicited connection requests.

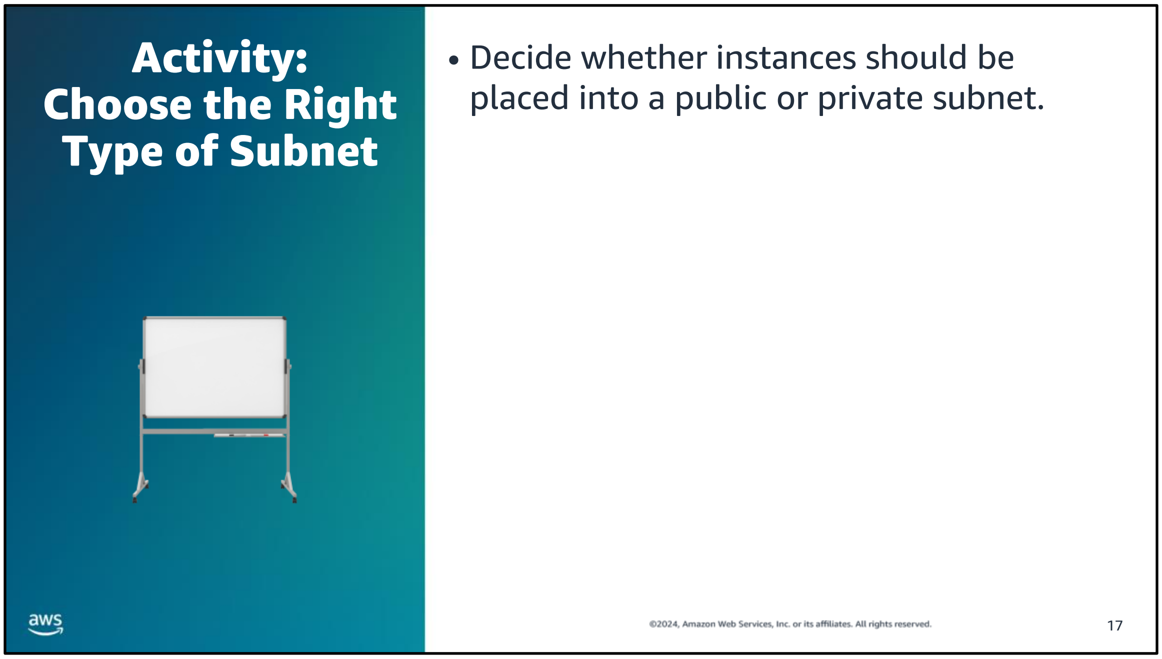
AWS offers two types of NAT device solutions. You can use an AWS managed NAT device offered by AWS, called a NAT gateway, or you can create your own NAT device on an EC2 instance, called a NAT instance. Because a NAT gateway is a managed AWS service, it incurs an hourly cost. A NAT device instance incurs EC2 costs.

1. In the image above, EC2 application instance 1 initiated a request to the NAT gateway using the private subnet route table. The NAT gateway replaces the source IPv4 address of the instance with the address of the NAT gateway. The NAT gateway sends the request to the internet gateway and receives the response. When sending the response back to the instance, the NAT gateway translates the IP address back to the private EC2 application instance 1 IPv4 address.

2. Similarly, EC2 application instance 2 initiated a request to the NAT instance using the private subnet route table. The NAT instance replaces the source IPv4 address of the instance with the address of the NAT instance. The NAT gateway sends the request to the internet gateway and receives the response. When sending response traffic to back the instance, the NAT instance translates the addresses back to the private EC2 application instance 2 IPv4 address.

AWS recommends that you use NAT gateways over NAT instances because they provide better availability and bandwidth and require less effort on your part to administer. For resilience when using multiple AZs, deploy a NAT gateway in each AZ.

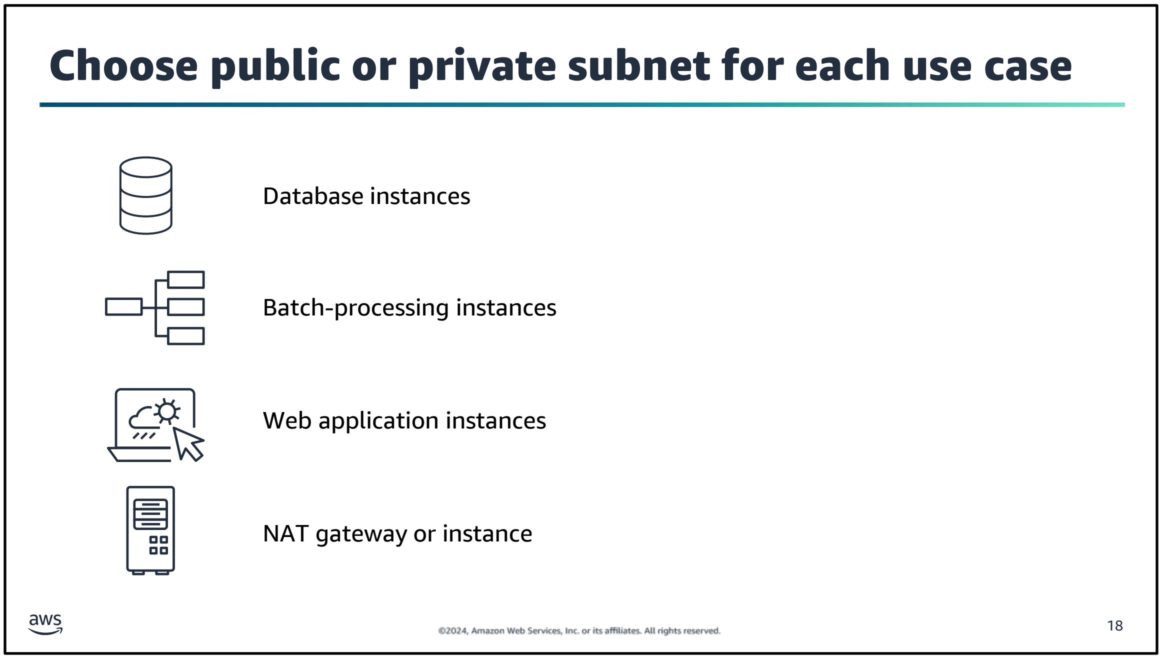
For IPv6 private subnets, use an egress-only internet gateway to allow outbound communication over IPv6 to the internet and prevent the internet from initiating an IPv6 connection with your instances.



Activity: Choose the Right Type of Subnet

- Decide whether instances should be placed into a public or private subnet.

The next slide provides examples for your review.

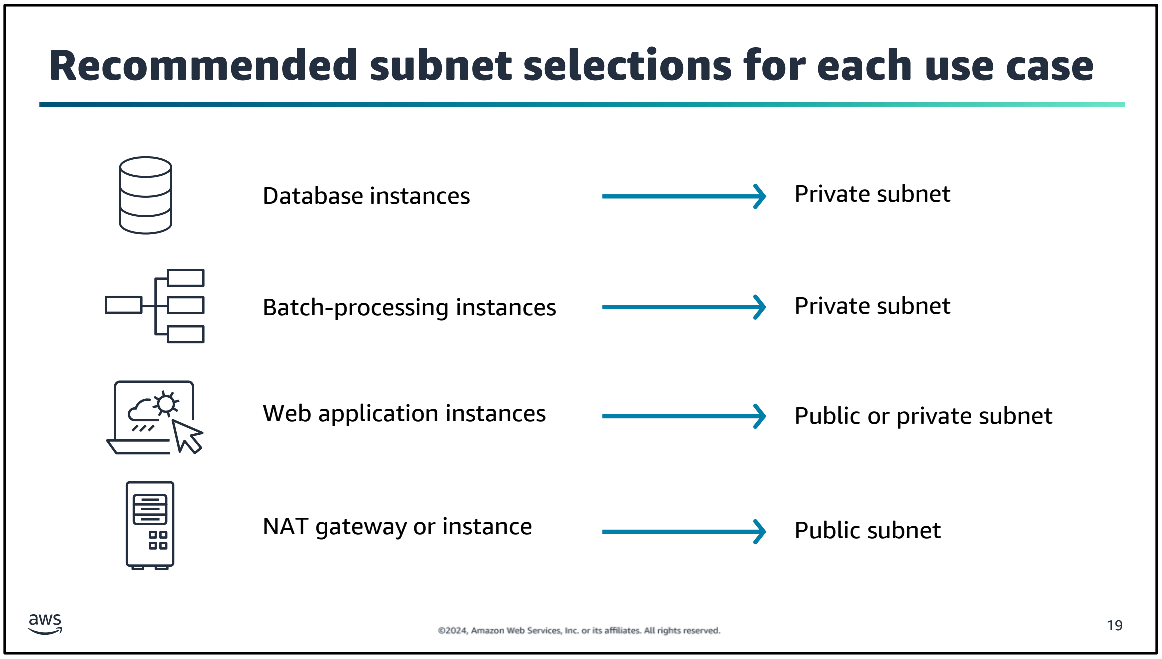


Choose public or private subnet for each use case

- Database instances
- Batch-processing instances
- Web application instances
- NAT gateway or instance

18

Take a moment to think about whether the instances in these examples should be put into a public or private subnet.



Recommended subnet selections for each use case

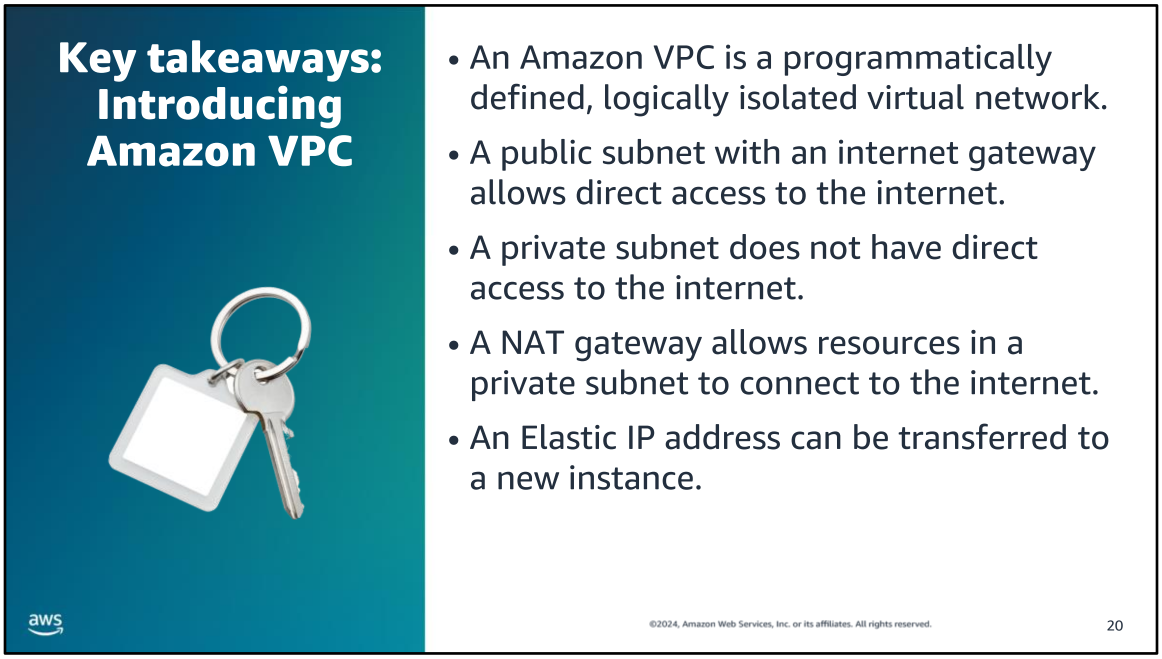
Database instances → Private subnet

Batch-processing instances → Private subnet

Web application instances → Public or private subnet

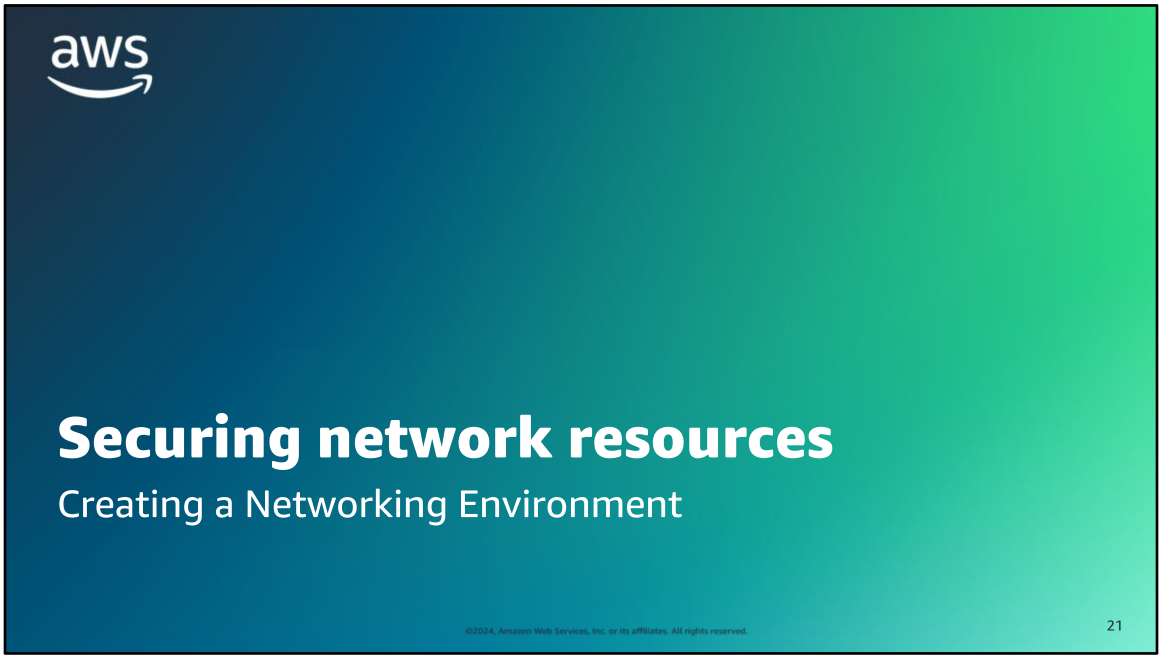
NAT gateway or instance → Public subnet

Data store instances and batch-processing instances should be placed into private subnets. You can put web-tier instances into a public subnet. However, AWS recommends that you place web-tier instances inside private subnets behind a load balancer. In some environments, you must attach web application instances to Elastic IP addresses directly (although you can also attach an Elastic IP address to a load balancer). In those cases, web application instances must be in a public subnet. A NAT gateway or instance must be placed in a public subnet to have access to an internet gateway.



Key takeaways: Introducing Amazon VPC

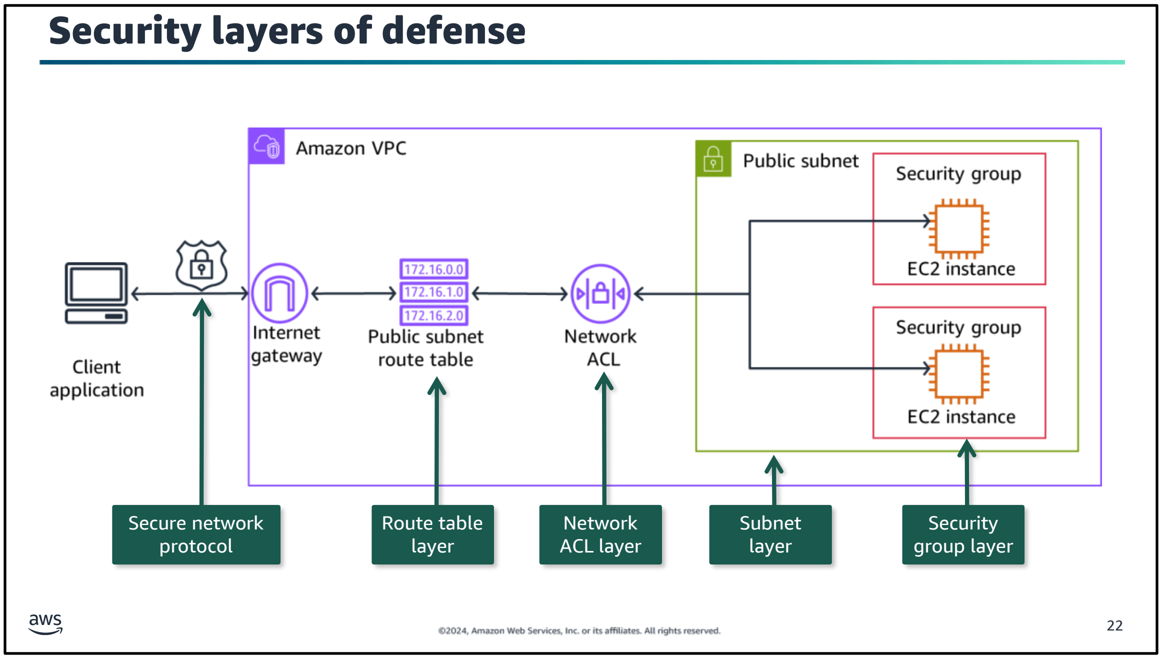
- An Amazon VPC is a programmatically defined, logically isolated virtual network.
- A public subnet with an internet gateway allows direct access to the internet.
- A private subnet does not have direct access to the internet.
- A NAT gateway allows resources in a private subnet to connect to the internet.
- An Elastic IP address can be transferred to a new instance.



Securing network resources

Creating a Networking Environment

This section goes over how to secure network resources.



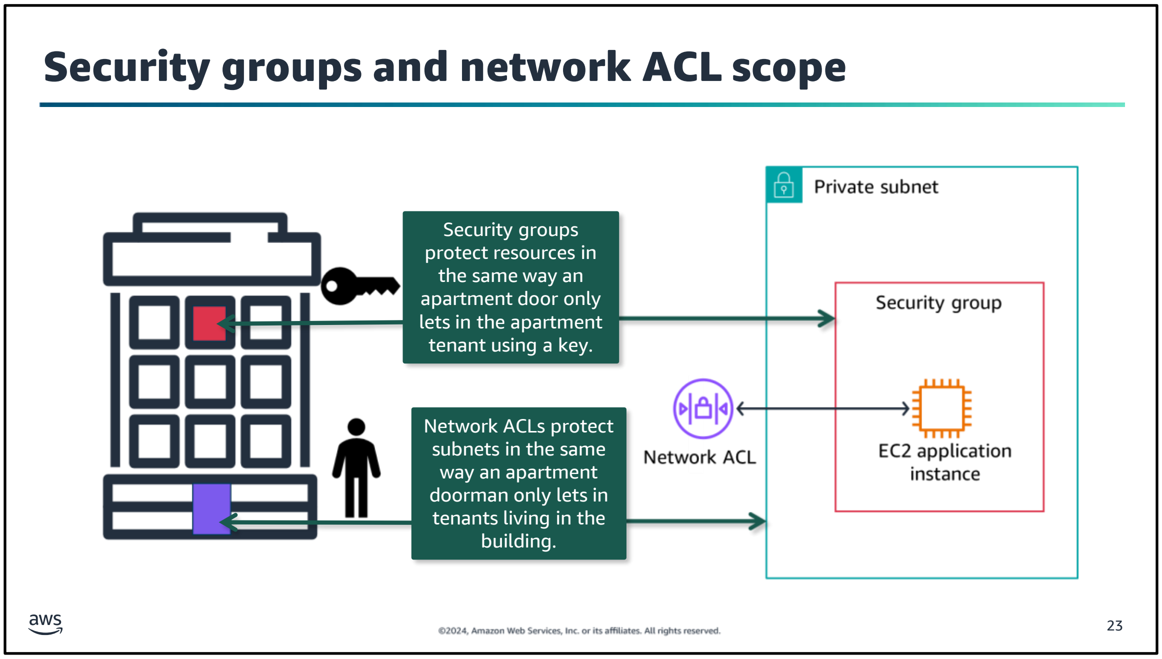
Security layers of defense

22

Image description: Network traffic flow from a client application using a secure network protocol to an internet gateway. The internet gateway forwards network traffic through the route table layer, network access control list (network ACL) layer, subnet layer, and security group layer to EC2 instance. **End of image description.** Isolating resources in a public subnet does not provide enough security measures for VPC resources. Now that you know how to design a network environment and connect it to the internet, you must isolate your applications and workloads.

As a best practice, you should secure your resources with multiple layers of defense. A client application can use a secure network protocol such as TLS and HTTPS to encrypt data payload in transit. This will ensure that third party actors can’t read or impersonate the traffic.

In a VPC, you can add both security groups and network ACLs to further protect your EC2 instances in a subnet. When you implement both network ACLs and security groups as a defense-in-depth way of controlling traffic, a mistake in the configuration of one of these controls won’t expose the instance to unwanted traffic.



Security groups and network ACL scope

Security groups protect resources in the same way an apartment door only lets in the apartment tenant using a key.

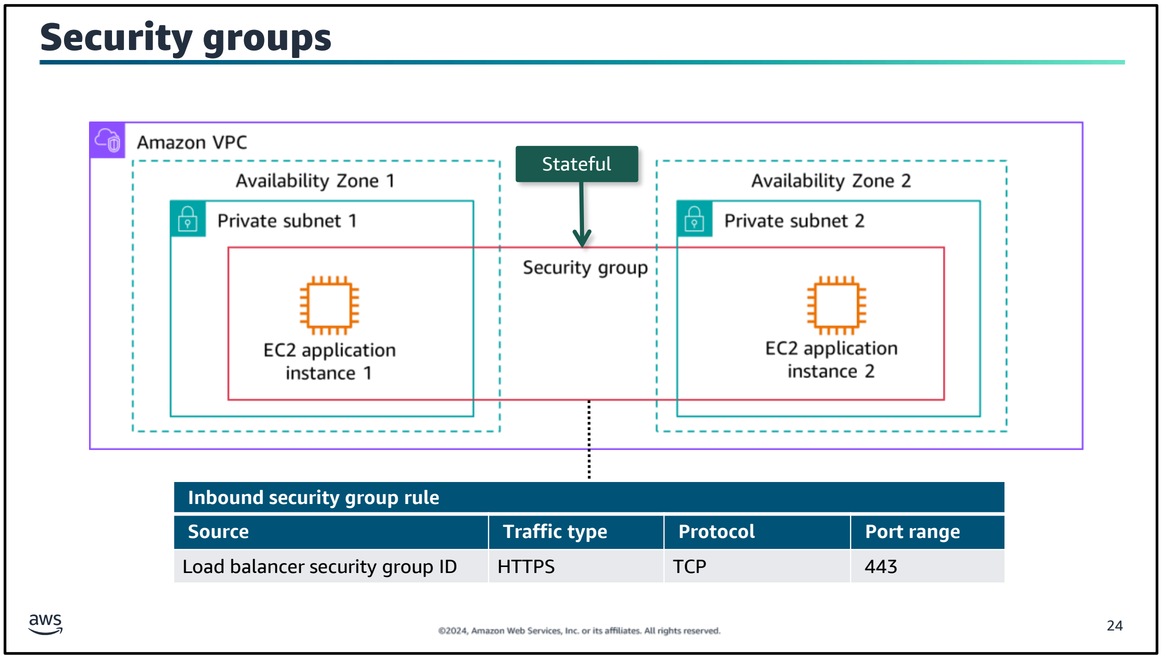
Network ACLs protect subnets in the same way an apartment doorman only lets in tenants living in the building.

23

Security groups and network ACLs both are mechanisms that allow you to define network traffic filters called rules. Both can be used simultaneously, or only one can be used. Think of it as a person that lives in a high rise apartment in a big city.

To get into their own apartment, they first have to enter the building. A doorman protects the building and only lets in the tenants living in the building. In the same way, network ACLs act as a subnet firewall for VPC subnets by only letting in the traffic allowed by the ACL rules.

After the person enters the building, they arrive at their locked apartment door, which protects the apartment. They can only gain entry into the apartment by using their key. In the same way, security groups act as a resource firewall for resources such as EC2 instances. They specify rules to filter network traffic to and from a resource.



Security groups

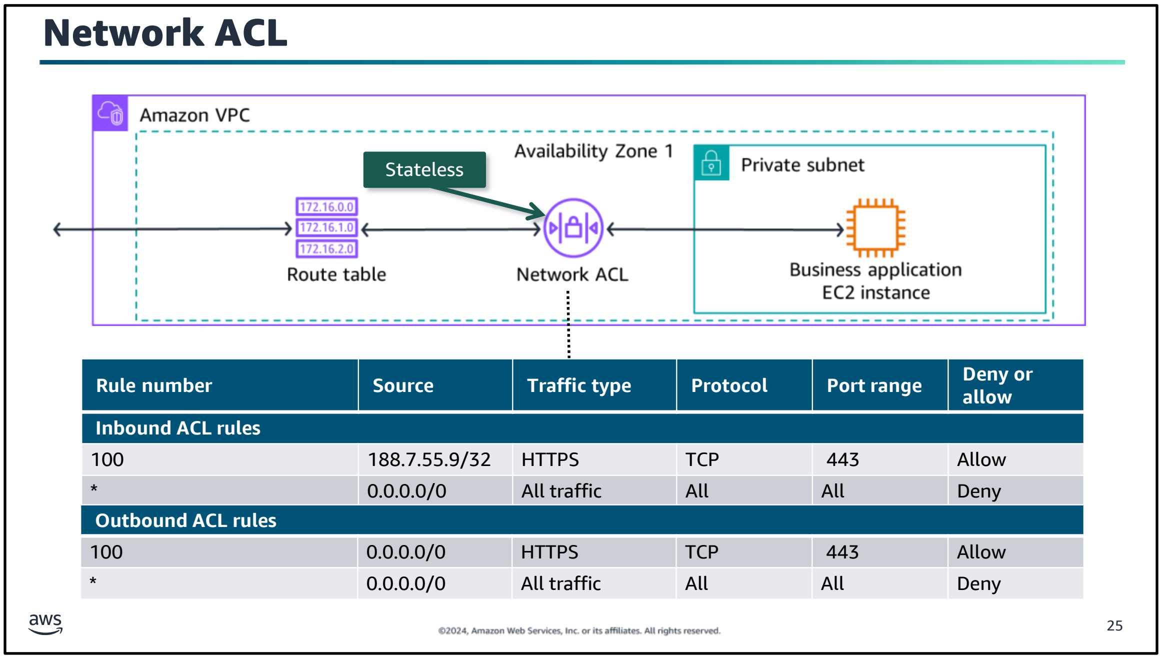
Inbound security group rule			
Source	**Traffic type**	**Protocol**	**Port range**
Load balancer security group ID	HTTPS	TCP	443

Security groups are stateful firewalls that act at the level of instance or a network interface. They can span can span multiple AZs. For each security group, you add separate sets of rules for inbound and outbound traffic. Resources with the same security requirements should be grouped together in one security group. In this example, business application instances across multiple AZs are associated with the same security group.

You can choose the ports and protocols to allow for inbound and outbound traffic. You can specify allow rules, but not deny rules. In the above example, the security group has an inbound security group rule to allow HTTPS TCP traffic from the load balancer security group on port 443. You can also define relationships between security groups. For example, instances within a database tier security group only accept traffic from instances within the application tier, by reference to the security groups applied to the instances involved.

Security groups are stateful. Stateful means that return traffic is automatically allowed, regardless of any rules. For example, say that you are browsing a website from your home computer. It sends a HTTPS request message to the EC2 application instance 1 using the TCP protocol on port 443. Because the inbound security group rules allow HTTPS traffic on port 443, information about the connection is tracked. The response traffic from the EC2 application instance 1 is not tracked as a new request. Instead, it is tracked as an established connection. The response is allowed to flow out of the instance, even if your outbound security group rules restrict outbound HTTPS traffic.

When you first create a security group, it has no inbound rules. Therefore, no inbound traffic is allowed until you add inbound rules to the security group. The new security group does have an outbound rule that allows all outbound traffic from the resource. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic is allowed.



Network ACL

Rule number	Source	Traffic type	Protocol	Port range	Deny or allow
Inbound ACL rules					
100	188.7.55.9/32	HTTPS	TCP	443	Allow
*	0.0.0.0/0	All traffic	All	All	Deny
Outbound ACL rules					
100	0.0.0.0/0	HTTPS	TCP	443	Allow
*	0.0.0.0/0	All traffic	All	All	Deny

25

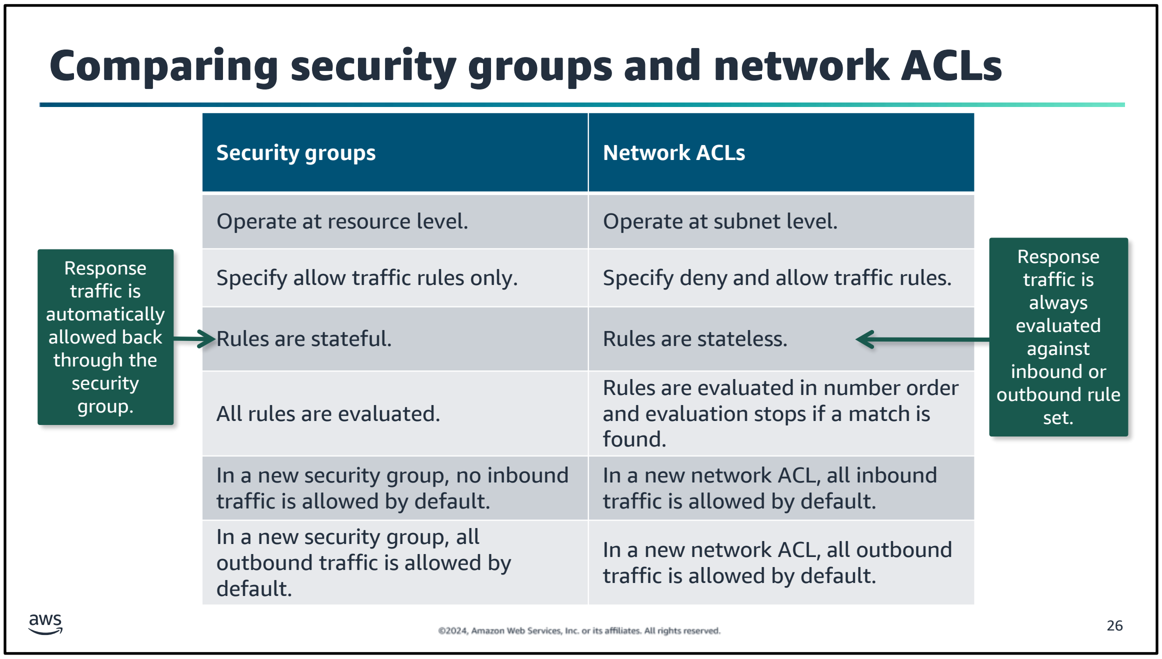
A network access control list (network ACL) is an optional layer of security for your subnet. It acts as a firewall for controlling traffic in and out of one or more subnets. To add another layer of security to your subnet, you can set up network ACLs with rules in the same way that security group rules are set up.

Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL. You can associate a network ACL with multiple subnets. However, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.

A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic. Your VPC automatically comes with a default network ACL. It allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic. It contains two inbound rules for all traffic: rule 100 to allow all traffic on all ports and an asterisk(*) rule to deny all traffic on all ports. The asterisk rule's purpose is to catch any traffic caused by misconfigured rules. Similarly, the outbound rules allows all traffic on all ports with the 100 rule and also has an asterisk rule to deny all traffic on all ports.

If you create a custom network ACL, it will include the star deny all traffic rule for inbound and outbound traffic. You can add numbered rules for specific IP addresses or CIDR blocks to allow or deny traffic for a specific traffic type and port. In the example above, the inbound ACL rule 100 only allows HTTPS TCP traffic from the IP address 188.7.55.9/32 on port 443. The inbound asterisk rule denies all traffic on all ports. The outbound 100 ACL rule allows HTTPS traffic on port 443, while the asterisk rule denies all traffic on all ports.

Network ACLs are stateless, which means that no information about a request is maintained after a request is processed. Return traffic must be explicitly allowed by rules.



Comparing security groups and network ACLs

Security groups	Network ACLs
Operate at resource level.	Operate at subnet level.
Specify allow traffic rules only.	Specify deny and allow traffic rules.
Rules are stateful.	Rules are stateless.
All rules are evaluated.	Rules are evaluated in number order and evaluation stops if a match is found.
In a new security group, no inbound traffic is allowed by default.	In a new network ACL, all inbound traffic is allowed by default.
In a new security group, all outbound traffic is allowed by default.	In a new network ACL, all outbound traffic is allowed by default.

Response traffic is automatically allowed back through the security group.

Response traffic is always evaluated against inbound or outbound rule set.

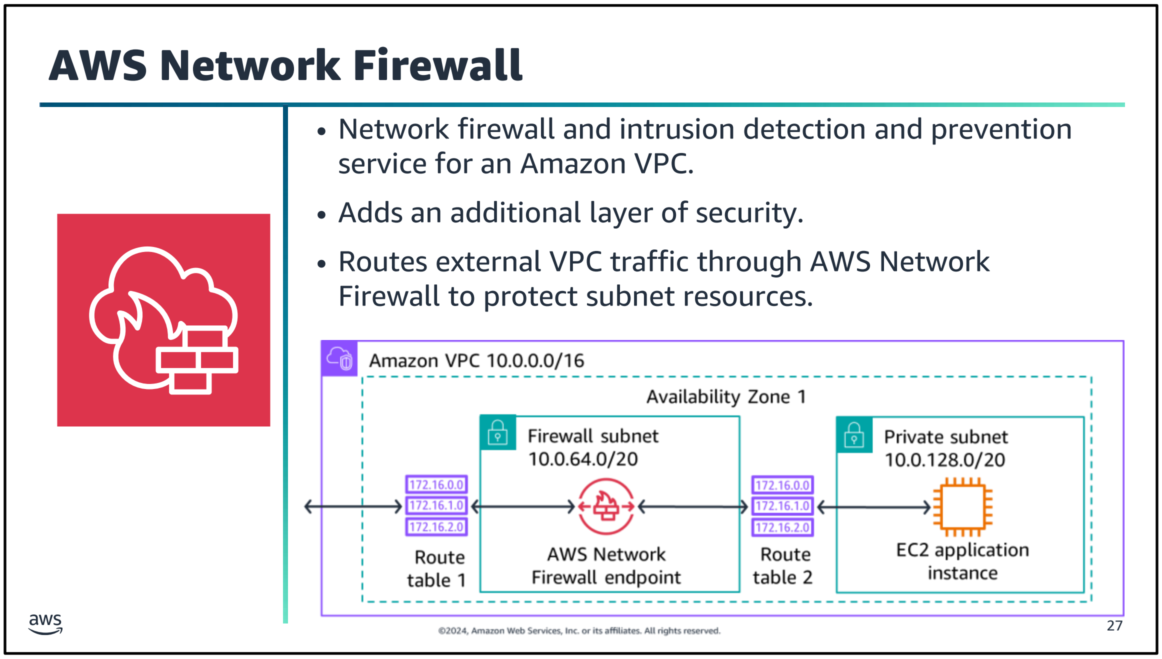
26

It’s important to note the differences between security groups and network ACLs. Security groups operate at resource level while network ACLs operate at subnet level. Security groups only specify allow traffic rules while network ACLs specify deny and allow traffic rules.

Security group rules are stateful, so response traffic is automatically allowed. Network ACLs are stateless, and response traffic is evaluated with an inbound or outbound rule set.

All inbound or outbound rules in the security group are evaluated at run time. With network ACLs, the inbound and outbound rules are evaluated in number order and the evaluation stops when the traffic matches the rule.

By default, security groups don’t allow any inbound traffic but allow all outbound traffic. Network ACLs allow all inbound and outbound traffic by default. AWS recommends to use security groups over network ACLs where possible for ease of maintenance and flexibility.



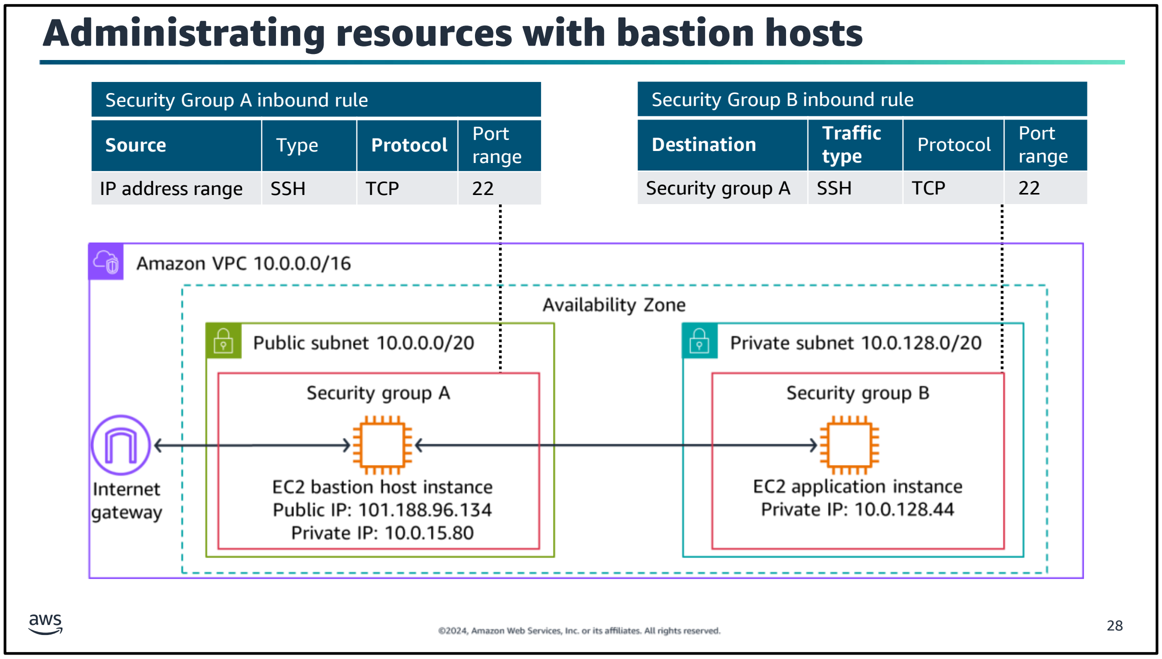
AWS Network Firewall

- Network firewall and intrusion detection and prevention service for an Amazon VPC.
- Adds an additional layer of security.
- Routes external VPC traffic through AWS Network Firewall to protect subnet resources.

When organizations are dealing with sensitive data or have strict compliance requirements, AWS Network Firewall is available as another security layer to add to security groups and network ACLs. AWS Network firewall acts as a buffer to ensure traffic sent to VPC resources is not malicious.

It’s a stateful, managed network firewall and intrusion detection and prevention service for a VPC. It’s deployed in a firewall subnet which inspects all incoming VPC traffic to protect VPC subnets. To implement an AWS Network Firewall endpoint, modify your Amazon VPC route tables to send your network traffic through the Network Firewall firewall endpoints. Route table 1 and route table 2 are modified to route traffic through the firewall endpoint. In the example, all traffic external to the VPC is then routed first through route table 1, to the AWS Network Firewall endpoint, and then through route table 2 to the EC2 application instance in the VPC private subnet.

Route table 1 would be modified to have an all traffic destination from an external target like an internet gateway ID. Route table 2 would accept only traffic from firewall subnet with a destination of all traffic and a target of firewall subnet ID.



Administrating resources with bastion hosts

Security Group A inbound rule			
Source	Type	**Protocol**	Port range
IP address range	SSH	TCP	22

Security Group B inbound rule			
Destination	**Traffic type**	Protocol	Port range
Security group A	SSH	TCP	22

Amazon VPC 10.0.0.0/16

Availability Zone

Public subnet 10.0.0.0/20

Security group A

EC2 bastion host instance
Public IP: 101.188.96.134
Private IP: 10.0.15.80

Private subnet 10.0.128.0/20

Security group B

EC2 application instance
Private IP: 10.0.128.44

Internet gateway

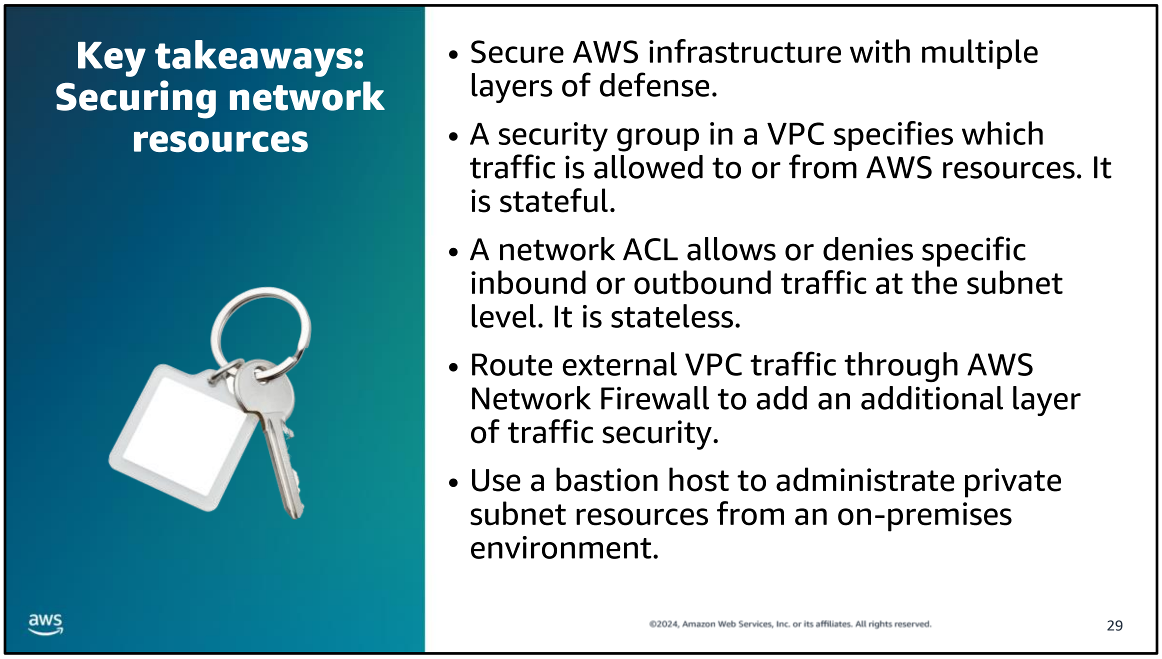
aws

28

One of the options companies use to administrate resources in a VPC private subnet, is to use a bastion host in a public subnet. A bastion host is a server that provides maintenance access to a private subnet from an external network instead of providing direct access to an instance in a private subnet. The administrator is granted access to the bastion host with a security group rule. The bastion host EC2 instances have IAM policies attached to provide access to resources in the VPC. You can use a bastion host to minimize the chances of potential attack on resources in your private subnet while allowing restricted access for server administration purposes.

For example, suppose you want to allow connections from an external network to Linux instances in a private subnet of your VPC through Secure Shell, or SSH. You can use a bastion host to mitigate the risk of allowing these external SSH connections to the instances in the private subnet.

In the example above, the bastion host is an EC2 instance in a public subnet of the VPC. The bastion host user connects from an on-premises environment to the bastion host. Security group A allows administration traffic with a rule allowing SSH TCP traffic on port 22 from a specified IP address range. The EC2 application instance in the private subnet in security group B allows SSH TCP traffic on port 22 from security group A. Ideally, the bastion host should be the only source of SSH traffic to your Linux instance in the private subnet.



Key takeaways: Securing network resources

- Secure AWS infrastructure with multiple layers of defense.
- A security group in a VPC specifies which traffic is allowed to or from AWS resources. It is stateful.
- A network ACL allows or denies specific inbound or outbound traffic at the subnet level. It is stateless.
- Route external VPC traffic through AWS Network Firewall to add an additional layer of traffic security.
- Use a bastion host to administrate private subnet resources from an on-premises environment.

29

Demo: Creating an Amazon VPC in the AWS Management Console

- This demo uses Amazon VPC features, security groups, and an Elastic IP address.
- In this demonstration, you will see how to create a public and private subnet in the VPC each with a subnet route table.
- You will see how to create an internet gateway and attach it to the VPC and configure internet routes.
- Create a NAT gateway an assign an Elastic IP address. Configure NAT gateway routes.
- Create a web server security group.
- Create a database server security group.

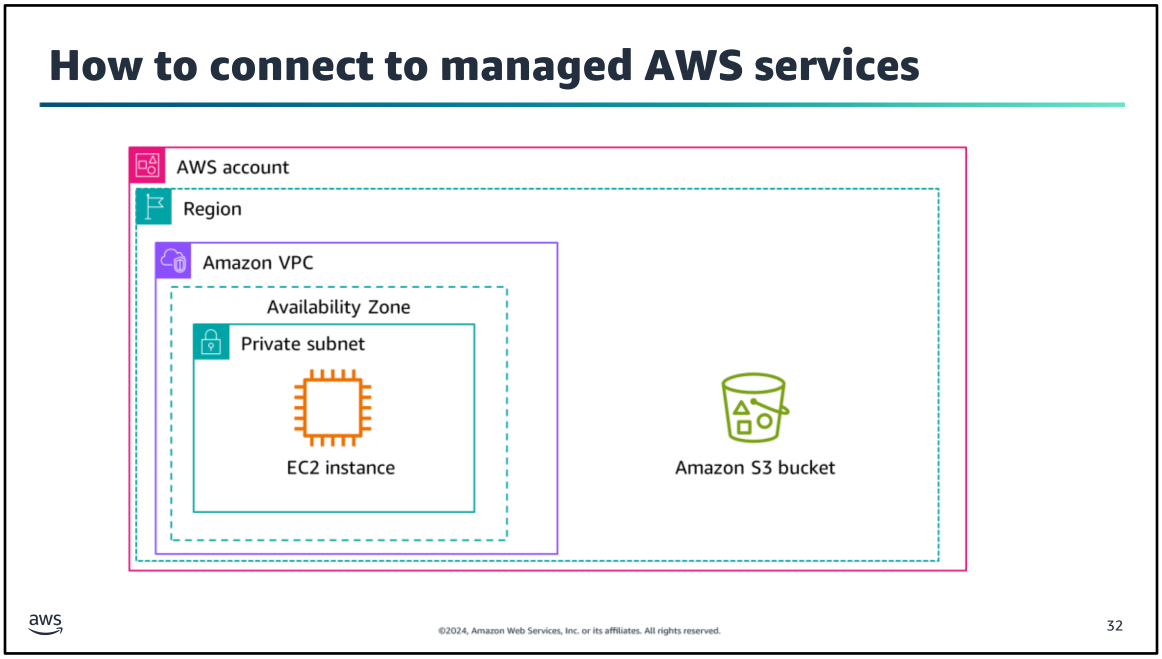
Find this recorded demo in your course as part of this module.

Connecting to managed AWS services

Creating a Networking Environment

31

This section goes over how to connect to managed AWS services from a VPC.



How to connect to managed AWS services

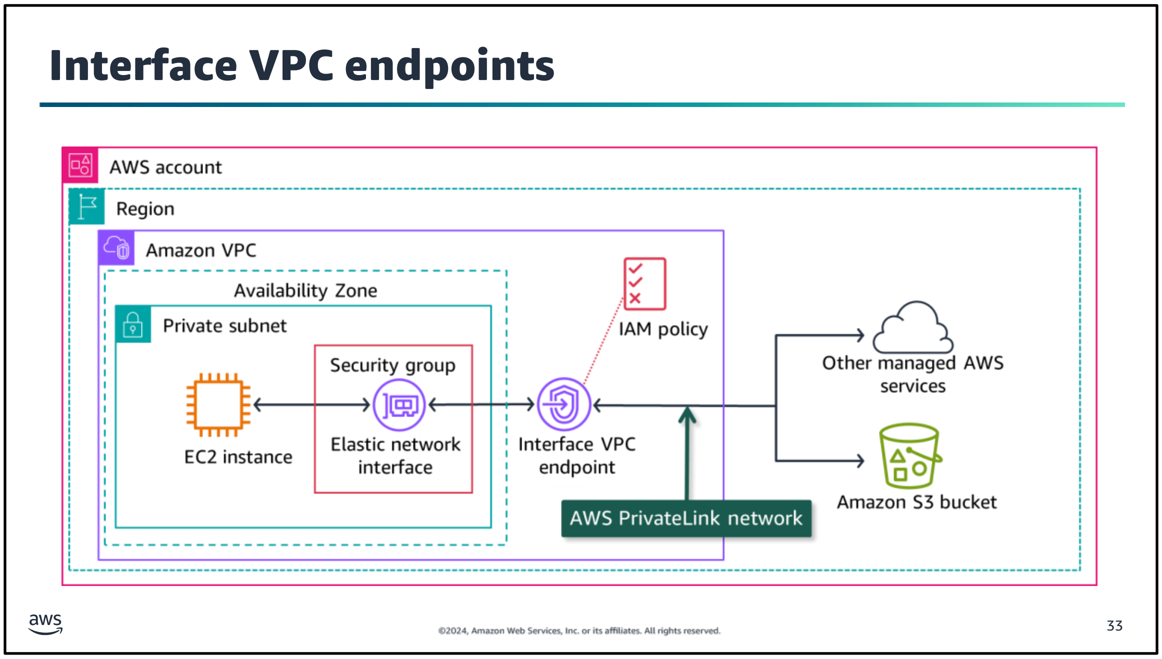
©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

32

Image description: Architectural diagram of an AWS VPC with an EC2 instance in the private subnet. An Amazon S3 bucket using separate Gateway VPC endpoint. The EC2 instance can access other AWS managed services and Amazon S3 using an interface VPC endpoint. **End of image description.**

Let’s say you successfully deployed and configured your workload on an EC2 instance in a private subnet. However, the workload needs access to an Amazon Simple Storage Service (Amazon S3) bucket in your AWS account in the same AWS Region. Amazon S3 is a managed AWS service and operates outside of your VPC. There’s no direct connectivity from the EC2 instance within your VPC to the Amazon S3 bucket.

One solution would be that Amazon S3 buckets can be reached with a public Region access point from VPC resources. The problem is that the request will be routed though the internet, incurring data transfer costs.



Interface VPC endpoints

AWS account

Region

Amazon VPC

Availability Zone

Private subnet

Security group

EC2 instance

Elastic network interface

IAM policy

Interface VPC endpoint

AWS PrivateLink network

Other managed AWS services

Amazon S3 bucket

33

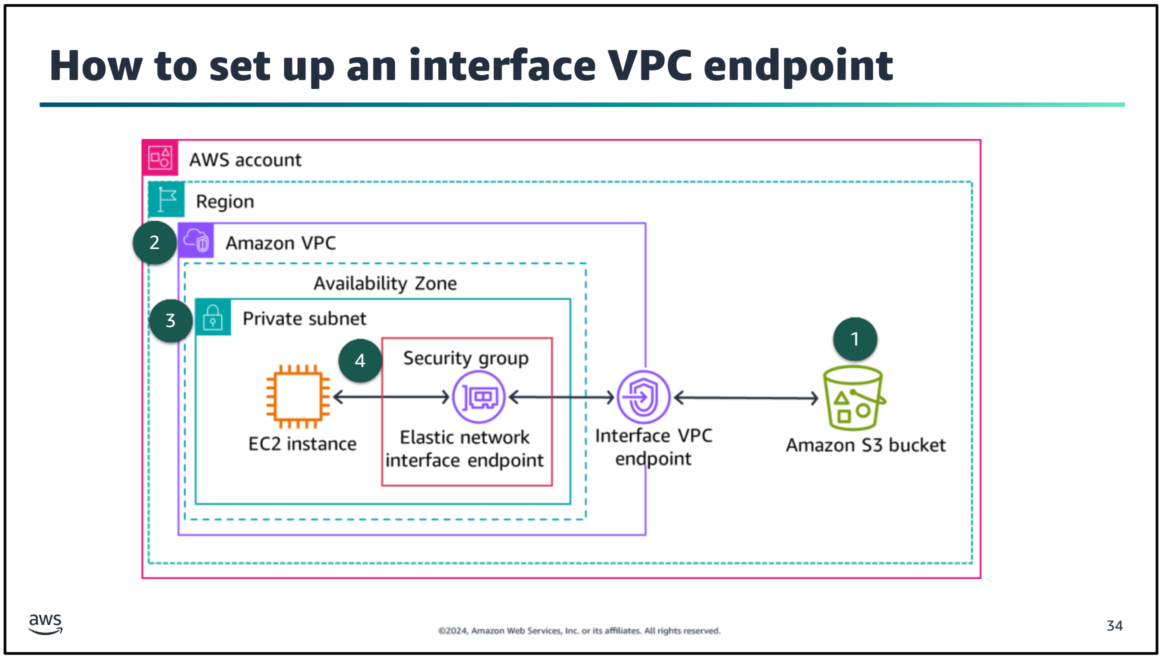
Image description: Architectural diagram of an AWS VPC with an interface VPC endpoint. The VPC contains a private subnet with an elastic network interface. The EC2 instance inside the private subnet can access other AWS managed services and Amazon S3 using the elastic network interface belonging to the interface VPC endpoint. The interface VPC endpoint has an IAM policy attached. **End of image description.**

A more direct and secure solution to connect the EC2 instance to an Amazon S3 bucket is to use a VPC endpoint. There are two types of VPC endpoints: interface VPC endpoints and gateway VPC endpoints. Interface endpoints are provided by AWS PrivateLink. The AWS PrivateLink service is designed to solve connectivity between VPCs and managed AWS services. For a complete list of AWS PrivateLink supported AWS services, refer to the resource list.

Interface VPC endpoints allows you to privately connect your VPC to AWS managed services as if they were in your VPC. In the above example, the EC2 instance can use the interface VPC endpoint to connect to the Amazon S3 bucket using AWS PrivateLink network.

For every interface VPC endpoint, AWS creates an elastic network interface in every VPC subnet and assigns it a private IP address from the subnet address range. An elastic network interface is a logical networking component in a VPC that represents a virtual network card.

To control user or application access to the VPC, you can use an AWS Identity and Access Management (IAM) resource policy. This will separately secure the VPC endpoint and accessible resources. The interface VPC endpoints are priced for hourly endpoint usage and the amount of data processed per month.



How to set up an interface VPC endpoint

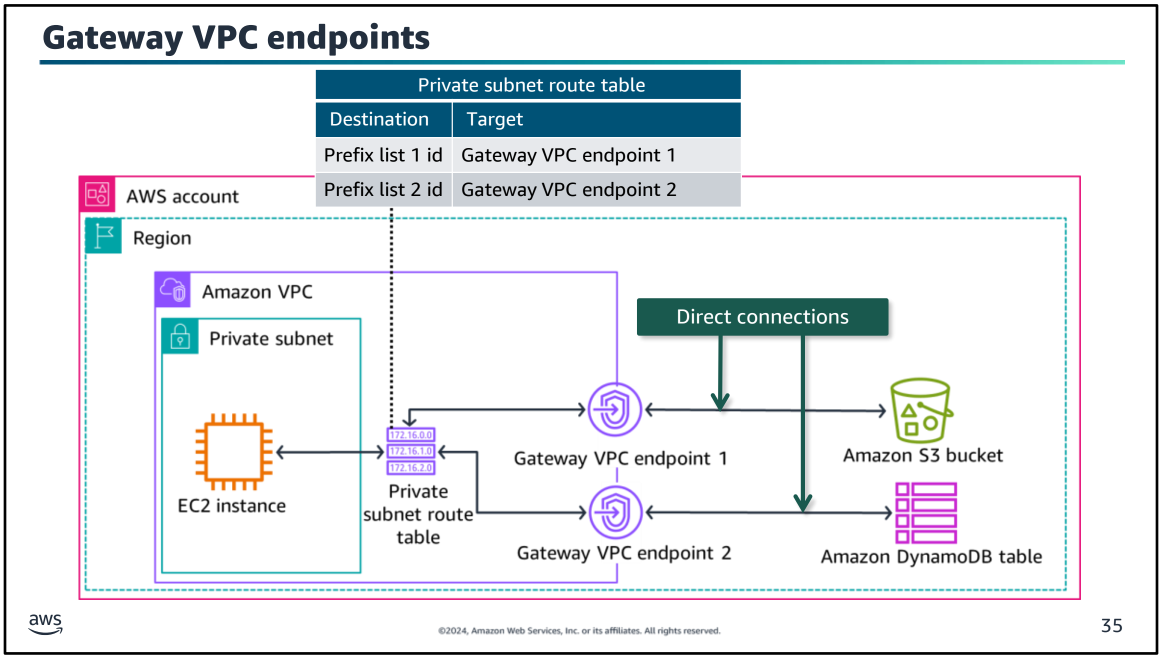
34

Image description: Architectural diagram of an AWS VPC with an interface VPC endpoint. The VPC contains a private subnet with an elastic network interface. The EC2 instance inside the private subnet can Amazon S3 using the network interface belonging to the interface VPC endpoint. **End of image description.**

To set up an interface endpoint, follow these general steps from the Amazon VPC console:

1. Specify the name of the AWS service, endpoint service, or AWS Marketplace service that you want to connect to.
2. Choose the VPC where you want to create the interface endpoint. You can specify more than one subnet in different Availability Zones (AZs), as supported by the service. Doing so helps ensure that your interface endpoint is resilient to AZ failures. In that case, an elastic network interface is created in each subnet that you specify.
3. Choose a subnet in your VPC that will use the interface endpoint. When you create an interface endpoint for a service in your VPC, a network interface is created in the selected subnet. The network interface has a private IP address that serves as an entry point for traffic destined to the service.
4. Specify the security groups to associate with the network interface. The security group rules control the traffic to the network interface from resources in your VPC. If you do not specify a security group, the default security group for the VPC is used.

Services cannot initiate requests to resources in your VPC through the endpoint. An endpoint only returns responses to traffic that are initiated from resources in your VPC.



Gateway VPC endpoints

Private subnet route table	
Destination	**Target**
Prefix list 1 id	Gateway VPC endpoint 1
Prefix list 2 id	Gateway VPC endpoint 2

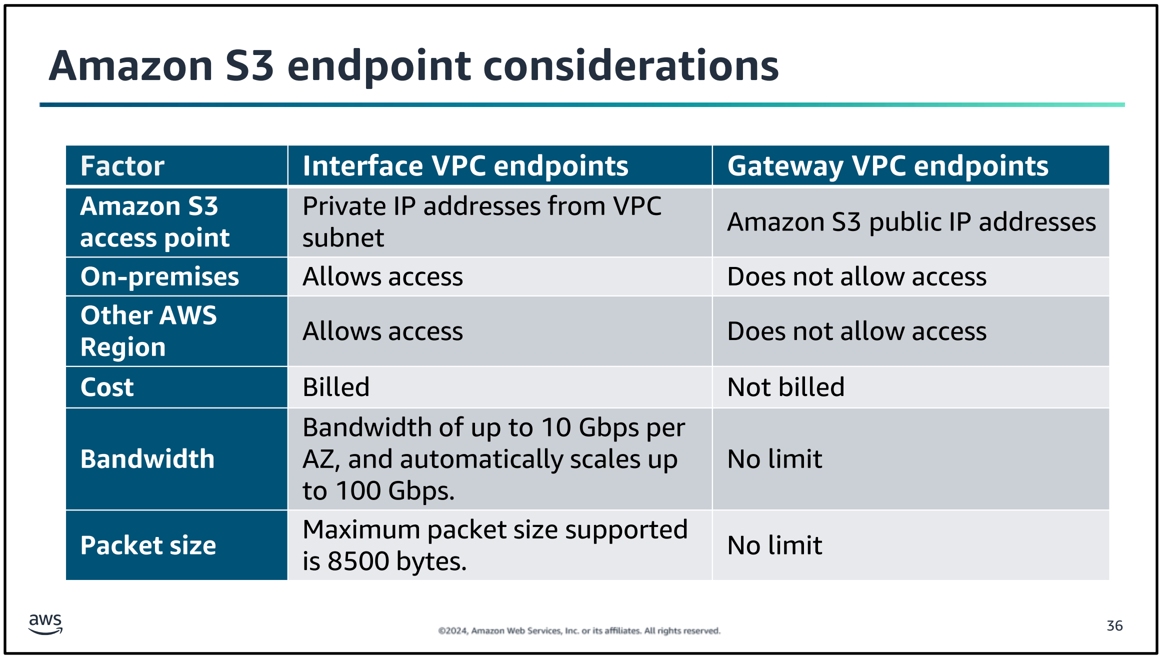
35

Image description: Architectural diagram of an AWS VPC with two gateway VPC endpoints. The EC2 instance in the private subnet can connect to Amazon DynamoDB and Amazon S3 each using separate gateway VPC endpoints through a private subnet route table. **End of image description.**

Another way to connect an EC2 instance to an Amazon S3 bucket is to use a gateway VPC endpoint. A gateway VPC endpoint connects directly to Amazon S3 and Amazon DynamoDB using route tables, not making use of AWS PrivateLink. Amazon S3 and DynamoDB are the only services that gateway VPC endpoints support.

In the example on the slide, the EC2 instance connects to an Amazon S3 bucket through Gateway VPC endpoint 1. The route specified in the private subnet route table is a destination of the prefix list 1 ID and a target of Gateway VPC endpoint 1. A prefix list is a group of CIDR blocks. Similarly, the EC2 instance can connect to a DynamoDB table through Gateway VPC endpoint 2.

Amazon S3 supports both gateway endpoints and interface endpoints. There’s no additional charge for using gateway endpoints. There are no throughput packet limits.



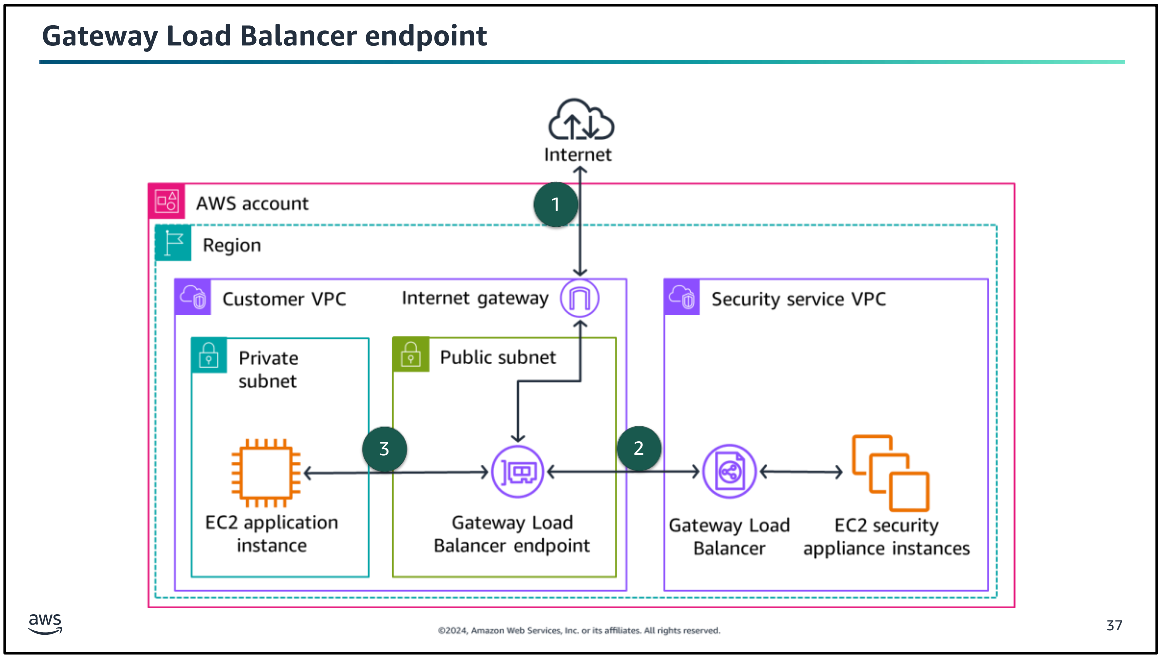
Amazon S3 endpoint considerations

Factor	Interface VPC endpoints	Gateway VPC endpoints
Amazon S3 access point	Private IP addresses from VPC subnet	Amazon S3 public IP addresses
On-premises	Allows access	Does not allow access
Other AWS Region	Allows access	Does not allow access
Cost	Billed	Not billed
Bandwidth	Bandwidth of up to 10 Gbps per AZ, and automatically scales up to 100 Gbps.	No limit
Packet size	Maximum packet size supported is 8500 bytes.	No limit

36

To decide whether to choose an interface VPC endpoint or a gateway VPC endpoint to communicate with S3, take the following items under consideration:

- Can the objects in the Amazon S3 bucket be accessed through the Amazon S3 public IP address or more securely through a private IP address in a VPC subnet?
- Should the Amazon S3 bucket allow on-premises access?
- Is access required from another AWS Region?
- Is the budget large enough to accommodate interface VPC endpoint costs?
- How much bandwidth throughput is required?
- What is the maximum network traffic packet size needed?



Gateway Load Balancer endpoint

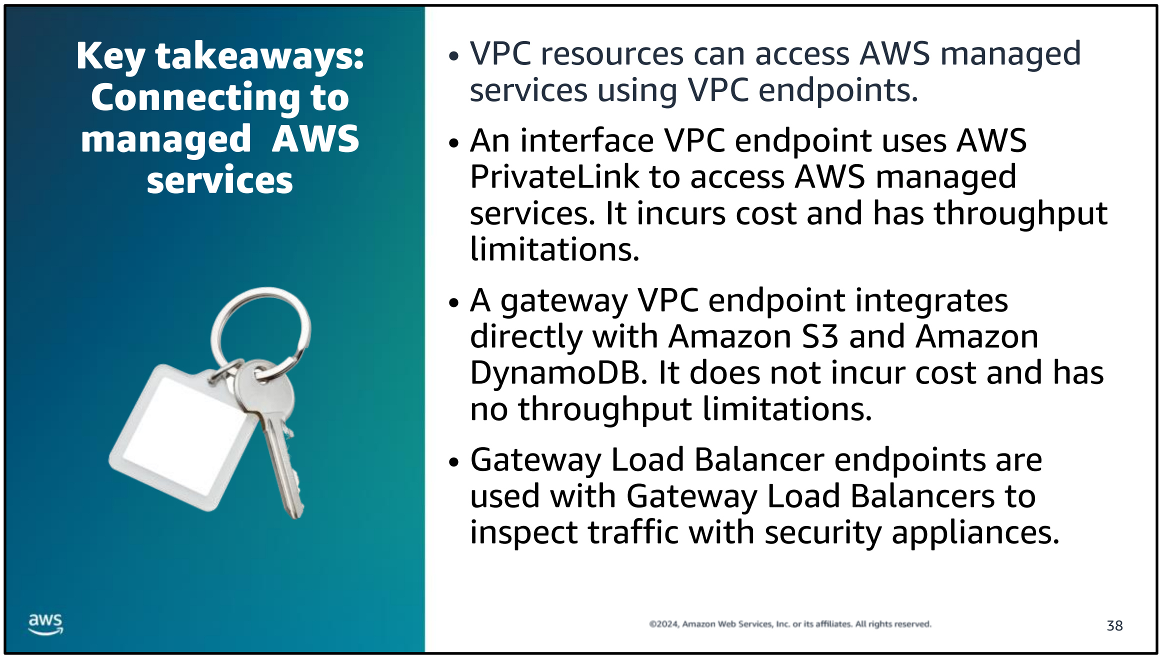
37

Image description: VPC 1 has a private subnet containing an EC2 instance and a public subnet with an attached Gateway Load Balancer endpoint and an internet gateway. VPC 2 has a Gateway Load Balancer and a EC2 security appliance instance. **End of image description.**

Another specialized type of VPC endpoint is a Gateway Load Balancer endpoint. In this example, the Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between security appliances in VPC 2 and the application instance in VPC 1. The Gateway Load Balancer is deployed in VPC 2.

1. All traffic entering VPC 1 through the internet gateway is first routed to the Gateway Load Balancer endpoint in VPC 1.
2. The traffic is then routed to the Gateway Load Balancer in VPC 2. The Gateway Load Balancer distributes the traffic to the EC2 security appliance for inspection. The security appliance responds to the Gateway Load Balancer which returns the inspected traffic to the Gateway Load Balancer endpoint.
3. The Gateway Load Balancer endpoint sends the traffic to the EC2 application instance.

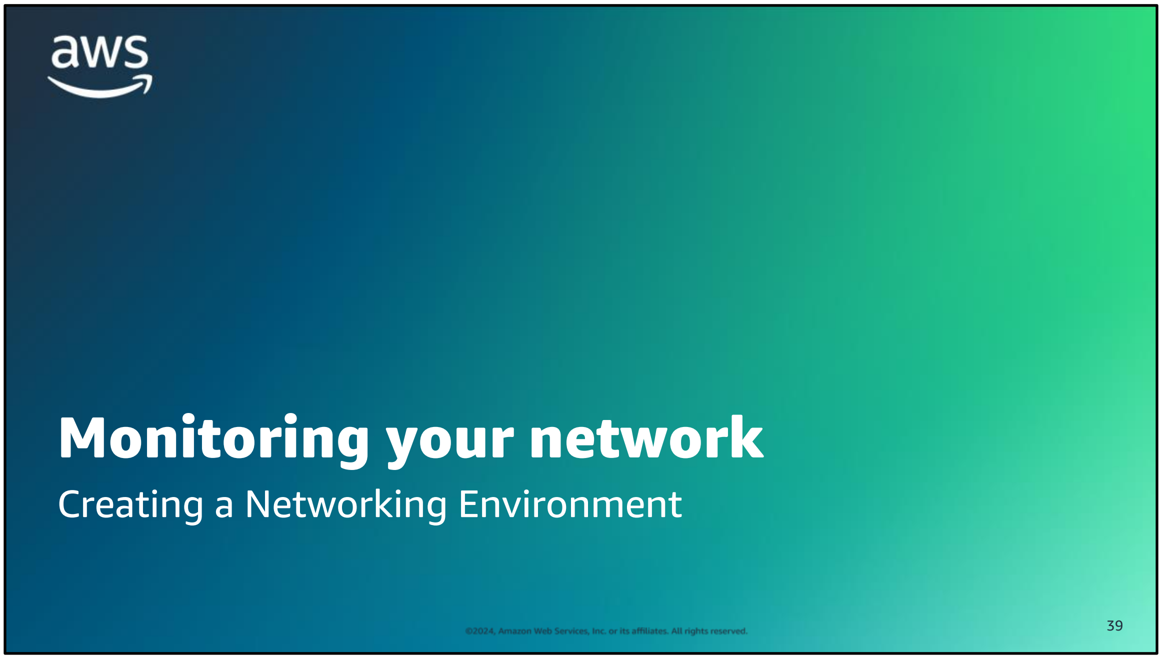
Similarly, all traffic leaving the EC2 application instance follows the same path as incoming traffic.



Key takeaways: Connecting to managed AWS services

- VPC resources can access AWS managed services using VPC endpoints.
- An interface VPC endpoint uses AWS PrivateLink to access AWS managed services. It incurs cost and has throughput limitations.
- A gateway VPC endpoint integrates directly with Amazon S3 and Amazon DynamoDB. It does not incur cost and has no throughput limitations.
- Gateway Load Balancer endpoints are used with Gateway Load Balancers to inspect traffic with security appliances.

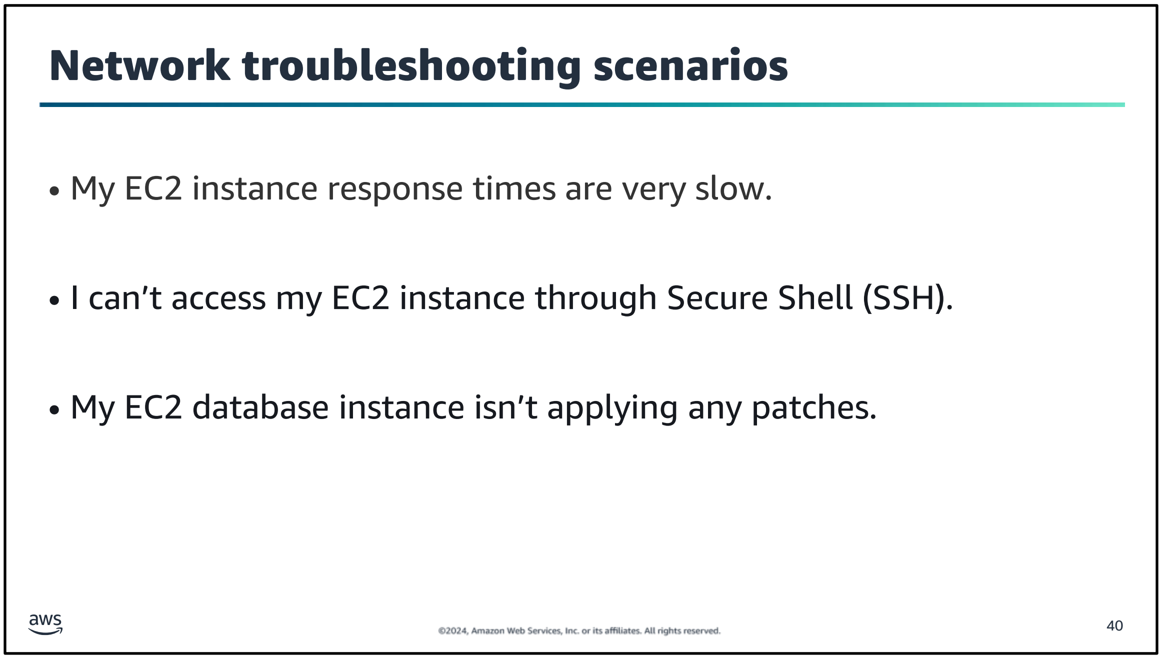
38



Monitoring your network

Creating a Networking Environment

This section goes over how to monitor your network in AWS.



Network troubleshooting scenarios

- My EC2 instance response times are very slow.
- I can’t access my EC2 instance through Secure Shell (SSH).
- My EC2 database instance isn’t applying any patches.

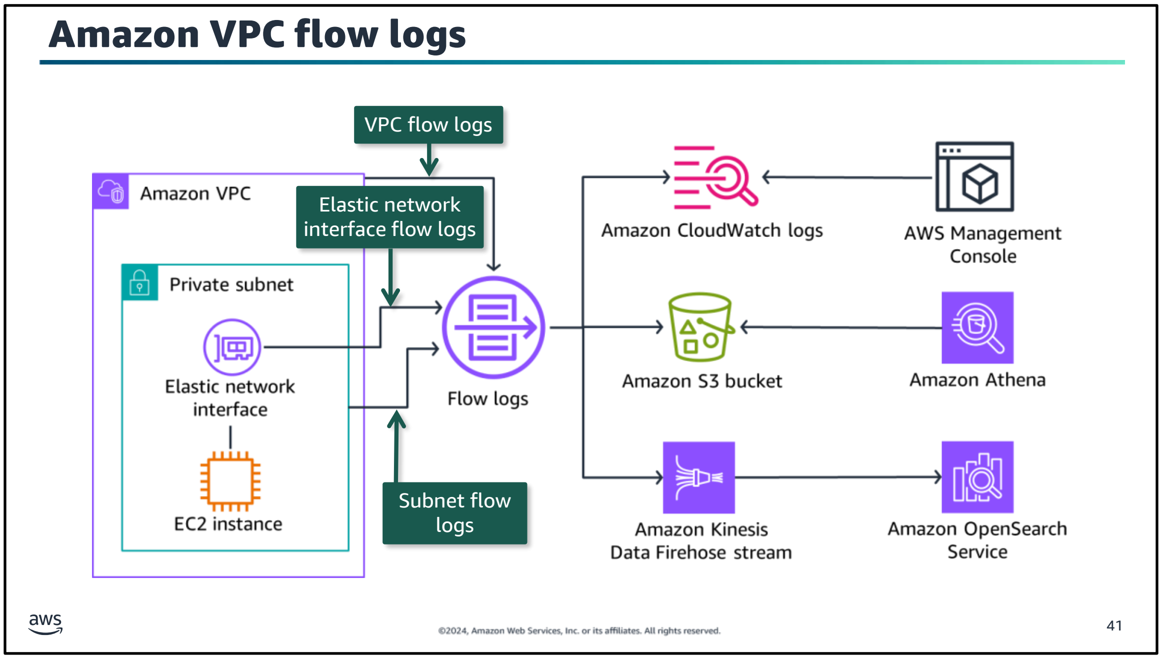
When your Amazon VPC is configured and in use with resources deployed in subnets, issues can occur where further investigation is needed. For this reason, it’s important to monitor your network and build automated recovery processes.

One scenario is to find out why EC2 instance response times are slowing down. Is there unnecessary or unwanted traffic using the EC2 instance resources? For example, distributed denial of service (DDoS) attacks are spam bots that generate messages to overload a network or a server to the point where legitimate user traffic can’t reach the server or be successfully processed.

When traffic isn’t reaching a destination, it can be the result of security groups that have rules that don’t allow the traffic through. This would be an example of a configuration issue. For example, if you are sending SSH traffic on port 22 to an instance and the security group is not configured to allow port 22 traffic, it will result in a connectivity error.

If you have an EC2 database instance in a private subnet and automated patches aren’t being applied, you would have to check the security group and route tables NAT gateway configurations.

To supply detailed information about traffic flow, you can use Amazon VPC flow logs to troubleshoot network issues.



Amazon VPC flow logs

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

41

Image description: Network traffic information is collected by VPC Flow Logs from Amazon VPC, a private subnet, and an elastic network interface attached to an EC2 instance. VPC Flow Logs delivers network traffic to Amazon CloudWatch logs, Amazon S3 buckets, or Amazon Kinesis Data Firehose streams. To view logs, users can use the AWS Management Console for Amazon CloudWatch logs and Amazon Athena for Amazon S3. Amazon Kinesis Data Firehose streams can deliver flow logs to Amazon OpenSearch Service. **Image description end.**

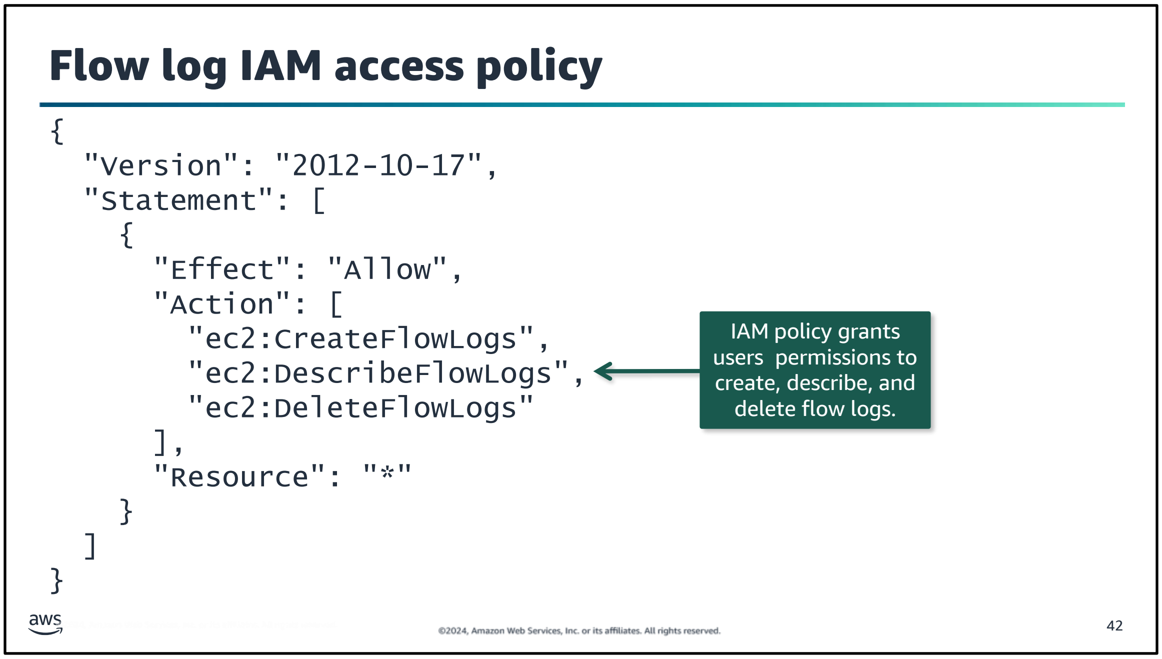
When you want to see detailed information about the traffic in your VPC, you can activate VPC Flow Logs. It’s an Amazon VPC feature that you can use to capture packet-level information about the network traffic in your VPC.

When creating a flow log, you can choose to capture all traffic, accepted traffic, or only rejected traffic. You also choose to capture metadata about the whole VPC, a specific subnet, or elastic network interface.

The flow logs can then be delivered to a destination of your choice. If flow logs are delivered to Amazon CloudWatch, users can use the AWS Management Console to filter and view logs. If flow logs are delivered to Amazon S3, then users can use services like Amazon Athena to interactively query the logs. Flow logs in Amazon S3 can be published as plain text or Parquet.

Amazon Kinesis Data Firehose can deliver logs to AWS services such as Amazon OpenSearch Service, which will make the logs available through the Amazon OpenSearch dashboard. Amazon Kinesis Data Firehose can also deliver logs to third party log solutions such as Splunk.

Flow logs operate outside of your VPC network and will not affect your VPC latency and performance.



Flow log IAM access policy

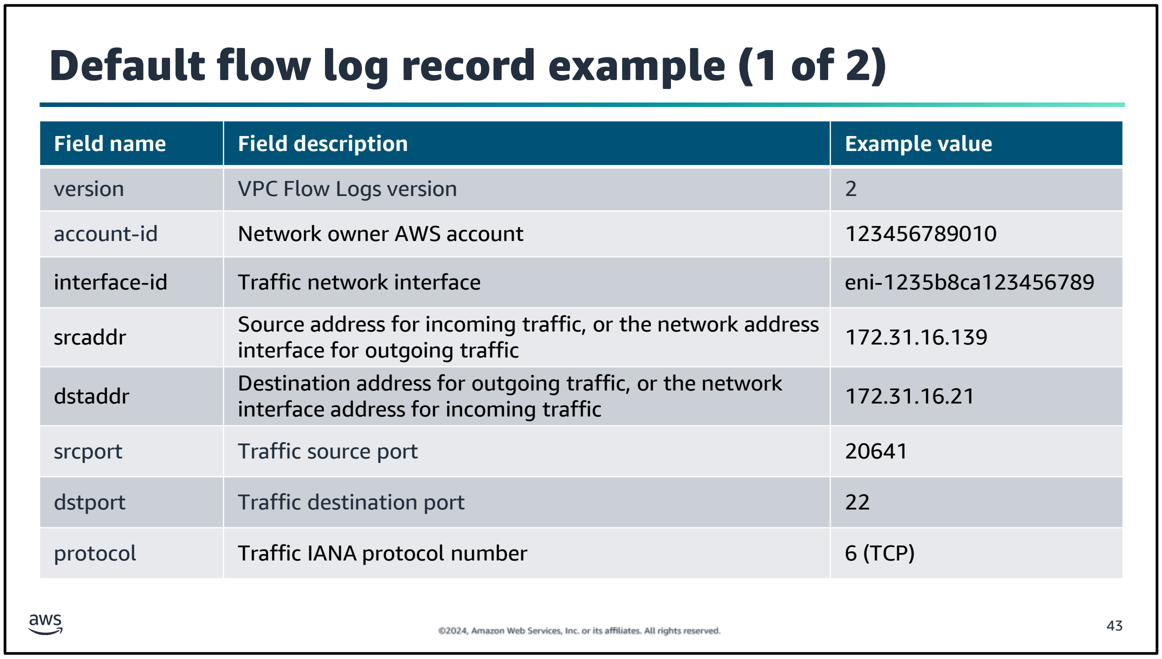
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateFlowLogs",
        "ec2:DescribeFlowLogs",
        "ec2:DeleteFlowLogs"
      ],
      "Resource": "*"
    }
  ]
}
```

IAM policy grants users permissions to create, describe, and delete flow logs.

42

By default, users don’t have permission to work with flow logs. You can create an AWS Identity and Access Management (IAM) role with a policy attached that grants users the permissions to create, describe, and delete flow logs as in the example above. Attach the role to an IAM user or user group to allow access to flow logs.

In the example above, the IAM policy has an effect of allowing an IAM user to run the following actions: create, describe, and delete EC2 flow logs on any AWS resource.



Default flow log record example (1 of 2)

Field name	Field description	Example value
version	VPC Flow Logs version	2
account-id	Network owner AWS account	123456789010
interface-id	Traffic network interface	eni-1235b8ca123456789
srcaddr	Source address for incoming traffic, or the network address interface for outgoing traffic	172.31.16.139
dstaddr	Destination address for outgoing traffic, or the network interface address for incoming traffic	172.31.16.21
srcport	Traffic source port	20641
dstport	Traffic destination port	22
protocol	Traffic IANA protocol number	6 (TCP)

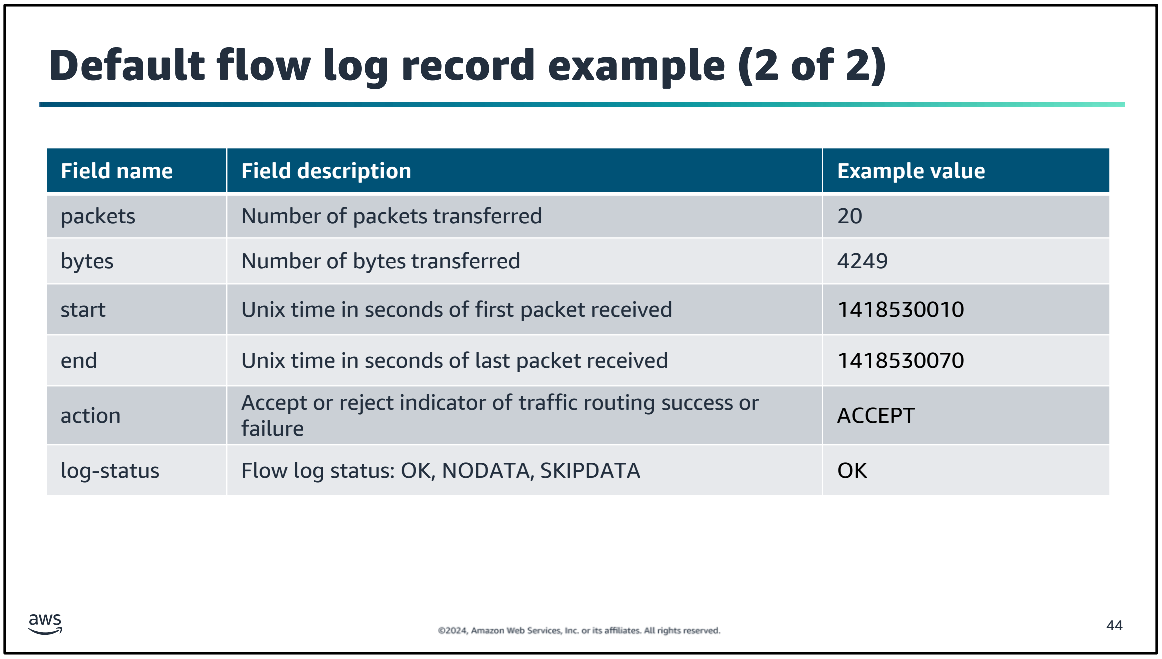
Image description: Image description end.

Flow log data for a monitored network interface is recorded as flow log records, which are log events consisting of fields that describe the traffic flow. The flow log record consists of all flows within an aggregation interval or capture window.

Flow logs have a version number depending on the fields that you choose to capture. Always include the highest version number used over all the fields. The default flow log format, as in the example above, uses version 2 fields.

Customer information such as AWS account and the network interface id can also be captured. Traffic metadata captured includes data such as source and destination IP addresses, source and destination ports, and Internet Assigned Numbers Authority (IANA) protocol number.

In the example above, example values are provided for SSH traffic with a destination port of 22 using TCP protocol to a network interface eni-1235b8ca123456789 in account 123456789010. This is a typical example of a system administrator sending a request to an EC2 instance in a private subnet from an on-premises corporate network environment.



Default flow log record example (2 of 2)

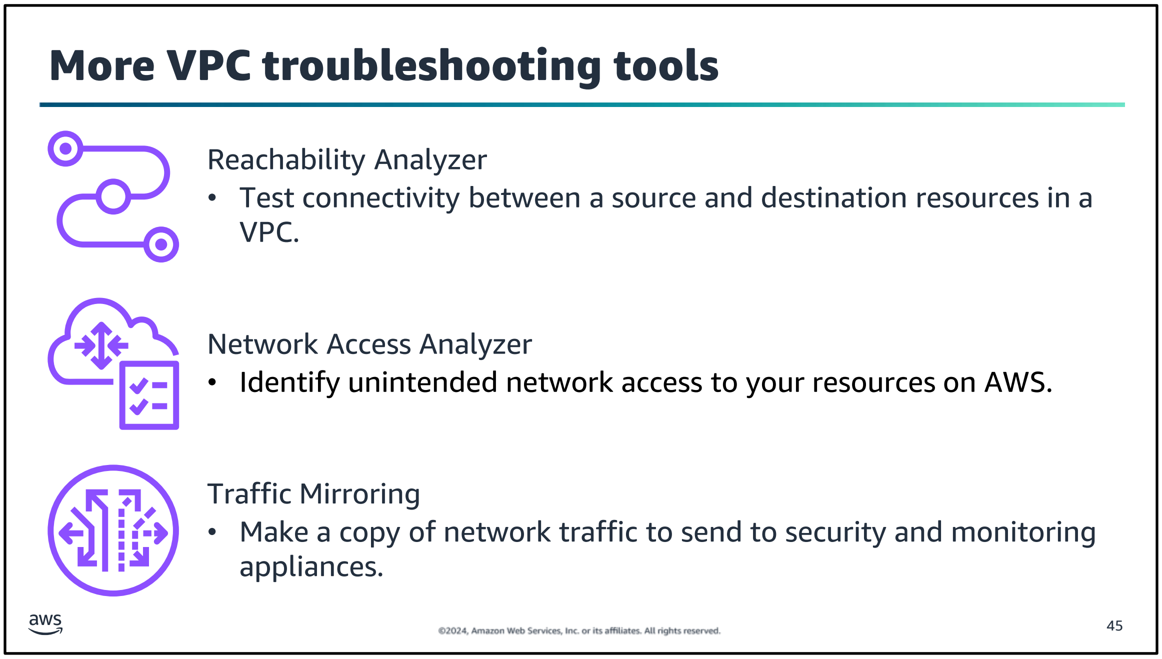
Field name	Field description	Example value
packets	Number of packets transferred	20
bytes	Number of bytes transferred	4249
start	Unix time in seconds of first packet received	1418530010
end	Unix time in seconds of last packet received	1418530070
action	Accept or reject indicator of traffic routing success or failure	ACCEPT
log-status	Flow log status: OK, NODATA, SKIPDATA	OK

44

Additional information that is available on the default flow logs are the number of packets and bytes transferred and the time of the first and last packet that was received.

The action field indicates whether the traffic was accepted or rejected. An example scenario of rejected traffic can be because of security group or network ACL rules preventing traffic delivery. Another reason traffic can be rejected is packets that arrived after the connection was closed.

To monitor whether the flow logs are being logged, you can use the log-status field. If it has a value of “OK,” then data logging is functioning normal to the chosen destinations. If the log-status field has a value of “NODATA,” it means that there was no network traffic to or from the network interface during the aggregation interval. If the log-status field has a value of “SKIPDATA,” then some flow log records were skipped during the aggregation interval because of a capacity constraint or an error.



More VPC troubleshooting tools

Reachability Analyzer

- Test connectivity between a source and destination resources in a VPC.

Network Access Analyzer

- Identify unintended network access to your resources on AWS.

Traffic Mirroring

- Make a copy of network traffic to send to security and monitoring appliances.

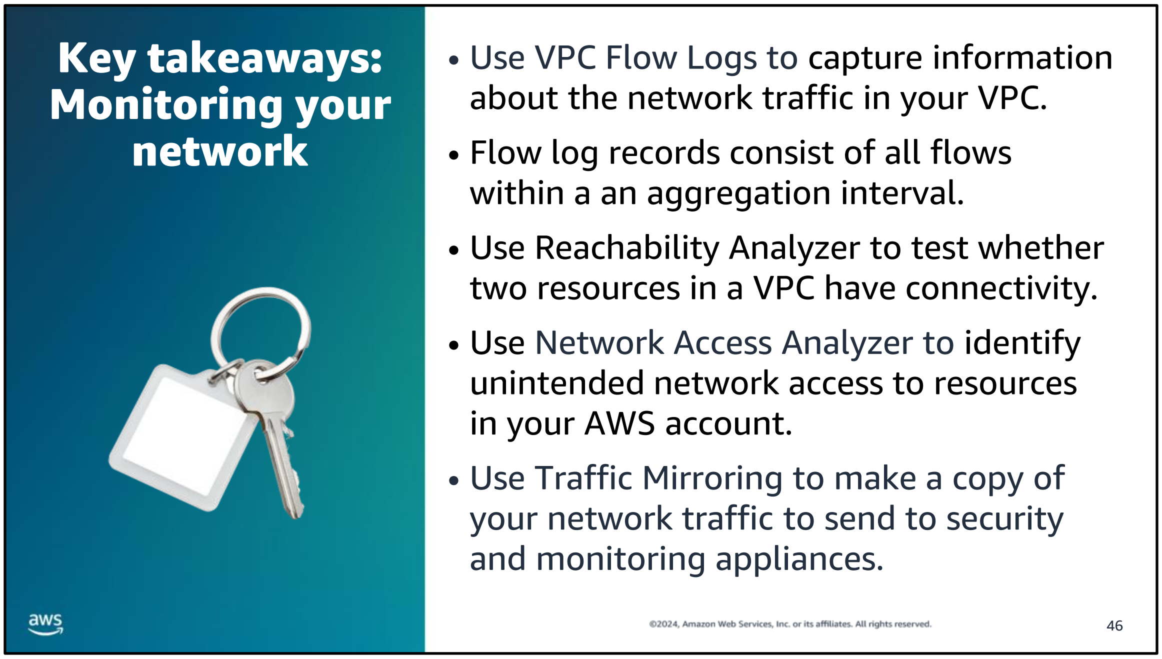
45

To solve specific types of network issues, Amazon VPC provides additional tools to help you resolve these issues.

To test whether two resources in a VPC have connectivity, you can use the Reachability Analyzer feature. When the destination is reachable, Reachability Analyzer produces hop-by-hop details of the virtual network path between the source and the destination. When the destination is not reachable, Reachability Analyzer identifies the blocking component. For example, paths can be blocked by configuration issues in a security group, network access control list (network ACL), route table, or load balancer. To use Reachability Analyzer, you specify the path for the traffic from a source to a destination. For example, you can specify an internet gateway as the source, an EC2 instance as the destination, 22 as the destination port, and TCP as the protocol. This allows you to verify that you can connect to the EC2 instance through the internet gateway using SSH.

To identify unintended network access to resources in your AWS account, you can use the Network Access Analyzer feature. You can use Network Access Analyzer to understand, verify, and improve your network security posture by eliminating unintended network access. It’s also useful to demonstrate that your network on AWS meets your compliance requirements. For example, you can verify that a separate logical network is used for systems that process credit card information and that it’s isolated from the rest of your AWS environment.

To inspect and monitor traffic, you can use the Amazon VPC Traffic Mirroring feature. It allows you to make a copy of your network traffic to send to your security and monitoring appliances. This provides deeper insight into network traffic by allowing you to analyze actual traffic content including the actual message payload. It’s useful for use-cases when you need to analyze the actual packets to determine the root cause a performance issue, reverse-engineer a sophisticated network attack, or detect and stop insider abuse or compromised workloads. For example, you might want to mirror inbound TCP and UDP traffic to separate security appliances for packet inspection.



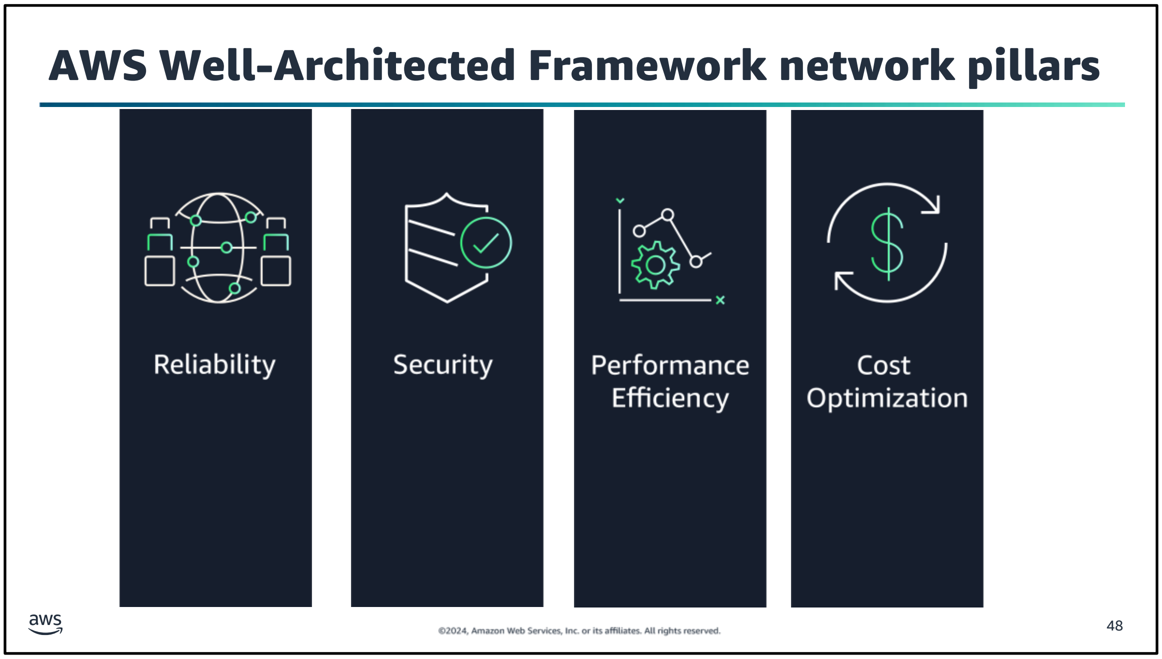
Key takeaways: Monitoring your network

- Use VPC Flow Logs to capture information about the network traffic in your VPC.
- Flow log records consist of all flows within a an aggregation interval.
- Use Reachability Analyzer to test whether two resources in a VPC have connectivity.
- Use Network Access Analyzer to identify unintended network access to resources in your AWS account.
- Use Traffic Mirroring to make a copy of your network traffic to send to security and monitoring appliances.

Applying Well-Architected Framework principles to a network

Creating a Networking Environment

This section goes over how to apply the AWS Well-Architected Framework principles to your VPC network.



AWS Well-Architected Framework network pillars

Reliability

Security

Performance Efficiency

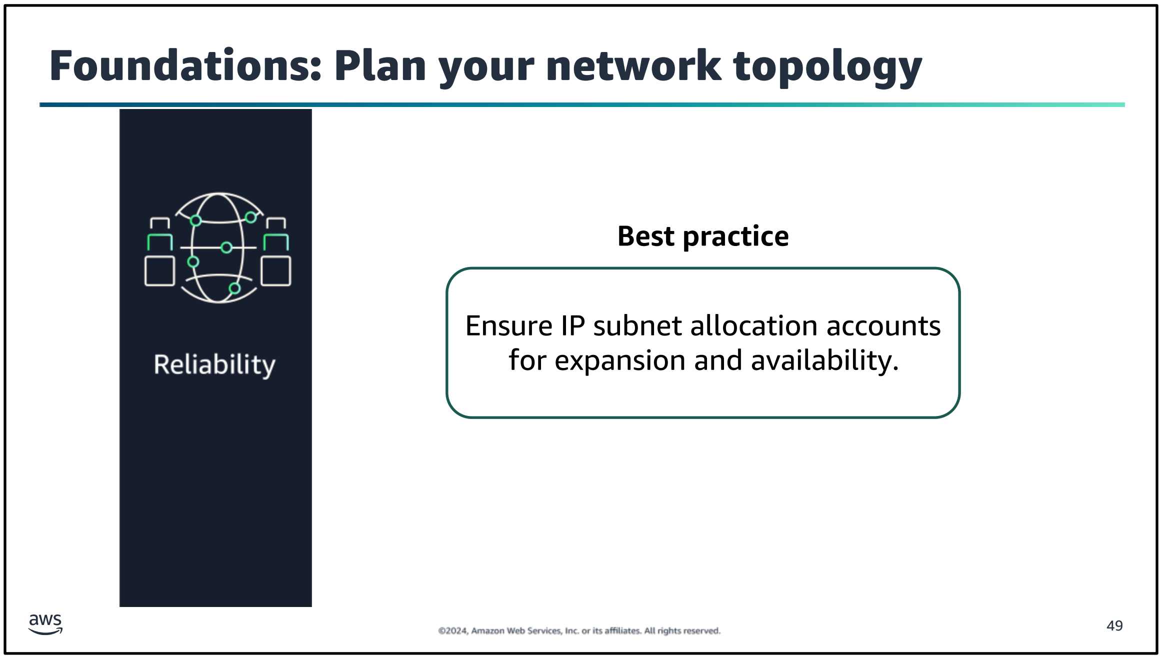
Cost Optimization

48

When you design a VPC network, you should consider all the future planned workloads in the VPC. How do you ensure that a workload will be resilient, secure, performant, and cost effective when deployed in a VPC? Your network design supports the workloads by meeting the same standards. Your VPC network should also be resilient, secure, performant, and cost effective.

The AWS Well-Architected Framework supplies best practices for workload design, operation, and maintenance. It helps you understand the pros and cons of the decisions that you make while building workloads on AWS. It's a set of foundational questions that help you to understand whether a specific architecture aligns well with cloud best practices.

The next few slides highlight some of the best practices for designing and maintaining your network using the Reliability, Security, Performance Efficiency, and Cost Optimization pillars. Some of these might seem familiar because they were covered earlier in the module.



Foundations: Plan your network topology

Reliability

Best practice

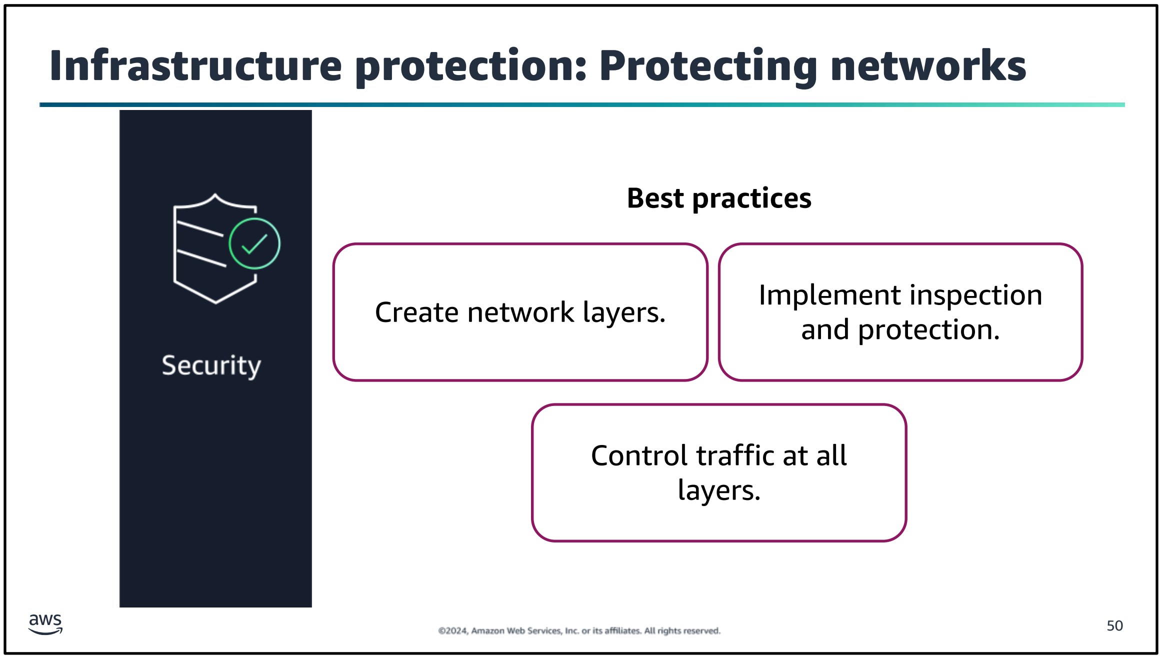
Ensure IP subnet allocation accounts for expansion and availability.

49

Resiliency forms part of the AWS Well-Architected Framework reliability pillar. The way to think about the reliability pillar is that it's used to measure the ability of a workload to perform its intended function correctly and consistently when it's expected to. That implies that your network supporting the workload should also function correctly and consistently when it's expected to. Your network should anticipate possible network failures and accommodate future traffic growth. In AWS, all workloads run on the AWS network with some workloads running in your VPC. So resiliency is a shared responsibility between AWS and you. AWS is responsible for the resiliency of the cloud infrastructure, while you are responsible for implementing resiliency in the cloud.

Ensure IP subnet allocation accounts for expansion and availability. When allocating a VPC address CIDR block, AWS recommends to follow the following best practices:

- Within a VPC, allow CIDR block space for multiple subnets that span multiple Availability Zones (AZs).
- Always allocate unused CIDR block space within a VPC for future expansion.
- Take into account that each subnet CIDR block has five reserved IP addresses for AWS use.
- Be aware of services that allocate additional IP addresses, such as container services.
- Deploy large VPC CIDR blocks because a CIDR block can't be changed or deleted after creation. You can add additional non-overlapping CIDR blocks to the VPC.
- Plan your subnet CIDR block ranges carefully as subnet IPv4 CIDRs can not be changed.



Infrastructure protection: Protecting networks

Security

Best practices

Create network layers.

Implement inspection and protection.

Control traffic at all layers.

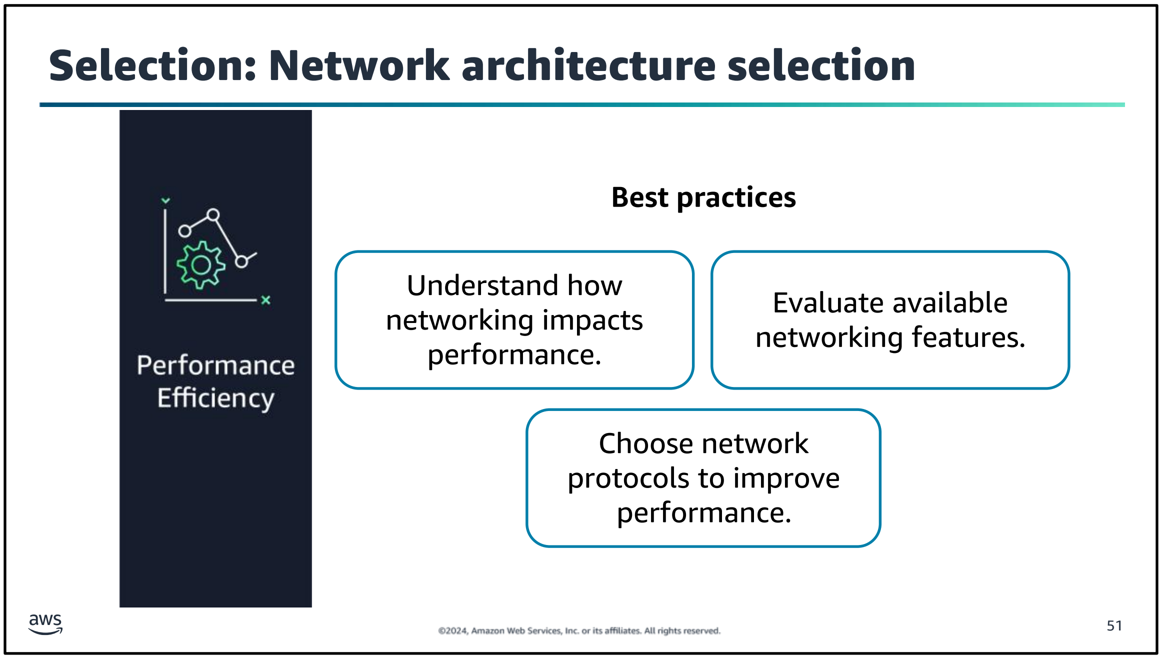
50

The users that use the workloads in your network can be located anywhere. In traditional security models, there tend to be too many access permissions to a network. AWS recommends to apply a Zero Trust approach to apply the principle of security at all layers. Zero Trust security is a model where application components are considered separate from each other and no component trusts any other component. The careful planning and management of your network design forms the foundation of how you provide isolation and boundaries for resources within your workload. Because many resources in your workload operate in a VPC and inherit the security properties, it's critical that the design is supported with inspection and protection mechanisms.

Create network layers: AWS recommends to logically group workload components that share the same sensitivity requirement together in a layer. For example, a database in a VPC with no need for internet access should be placed in subnets with no route to or from the internet. Traffic should only flow from the adjacent next least sensitive resource.

Control traffic at all layers: AWS recommends to apply multiple security controls with a defense in depth approach for inbound and outbound traffic. Define allowed traffic paths in the VPC with security groups, network ACLs, subnets, route tables, internet gateways, and NAT gateways.

Implement inspection and protection: AWS recommends to inspect and filter your traffic at each layer. For example, you can inspect your VPC configurations for potential unintended access using the VPC Network Access Analyzer.



Selection: Network architecture selection

Performance Efficiency

Best practices

Understand how networking impacts performance.

Evaluate available networking features.

Choose network protocols to improve performance.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

51

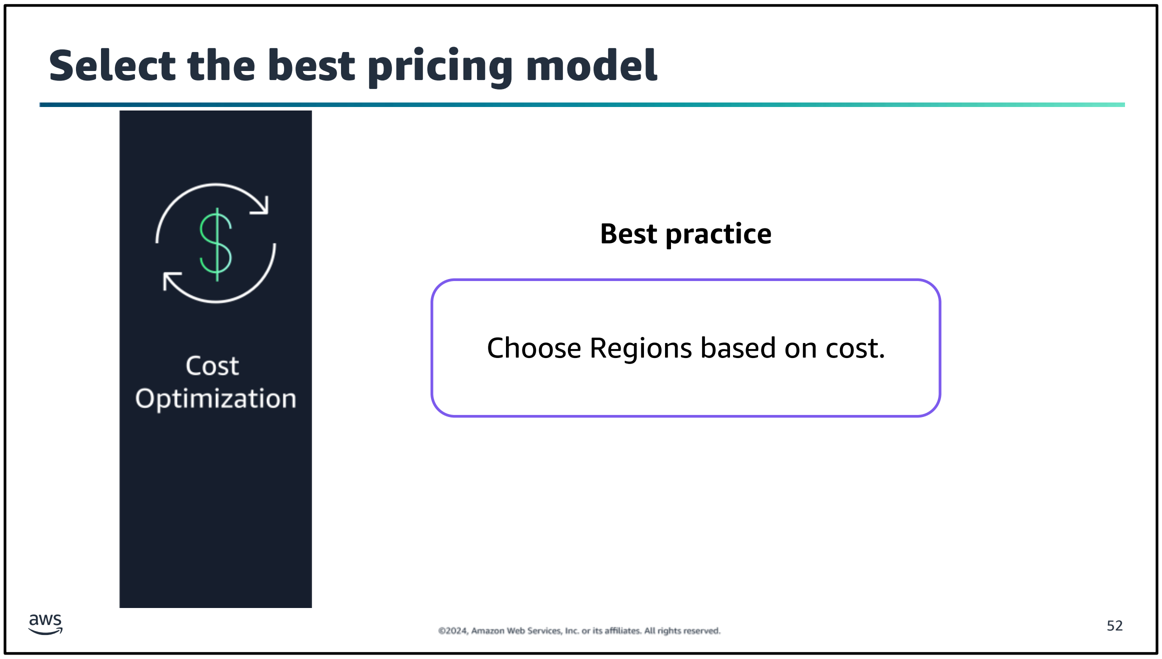
The performance efficiency pillar addresses best practices for managing production environments. You want to avoid scenarios where a workload on a network uses too much network bandwidth. Each workload that you deploy in a VPC has different requirements for latency, throughput, amount of jitter allowed, and bandwidth. Jitter occurs when there is a time delay caused by network congestion or route changes.

The following performance efficiency pillar best practices can help you to optimize your network design:

Understand how networking impacts performance: Analyze and understand how network-related decisions impact workload performance. For example, implementing database synchronous replication between AWS Regions is not possible due to double digit or more millisecond latency between Regions. Because latency between AZs are measured in single digit milliseconds, you can implement database synchronous replication between AZs.

Evaluate available networking features: Benchmark your workload performance metrics, including network metrics. You should continually evaluate the workload for areas to improve, such as removing performance bottlenecks. You can use Network Access Analyzer to help identify network paths and routes.

Choose network protocols to improve performance: Choose network protocols that will optimize your workload's performance. For example, do not use TCP for all workloads regardless of performance requirements. As an alternative, use both TCP and UDP together for Virtual Desktop Infrastructure workloads. This can take advantage of the reliability of TCP for critical data and the speed of UDP for real-time data.



Select the best pricing model

Cost Optimization

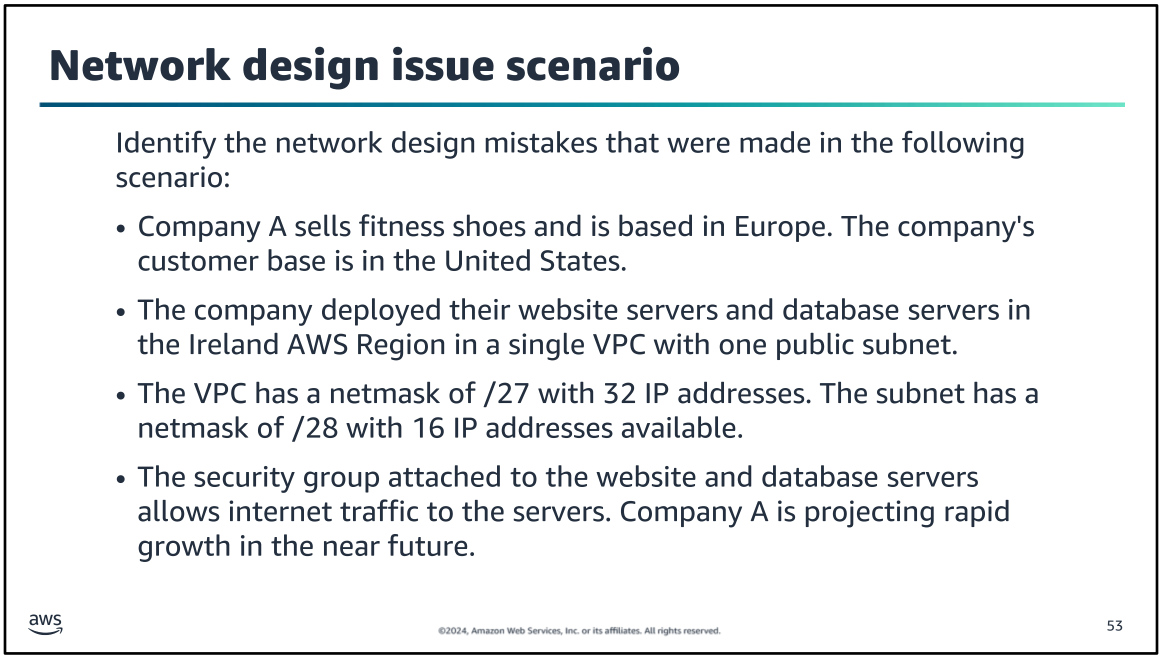
Best practice

Choose Regions based on cost.

52

A workload on a network should fully utilize all resources, achieve outcomes at the lowest possible price point, and meet functional requirements. Network costs should be included in the workload cost benchmark. The key to save costs is to select the best pricing model with appropriate network configurations for your workloads.

Implement Regions based on cost: AWS recommends using the AWS Region that delivers the best overall global cost solution. Each AWS Region operates within local market conditions. Resource pricing is different in each Region due to differences such as the cost of land, fiber, electricity, and taxes. Choose a specific Region to operate a component of your entire solution so that you can run at the lowest possible price globally. When you architect your solutions, a best practice is to seek to place computing resources closer to users to provide lower latency and strong data sovereignty. Select the geographic location based on your business, data privacy, performance, and security requirements. For applications with global end users, use multiple locations.



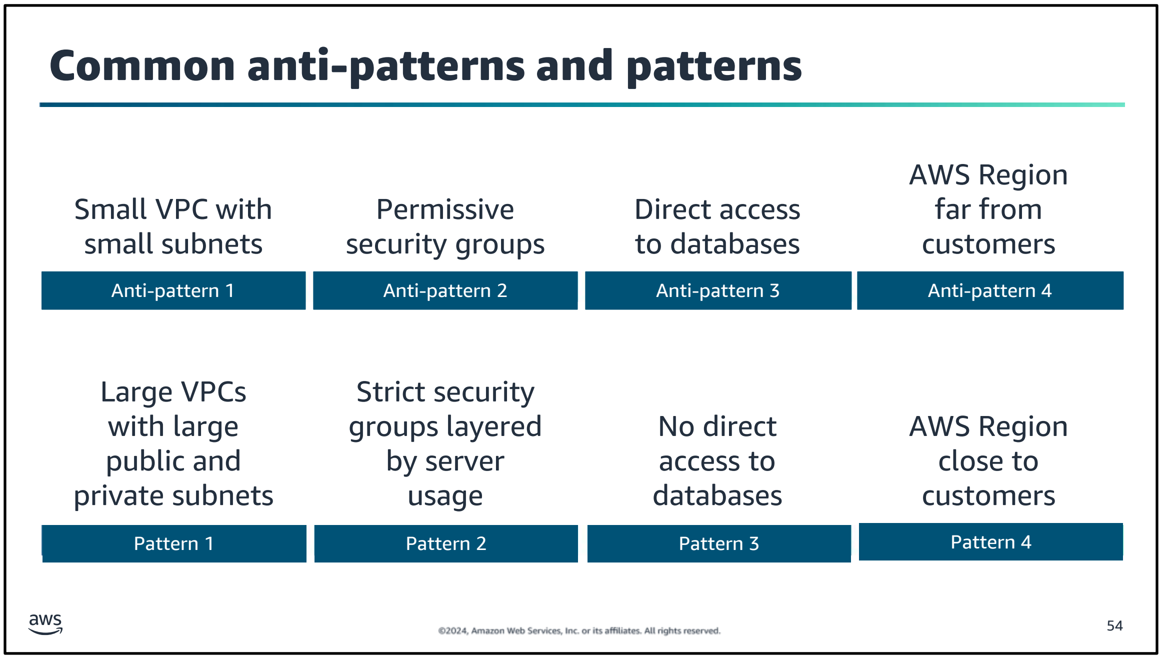
Network design issue scenario

Identify the network design mistakes that were made in the following scenario:

- Company A sells fitness shoes and is based in Europe. The company's customer base is in the United States.
- The company deployed their website servers and database servers in the Ireland AWS Region in a single VPC with one public subnet.
- The VPC has a netmask of /27 with 32 IP addresses. The subnet has a netmask of /28 with 16 IP addresses available.
- The security group attached to the website and database servers allows internet traffic to the servers. Company A is projecting rapid growth in the near future.

53

The goal is for you to identify design issues or anti-patterns and provide a better alternative (pattern). Anti-patterns are bad designs that bring inherent problems to a solution. They are the opposite of best practice patterns and should be avoided. The AWS Well-Architected Framework describes anti-patterns in order to contrast best practices.



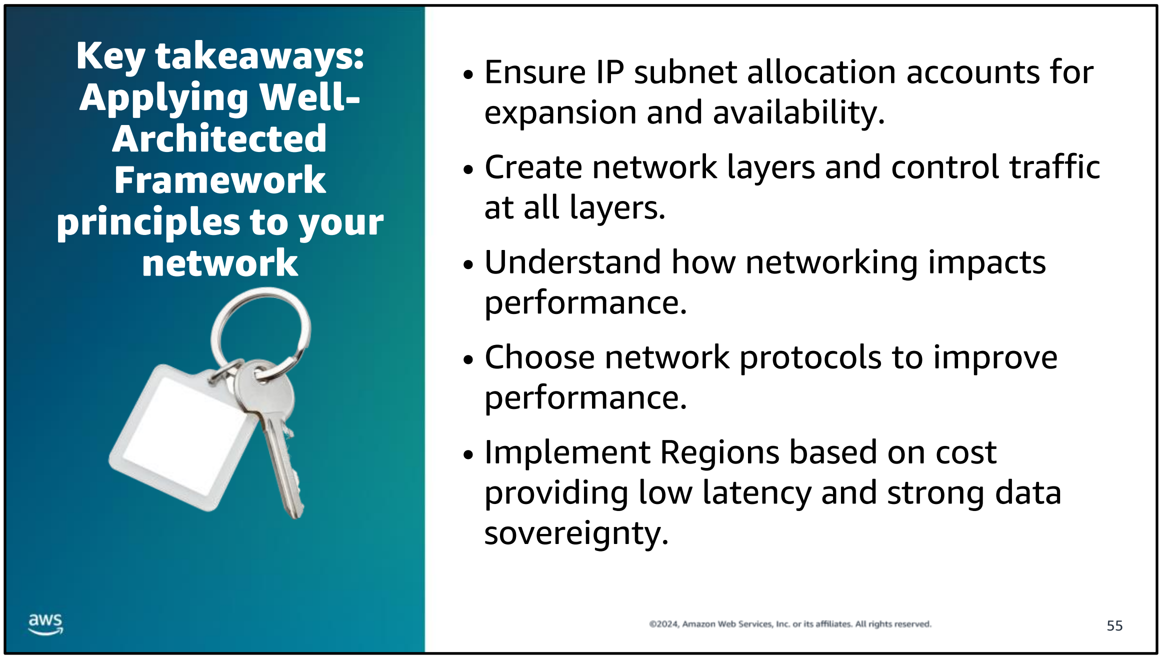
Common anti-patterns and patterns

Small VPC with small subnets	Permissive security groups	Direct access to databases	AWS Region far from customers
Anti-pattern 1	Anti-pattern 2	Anti-pattern 3	Anti-pattern 4
Large VPCs with large public and private subnets	Strict security groups layered by server usage	No direct access to databases	AWS Region close to customers
Pattern 1	Pattern 2	Pattern 3	Pattern 4

54

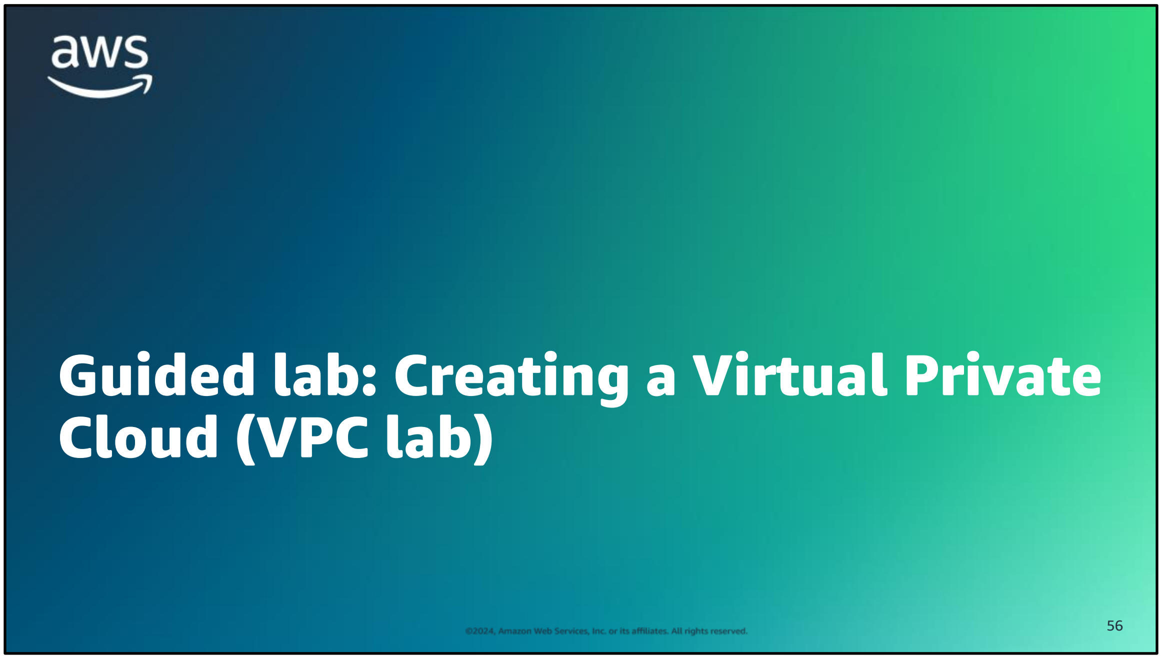
There are four major network issues identified in the scenario. They align to the anti-patterns that are shown on this slide.

- Company A should not be using a single small VPC with limited IP addresses, but rather large VPCs with enough IP addresses for future growth.
- The website servers should have their own security group with internet access allowed in a public subnet. The database servers should have their own security group allowing access from the website server security group and database support.
- Database servers should be placed in a private subnet. No internet direct access to the database servers should be allowed. Maintenance access can be configured in the database security group and allow server patching with access to a NAT gateway.
- Instead of using an AWS Region in Europe, an AWS Region in the US closer to the customer base should be used for lower latency and strong data sovereignty.



Key takeaways: Applying Well-Architected Framework principles to your network

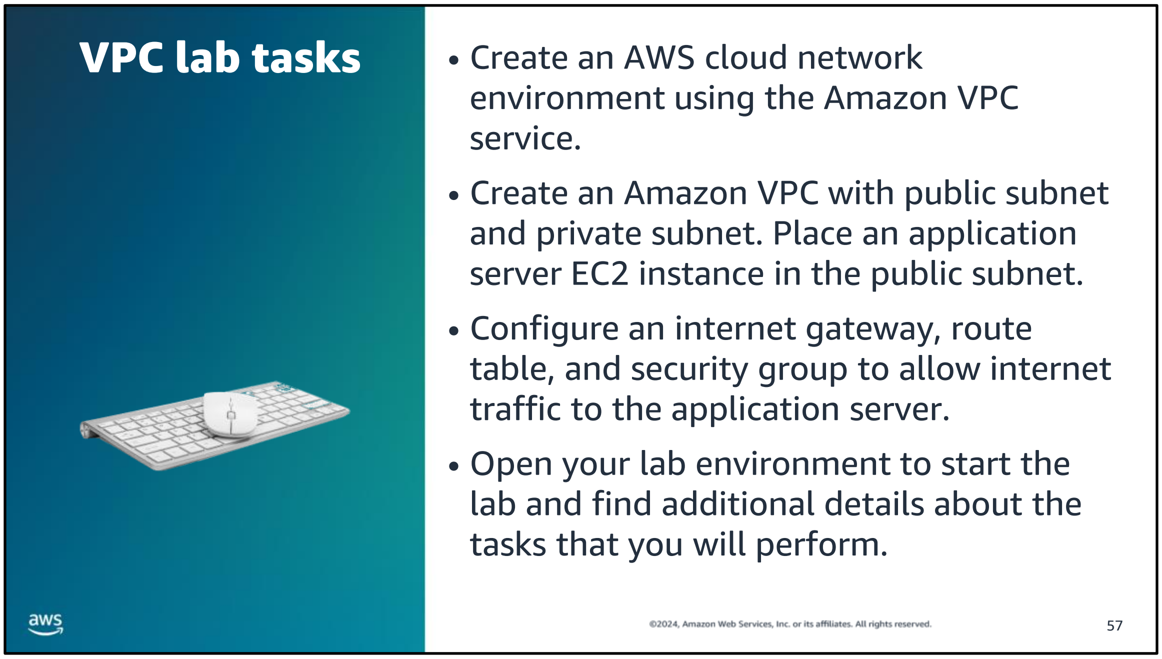
- Ensure IP subnet allocation accounts for expansion and availability.
- Create network layers and control traffic at all layers.
- Understand how networking impacts performance.
- Choose network protocols to improve performance.
- Implement Regions based on cost providing low latency and strong data sovereignty.



Guided lab: Creating a Virtual Private Cloud (VPC lab)

56

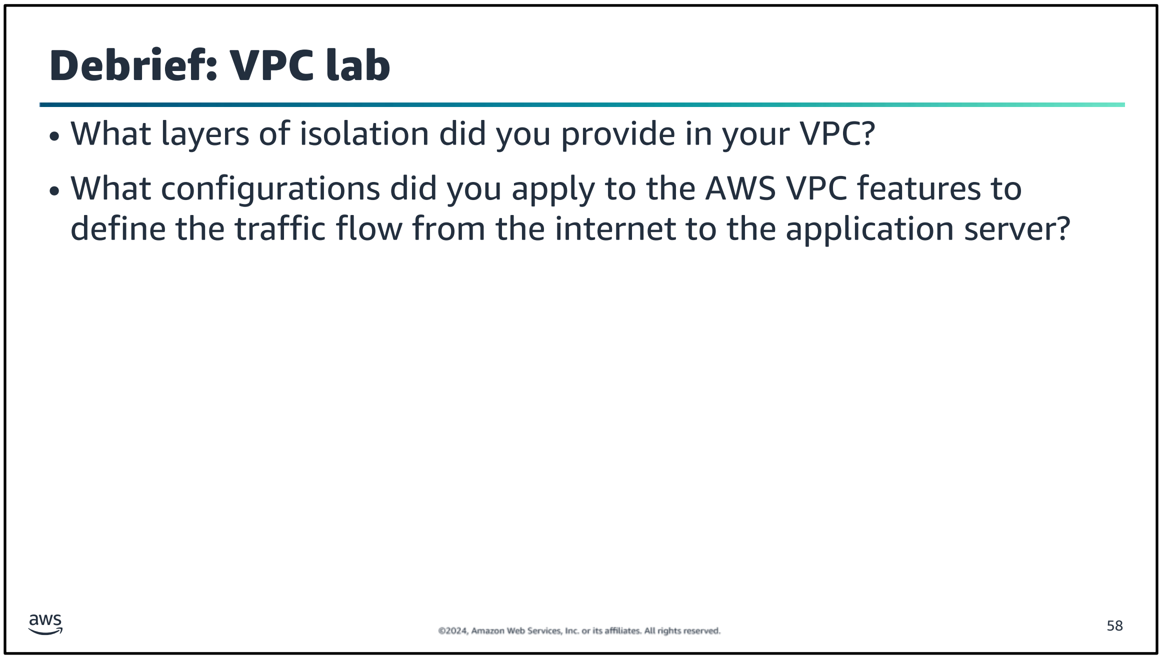
You will now complete a guided lab. The next slide summarizes what you will do in the lab, and you will find the detailed instructions in the lab environment.



VPC lab tasks

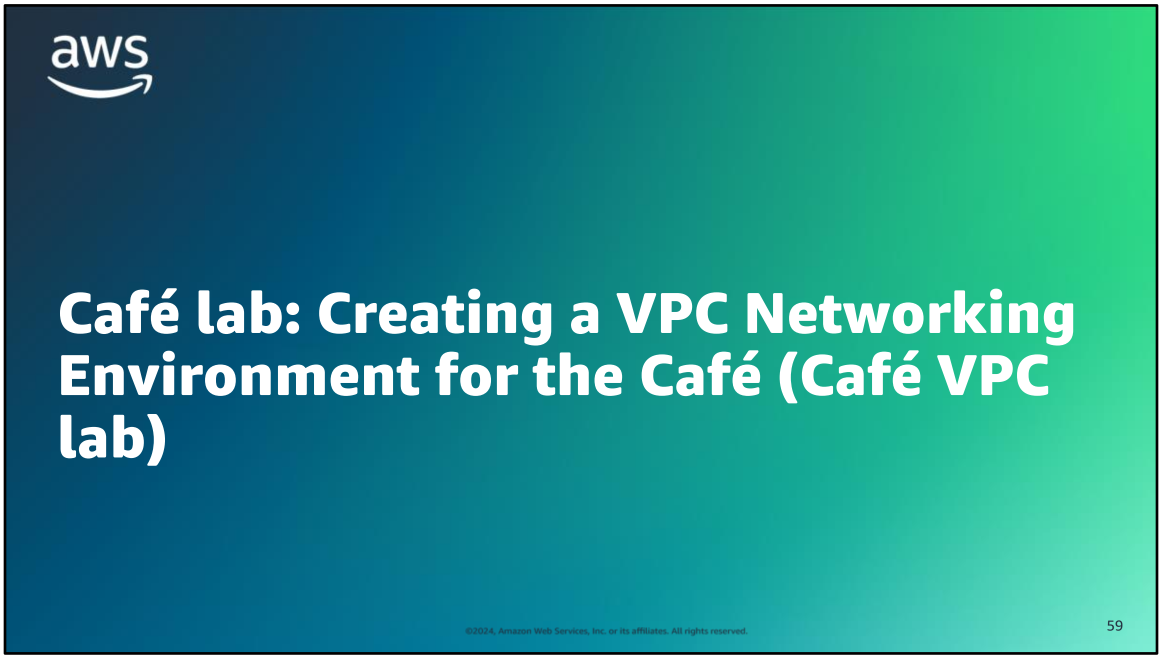
- Create an AWS cloud network environment using the Amazon VPC service.
- Create an Amazon VPC with public subnet and private subnet. Place an application server EC2 instance in the public subnet.
- Configure an internet gateway, route table, and security group to allow internet traffic to the application server.
- Open your lab environment to start the lab and find additional details about the tasks that you will perform.

Access the lab environment through your online course to get additional details and complete the lab.



Debrief: VPC lab

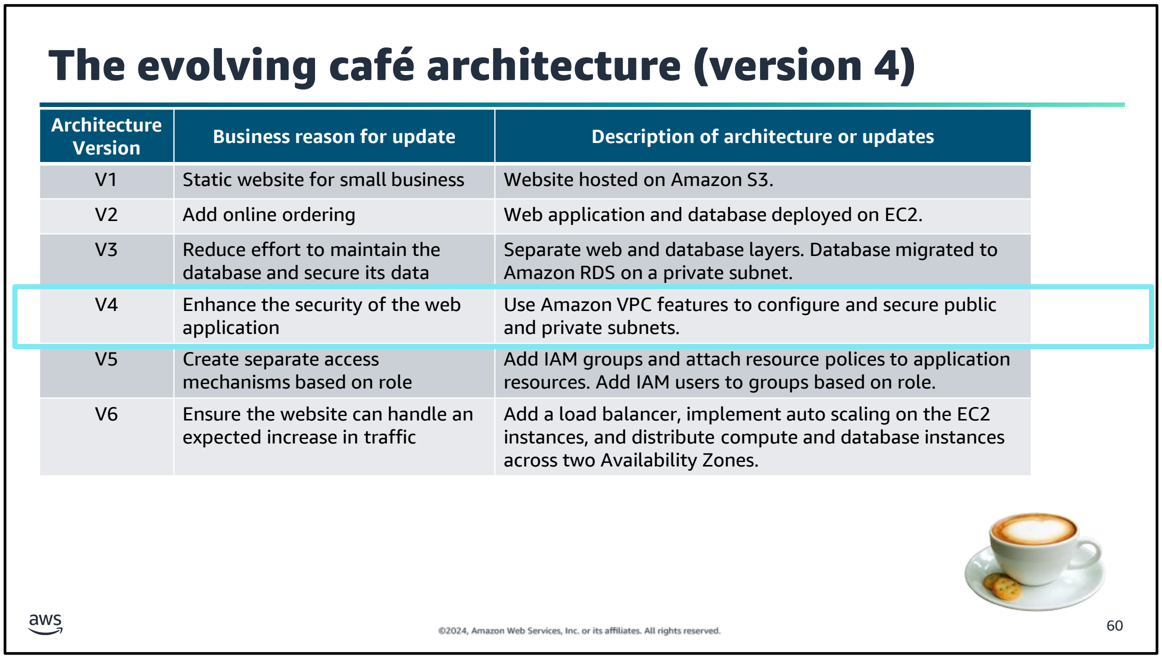
- What layers of isolation did you provide in your VPC?
- What configurations did you apply to the AWS VPC features to define the traffic flow from the internet to the application server?



Café lab: Creating a VPC Networking Environment for the Café (Café VPC lab)

59

You will now complete a challenge lab. The next slide summarizes what you will do in the lab, and you will find the detailed instructions in the lab environment.



The evolving café architecture (version 4)

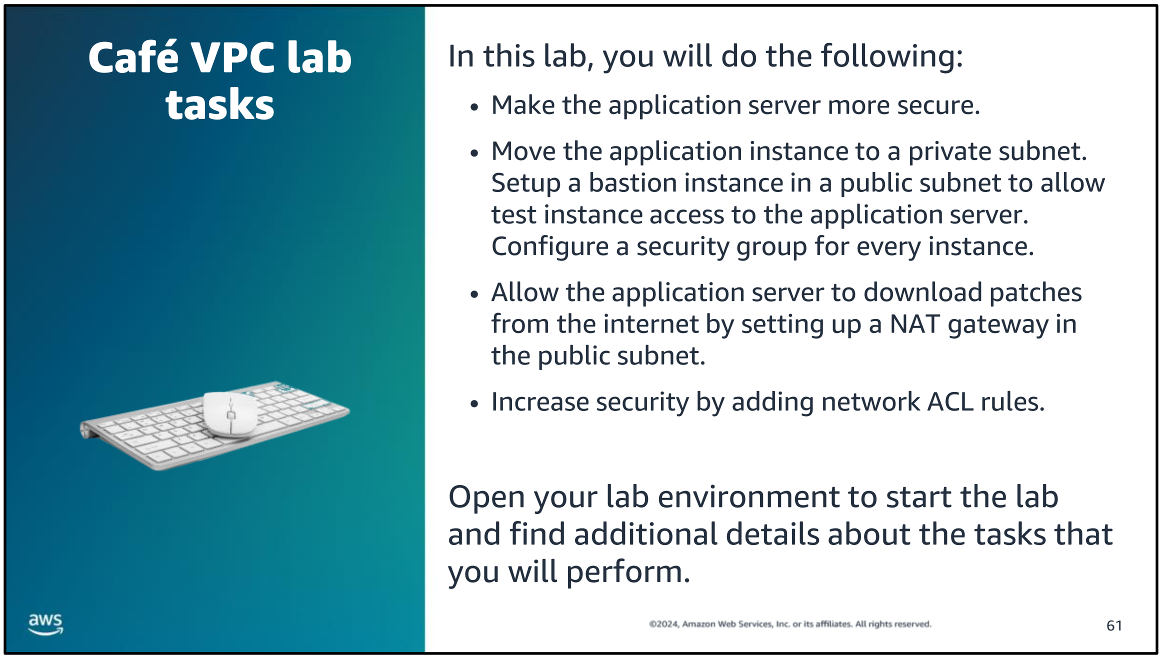
Architecture Version	Business reason for update	Description of architecture or updates
V1	Static website for small business	Website hosted on Amazon S3.
V2	Add online ordering	Web application and database deployed on EC2.
V3	Reduce effort to maintain the database and secure its data	Separate web and database layers. Database migrated to Amazon RDS on a private subnet.
V4	Enhance the security of the web application	Use Amazon VPC features to configure and secure public and private subnets.
V5	Create separate access mechanisms based on role	Add IAM groups and attach resource polices to application resources. Add IAM users to groups based on role.
V6	Ensure the website can handle an expected increase in traffic	Add a load balancer, implement auto scaling on the EC2 instances, and distribute compute and database instances across two Availability Zones.

60

The café's business has been steadily increasing. Sofía and Nikhil have become friends with a few of the café regulars, who are AWS consultants, and they started to discuss the café's current architecture. Olivia, one of the regulars and an AWS Solutions Architect, identified a need for the café's online business to scale. Scaling will require additional servers to run the online ordering application, but the current subnet size is too small and can't support this growth. Therefore, they will need to rearchitect some aspects of the network that the application runs in.

On further review of the café's architecture, Olivia also noticed a vulnerability: the TCP port that's used to administer the application server is accessible to the internet. Sofía explained that she and Nikhil must be able to manage and maintain the server. Olivia advises them to set up a bastion host to reduce public access to the server and to make it more secure.

In this lab, you will update the café architecture to enhance the security of the web application using Amazon VPC features to configure private and public subnets.



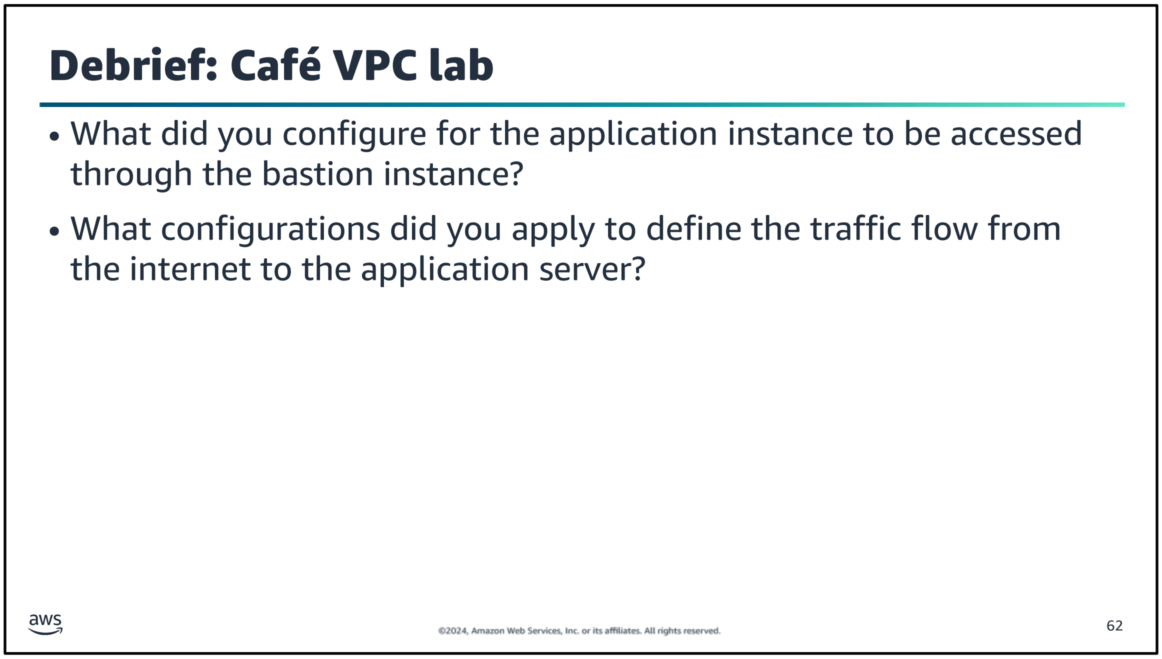
Café VPC lab tasks

In this lab, you will do the following:

- Make the application server more secure.
- Move the application instance to a private subnet. Setup a bastion instance in a public subnet to allow test instance access to the application server. Configure a security group for every instance.
- Allow the application server to download patches from the internet by setting up a NAT gateway in the public subnet.
- Increase security by adding network ACL rules.

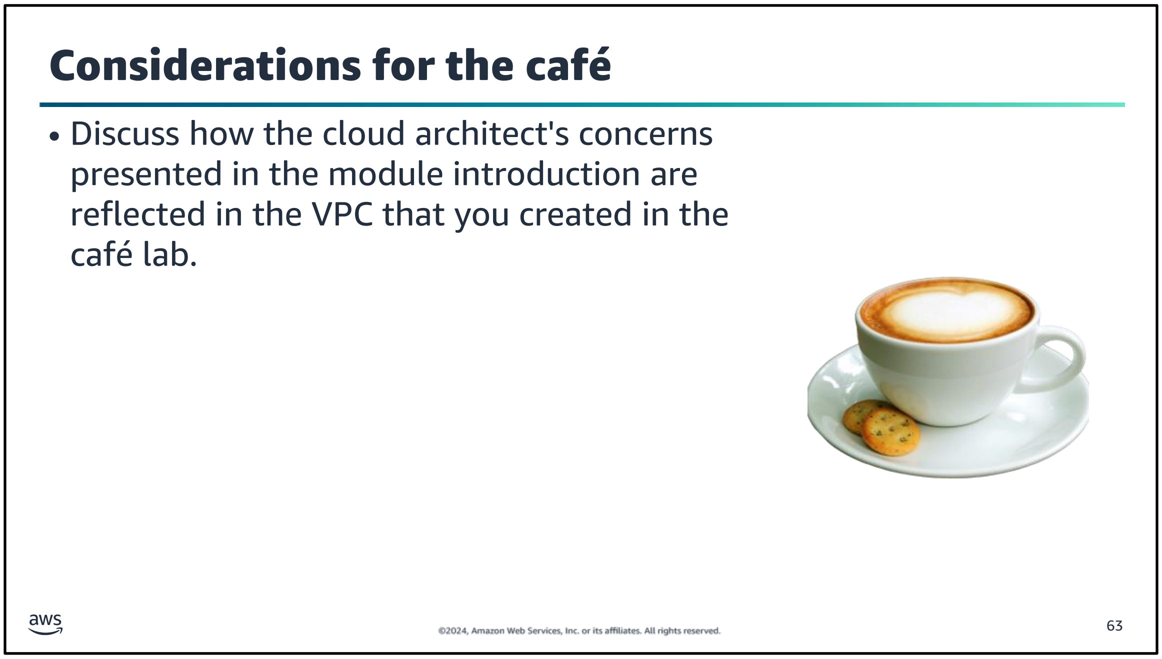
Open your lab environment to start the lab and find additional details about the tasks that you will perform.

Access the lab environment through your online course to get additional details and complete the lab.



Debrief: Café VPC lab

- What did you configure for the application instance to be accessed through the bastion instance?
- What configurations did you apply to define the traffic flow from the internet to the application server?



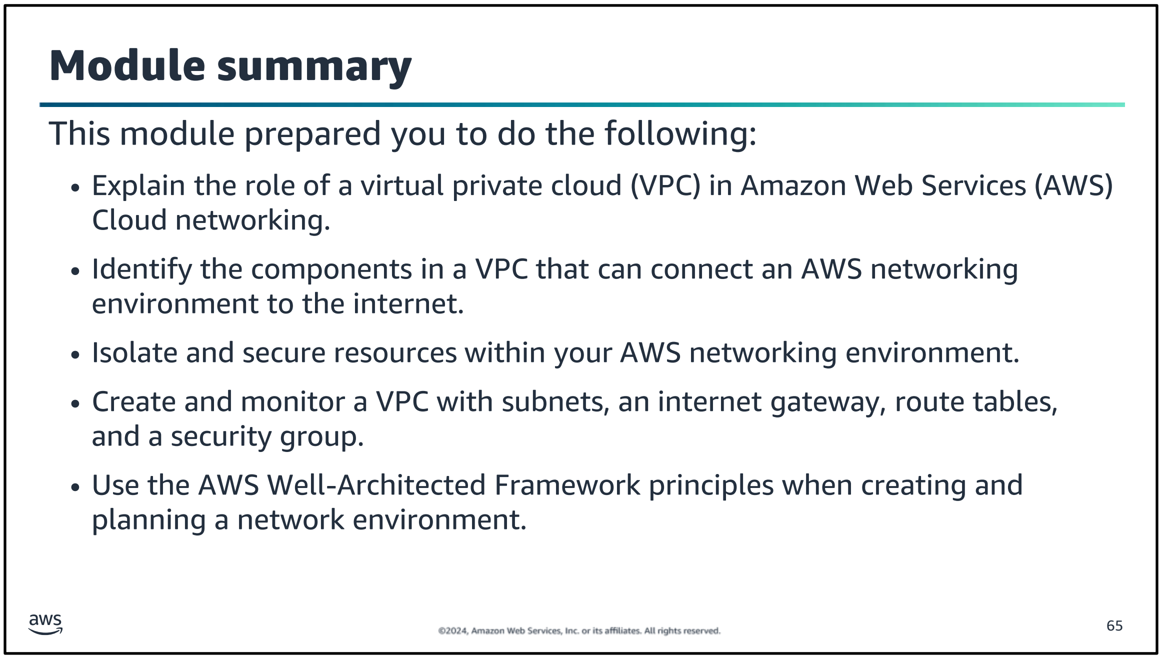
Considerations for the café

- Discuss how the cloud architect's concerns presented in the module introduction are reflected in the VPC that you created in the café lab.

Module wrap-up

Creating a Networking Environment

This section summarizes what you learned and brings the module to a close.

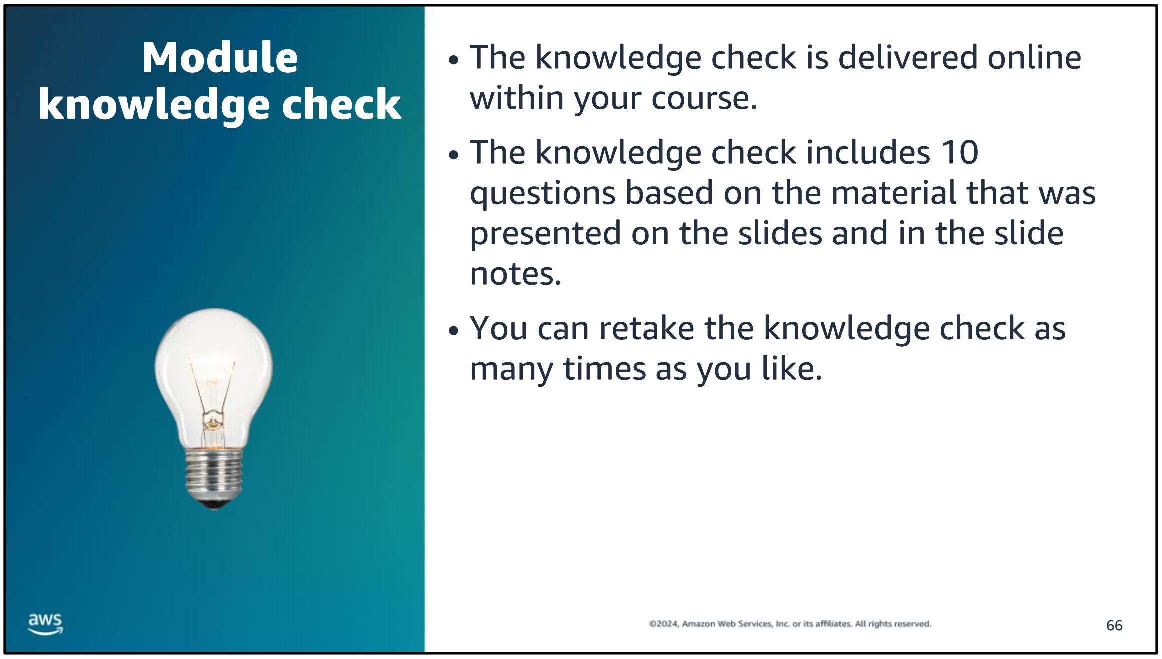


Module summary

This module prepared you to do the following:

- Explain the role of a virtual private cloud (VPC) in Amazon Web Services (AWS) Cloud networking.
- Identify the components in a VPC that can connect an AWS networking environment to the internet.
- Isolate and secure resources within your AWS networking environment.
- Create and monitor a VPC with subnets, an internet gateway, route tables, and a security group.
- Use the AWS Well-Architected Framework principles when creating and planning a network environment.

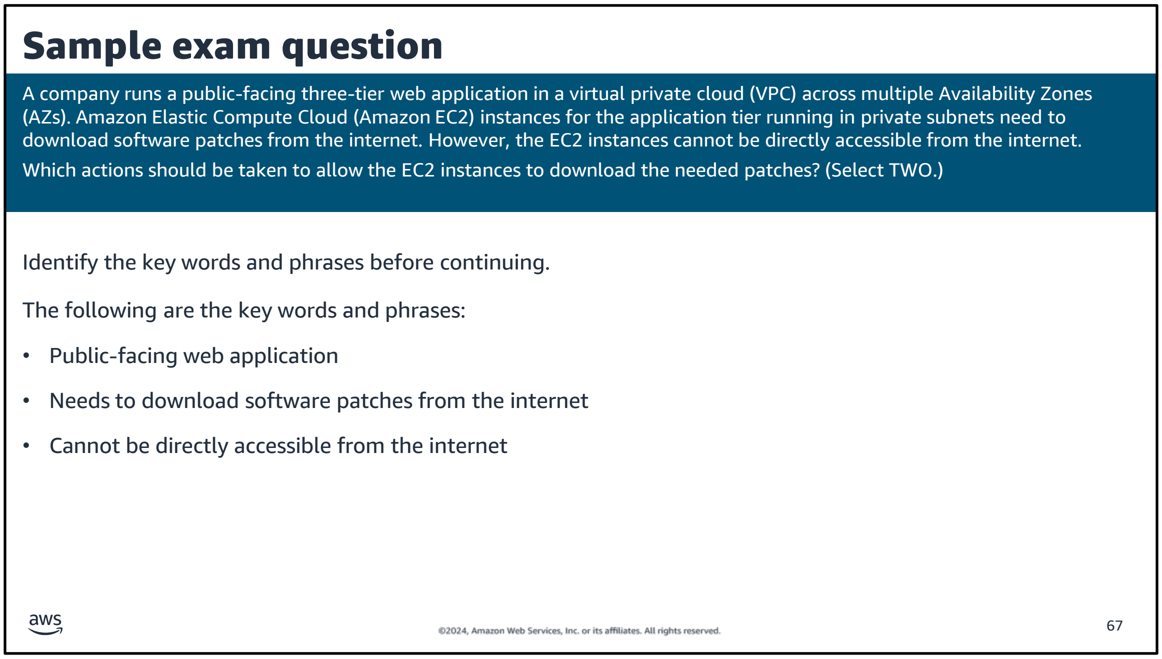
65



Module knowledge check

- The knowledge check is delivered online within your course.
- The knowledge check includes 10 questions based on the material that was presented on the slides and in the slide notes.
- You can retake the knowledge check as many times as you like.

Use your online course to access the knowledge check for this module.



Sample exam question

A company runs a public-facing three-tier web application in a virtual private cloud (VPC) across multiple Availability Zones (AZs). Amazon Elastic Compute Cloud (Amazon EC2) instances for the application tier running in private subnets need to download software patches from the internet. However, the EC2 instances cannot be directly accessible from the internet.

Which actions should be taken to allow the EC2 instances to download the needed patches? (Select TWO.)

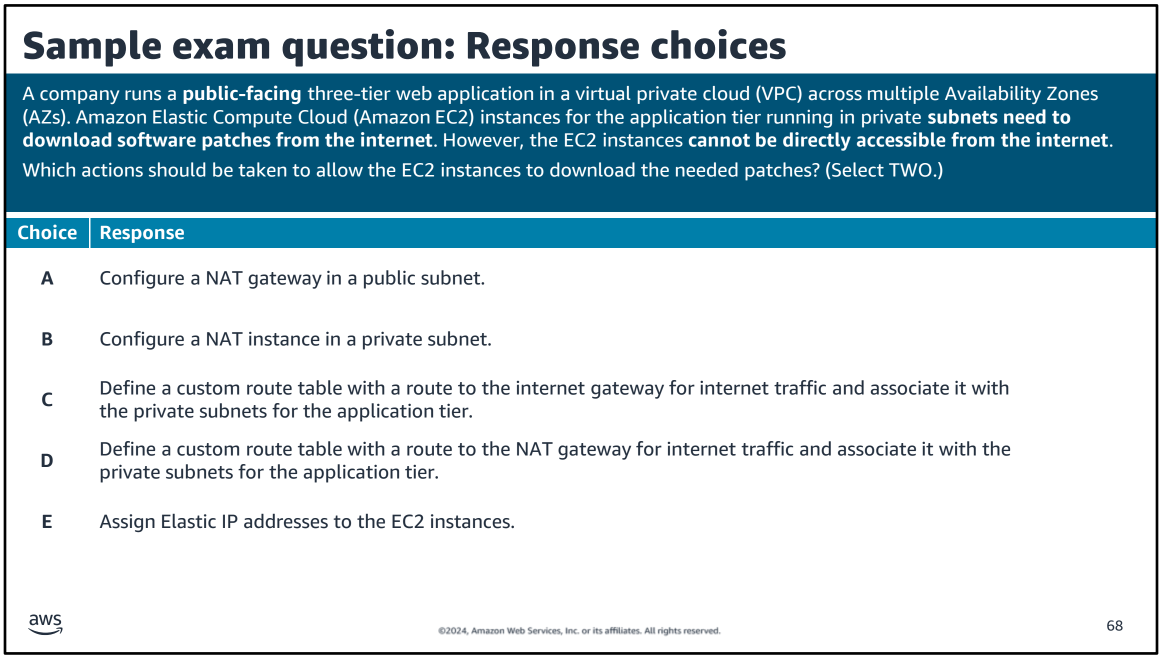
Identify the key words and phrases before continuing.

The following are the key words and phrases:

- Public-facing web application
- Needs to download software patches from the internet
- Cannot be directly accessible from the internet

67

Key elements of the question include the fact that the web application is public facing, patches must be downloaded from the internet, and the EC2 instances where the application runs cannot be directly accessible.



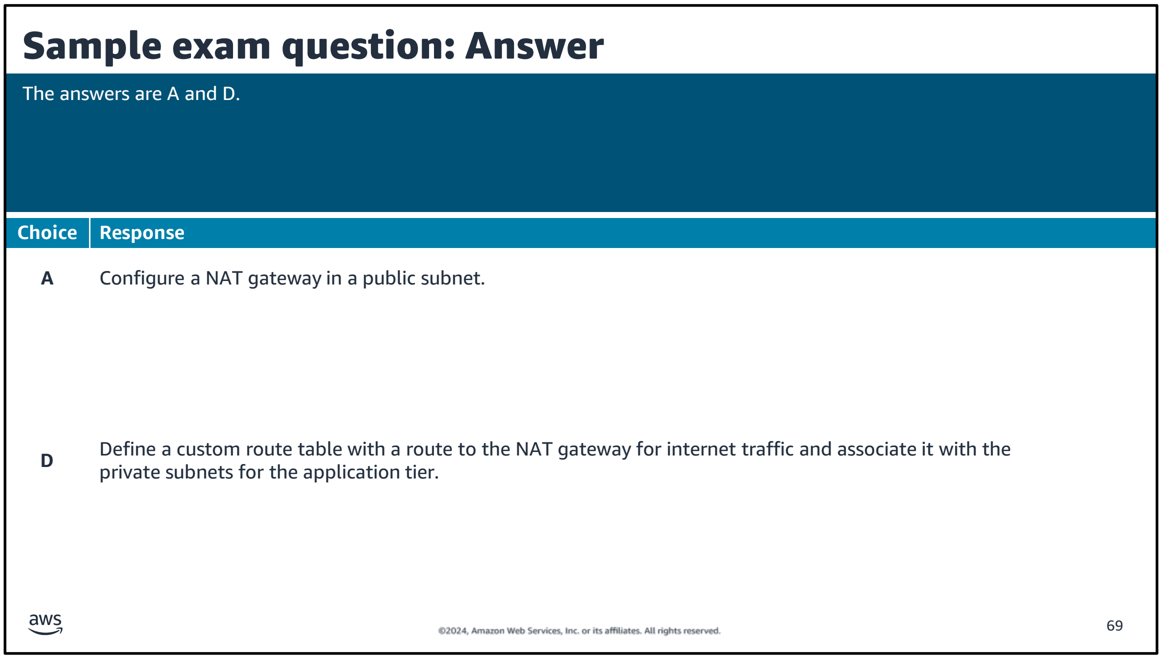
Sample exam question: Response choices

A company runs a **public-facing** three-tier web application in a virtual private cloud (VPC) across multiple Availability Zones (AZs). Amazon Elastic Compute Cloud (Amazon EC2) instances for the application tier running in private **subnets need to download software patches from the internet**. However, the EC2 instances **cannot be directly accessible from the internet**.

Which actions should be taken to allow the EC2 instances to download the needed patches? (Select TWO.)

Choice	Response
A	Configure a NAT gateway in a public subnet.
B	Configure a NAT instance in a private subnet.
C	Define a custom route table with a route to the internet gateway for internet traffic and associate it with the private subnets for the application tier.
D	Define a custom route table with a route to the NAT gateway for internet traffic and associate it with the private subnets for the application tier.
E	Assign Elastic IP addresses to the EC2 instances.

Use the key words that you identified on the previous slide and review each of the possible responses to determine which two best address the question.



Sample exam question: Answer

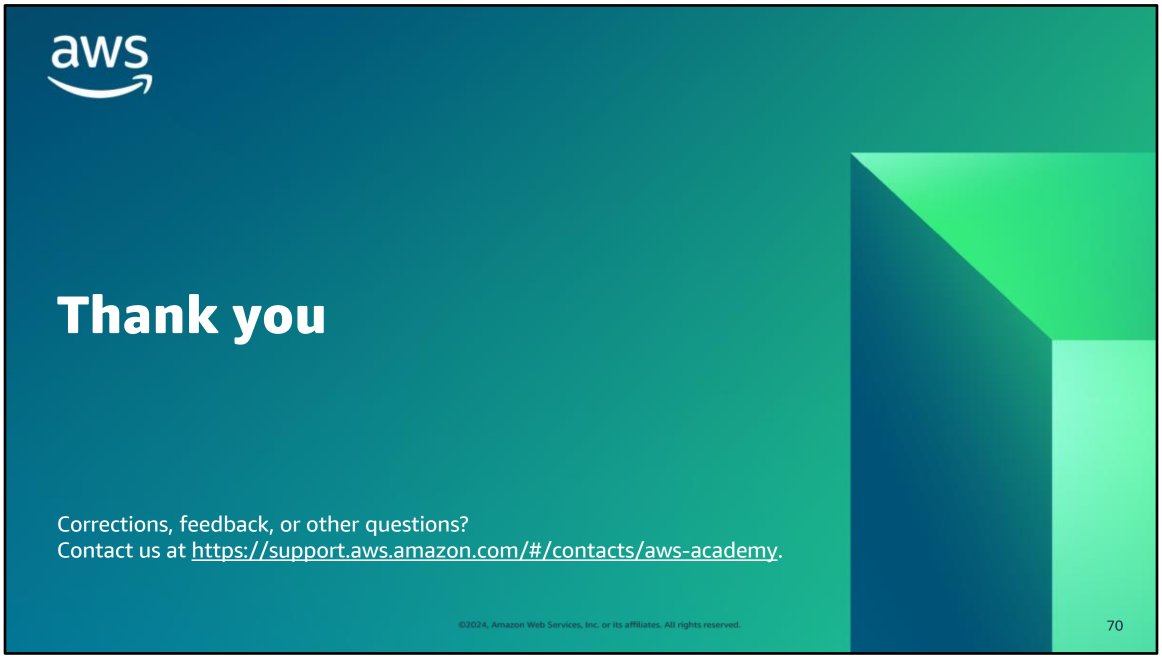
The answers are A and D.

Choice	Response
A	Configure a NAT gateway in a public subnet.
D	Define a custom route table with a route to the NAT gateway for internet traffic and associate it with the private subnets for the application tier.

Choices B, C and E would not be an appropriate way to connect the EC2 instances in the private subnet to the internet.

A NAT gateway in a public subnet forwards traffic from the EC2 instances in the private subnet to the internet or other AWS services, and then sends the response back to the instances.

After a NAT gateway is created, the route tables for private subnets must be updated to point internet traffic to the NAT gateway.



Thank you

Corrections, feedback, or other questions?
Contact us at https://support.aws.amazon.com/#/contacts/aws-academy.

70

That concludes this module. The Content Resources page of your course includes links to additional resources that are related to this module.