

AWS Cloud Security Foundations– AWS-CSF

Dr. Ibrahim Gomaa

Lecture 2 – AWS Identity, Networking and Content Delivery



Topics

- AWS shared responsibility model
- AWS Identity and Access Management (IAM)
- Networking basics
- Amazon VPC, Networking and Security
- Amazon Route 53
- Amazon CloudFront

Activities

- Label a network diagram

Demo

- VPC demonstration

Lab

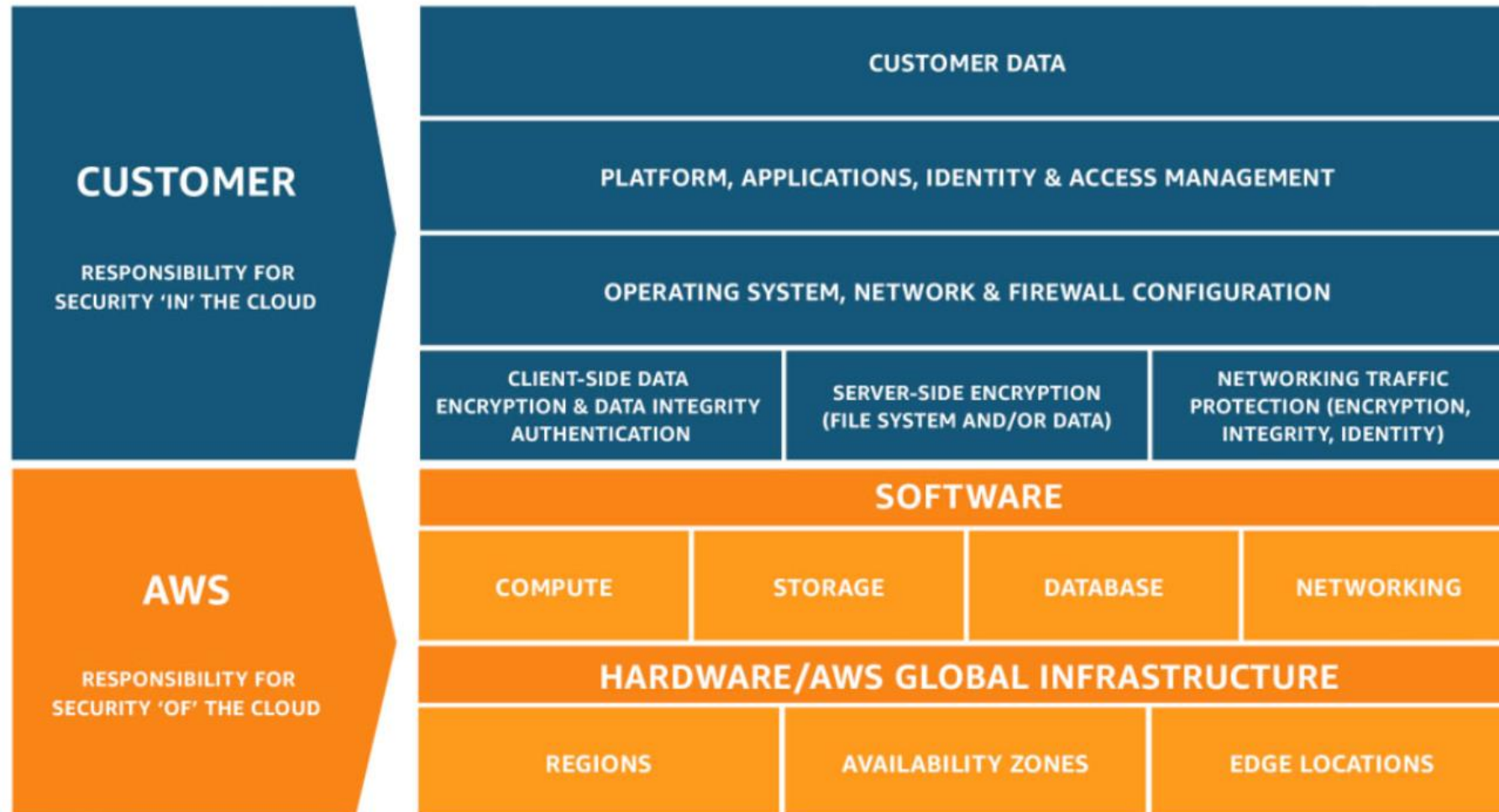
- Build your VPC and launch a web server



Knowledge check

AWS shared responsibility model





AWS responsibility: Security *of* the cloud

AWS services



Compute



Storage



Database



Networking

AWS Global Infrastructure



Regions

Availability Zones



Edge locations

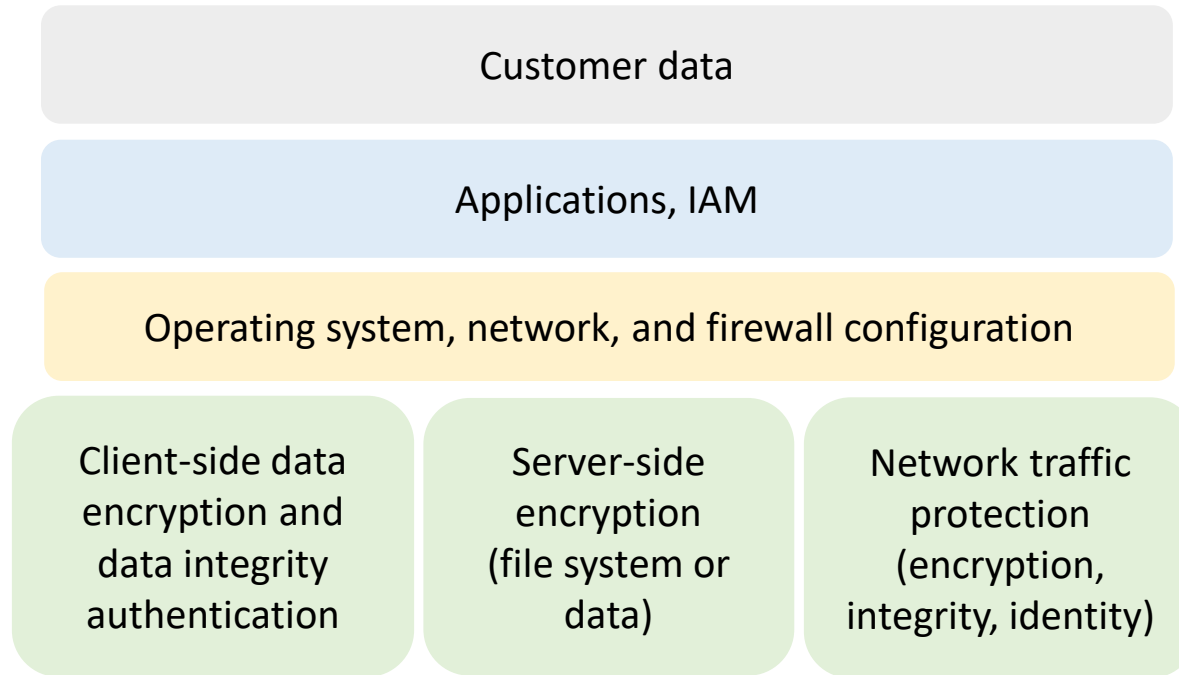
AWS operates, manages, and controls the components from the bare metal host operating system and hypervisor virtualization layer down to the physical security of the facilities where the services operate

AWS responsibilities:

- Physical security of data centers
 - Controlled, need-based access
- Hardware and software infrastructure
 - Storage decommissioning, host operating system (OS) access logging, and auditing
- Network infrastructure
 - Intrusion detection
- Virtualization infrastructure
 - Instance isolation



Customer responsibility: Security *in* the cloud



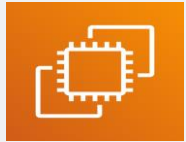
Customer-configurable

Customer responsibilities:

- Amazon Elastic Compute Cloud (Amazon EC2) instance **operating system**
 - Including patching, maintenance
- **Applications**
 - Passwords, role-based access, etc.
- **Security group** configuration
- OS or host-based **firewalls**
 - Including intrusion detection or prevention systems
- **Network** configurations
- Account management
 - Login and permission settings for each user

Service characteristics and security responsibility

Example services managed by the customer



Amazon
EC2



Amazon Elastic Block
Store (Amazon EBS)



Amazon
Virtual Private Cloud
(Amazon VPC)

Example services managed by AWS



AWS
Lambda



Amazon
Relational Database Service
(Amazon RDS)



AWS Elastic
Beanstalk

Infrastructure as a service (IaaS)

Customer has more flexibility over configuring networking and storage settings

Customer is responsible for managing more aspects of the security

Customer configures the access controls

Platform as a service (PaaS)

Customer does not need to manage the underlying infrastructure

AWS handles the operating system, database patching, firewall configuration, and disaster recovery

Customer can focus on managing code or data

Service characteristics and security responsibility (continued)

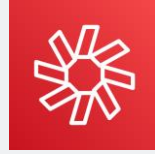
SaaS examples



AWS Trusted
Advisor



AWS Shield



Amazon Chime

AWS Trusted Advisor is an online tool that analyzes your AWS environment and provides real-time guidance and recommendations to help you provision your resources by following AWS best practices

Amazon Chime is a communications service that enables you to meet, chat, and place business calls inside and outside your organization

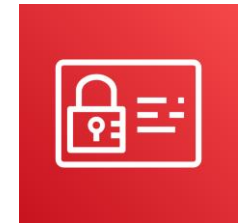
Software as a service (SaaS)

- Software is centrally hosted
- Licensed on a subscription model or pay-as-you-go basis.
- Services are typically accessed via web browser, mobile app, or application programming interface (API)
- Customers do not need to manage the infrastructure that supports the service

AWS Identity and Access Management (IAM)



- Use **IAM** to manage access to **AWS resources** –
 - A resource is an entity in an AWS account that you can work with
 - Example resources; An Amazon EC2 instance or an Amazon S3 bucket
- *Example* – Control who can terminate Amazon EC2 instances
- Define fine-grained access rights –
 - **Who** can access the resource
 - **Which** resources can be accessed and what can the user do to the resource
 - **How** resources can be accessed
- IAM is a no-cost AWS account feature



AWS Identity and Access
Management
(IAM)

IAM: Essential components



IAM user

A **person** or **application** that can authenticate with an AWS account.



IAM group

A **collection of IAM users** that are granted identical authorization.



IAM policy

The document that defines **which resources can be accessed** and the **level of access** to each resource.



IAM role

Useful mechanism to grant a set of permissions for making AWS service requests.

Authenticate as an IAM user to gain access

When you define an **IAM user**, you select what *types of access* the user is permitted to use.

Programmatic access

- Authenticate using:
 - Access key ID
 - Secret access key
- Provides AWS CLI and AWS SDK access



AWS CLI

AWS Tools
and SDKs

AWS Management Console access

- Authenticate using:
 - 12-digit Account ID or alias
 - IAM user name
 - IAM password
- If enabled, **multi-factor authentication (MFA)** prompts for an authentication code.



AWS Management Console



AWS Identity and Access Management (IAM)

IAM MFA

- MFA provides increased security.
- In addition to **user name** and **password**, MFA requires a unique **authentication code** to access AWS services.

Account:

User Name:

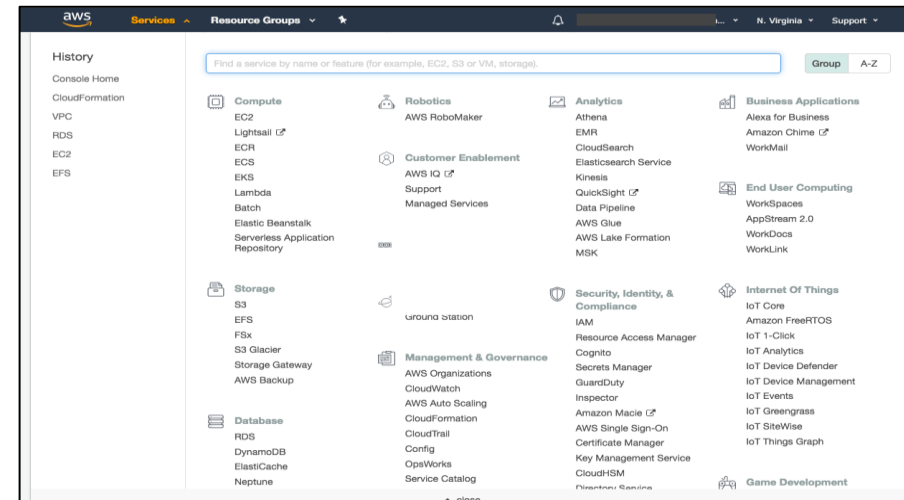
Password:

MFA users, enter your code on the next screen.

Sign In

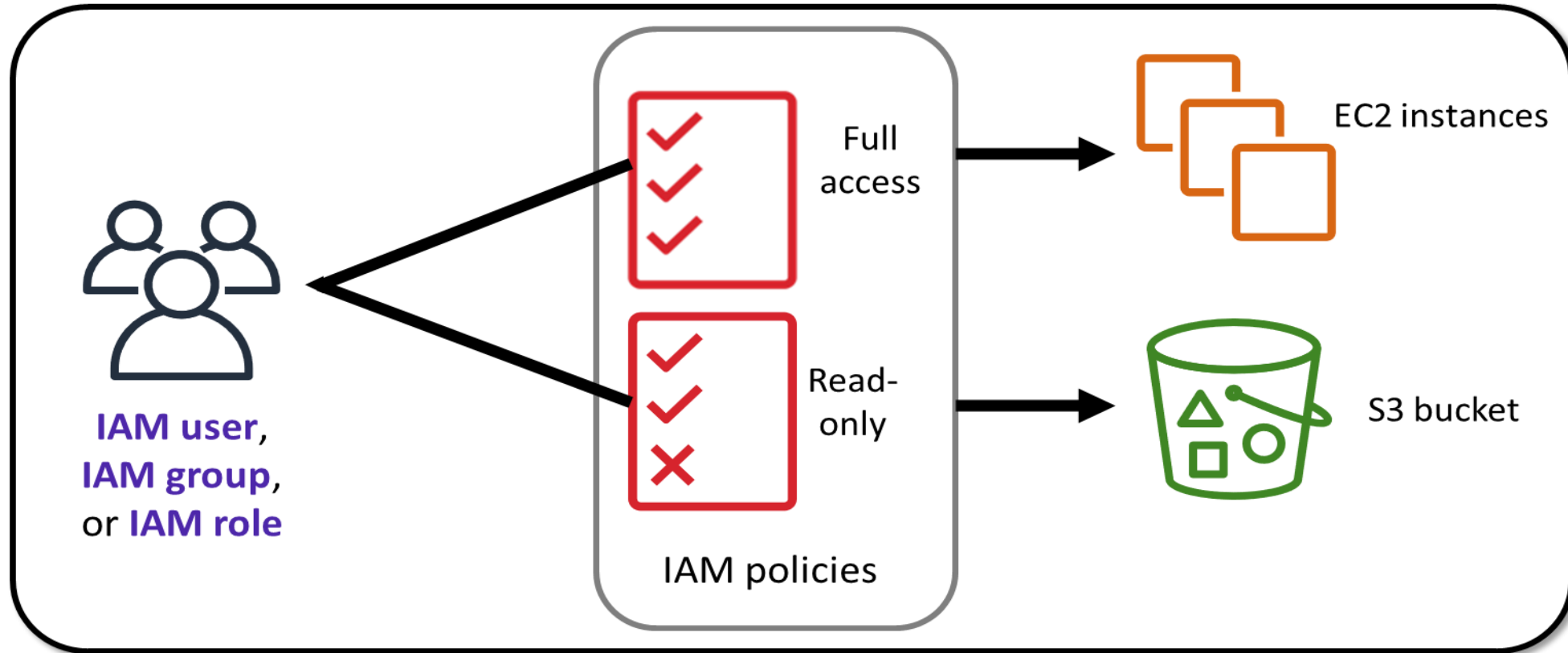

User name
and password →

MFA token



AWS Management Console

Authorization: What actions are permitted



IAM: Authorization

- Assign permissions by creating an IAM policy.
- Permissions determine **which resources and operations** are allowed:
 - All permissions are implicitly denied by default.
 - If something is explicitly denied, it is never allowed.



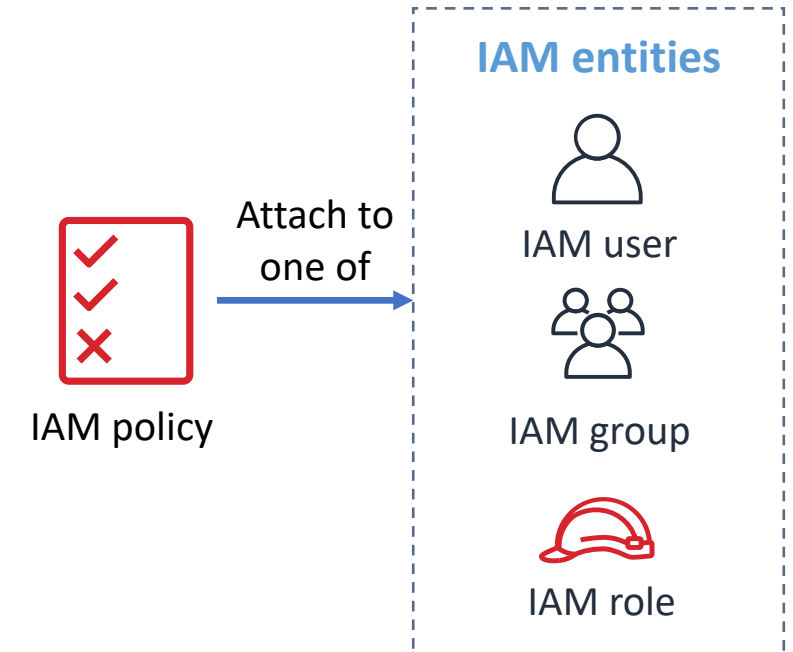
IAM
permissions

Best practice: Follow the **principle of least privilege**.

Note: The scope of IAM service configurations is **global**. Settings apply across all AWS Regions.

IAM policies

- An IAM policy is a document that defines permissions
 - Enables fine-grained access control
- Two types of policies – *identity-based* and *resource-based*
- Identity-based policies –
 - Attach a policy to any IAM entity
 - An IAM user, an IAM group, or an IAM role
 - Policies specify:
 - Actions that **may** be performed by the entity
 - Actions that **may not** be performed by the entity
 - A single *policy* can be attached to multiple *entities*
 - A single *entity* can have multiple *policies* attached to it
- Resource-based policies
 - Attached to a resource (such as an S3 bucket)



IAM policy example

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["dynamodb:*", "s3:*"],
    "Resource": [
      "arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
  },
  {
    "Effect": "Deny",
    "Action": ["dynamodb:*", "s3:*"],
    "NotResource": ["arn:aws:dynamodb:region:account-number-without-hyphens:table/table-
name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
  }
]

```

Explicit allow gives users access to a specific DynamoDB table and...

...Amazon S3 buckets.

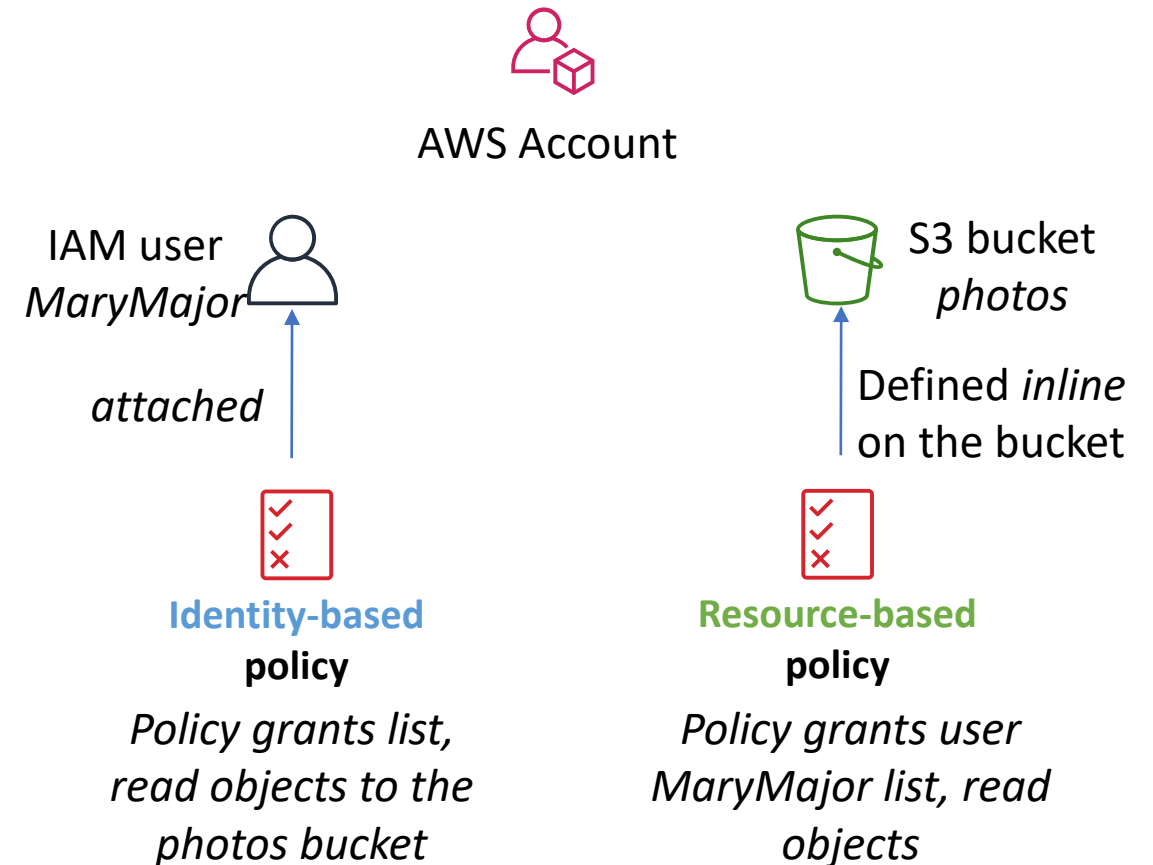
Explicit deny ensures that the users cannot use any other AWS actions or resources other than that table and those buckets.

An explicit deny statement **takes precedence** over an allow statement.



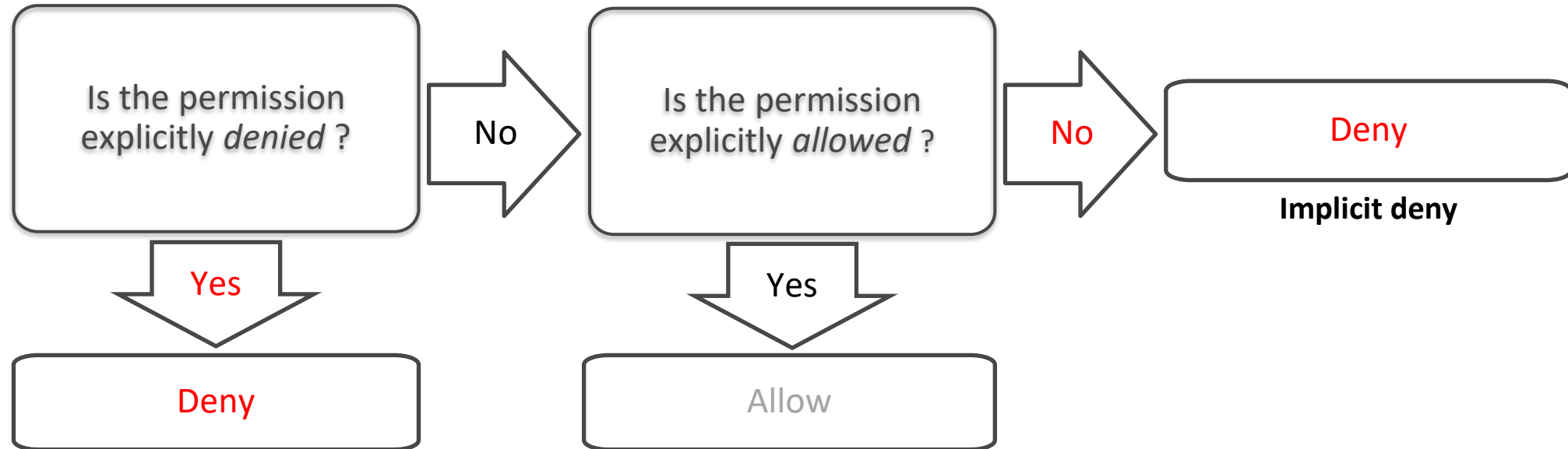
Resource-based policies

- **Identity-based policies** are attached to a user, group, or role
- **Resource-based policies** are attached to a resource (*not* to a user, group or role)
- Characteristics of resource-based policies –
 - Specifies who has access to the resource and what actions they can perform on it
 - The policies are *inline* only, not managed
- Resource-based policies are supported only by some AWS services



IAM permissions

How IAM determines permissions:



IAM groups

- An **IAM group** is a collection of IAM users
- A group is used to grant the same permissions to multiple users
 - Permissions granted by attaching IAM *policy* or policies to the group
- A user can belong to multiple groups
- There is no default group
- Groups cannot be nested



IAM roles

- An **IAM role** is an IAM identity with specific permissions
- Similar to an IAM user
 - Attach permissions policies to it
- Different from an IAM user
 - Not uniquely associated with one person
 - Intended to be *assumable* by a **person**, **application**, or **service**
- Role provides *temporary* security credentials
- Examples of how IAM roles are used to **delegate** access –
 - Used by an IAM user in the same AWS account as the role
 - Used by an AWS service—such as Amazon EC2—in the same account as the role
 - Used by an IAM user in a different AWS account than the role



IAM role

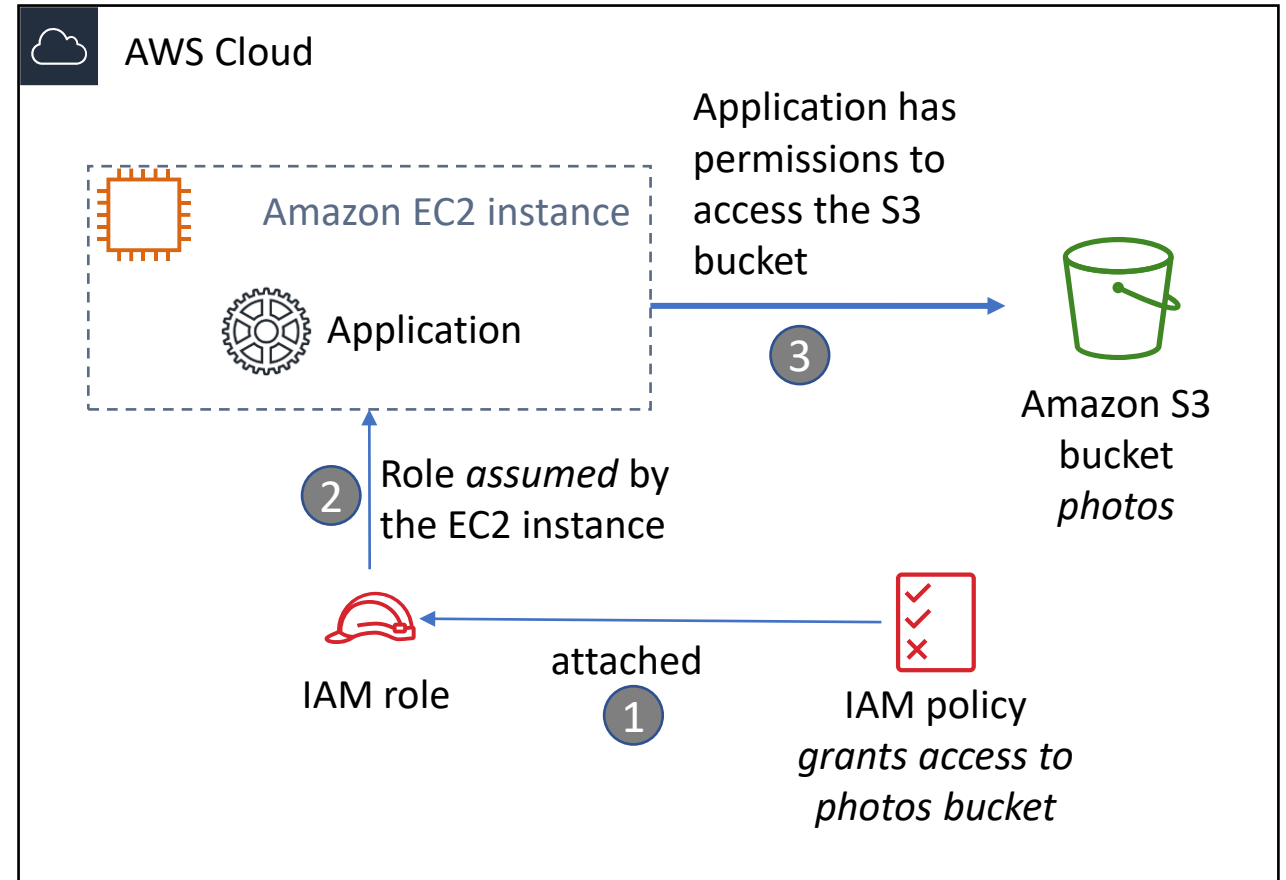
Example use of an IAM role

Scenario:

- An application that runs on an EC2 instance needs access to an S3 bucket

Solution:

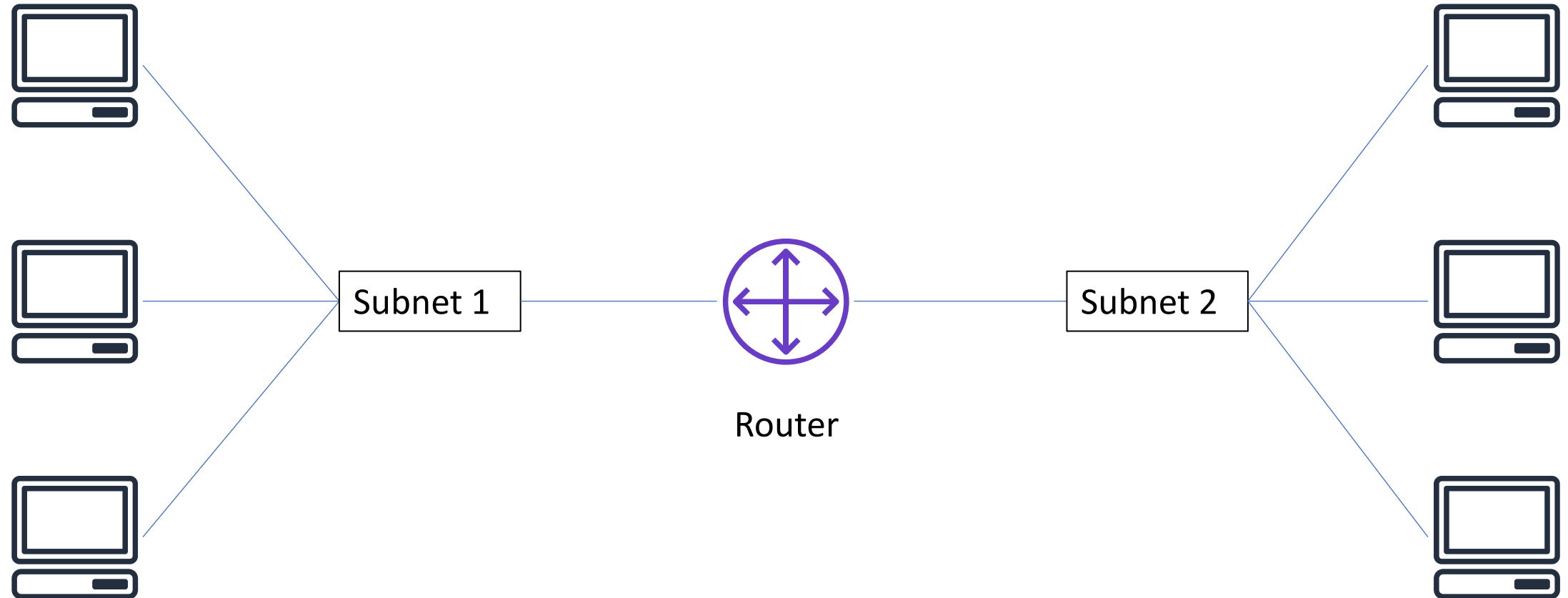
- Define an IAM policy that grants access to the S3 bucket.
- Attach the policy to a role
- Allow the EC2 instance to assume the role



Networking basics



Networks



IP addresses

192 . 0 . 2 . 0



11000000



00000000



00000010



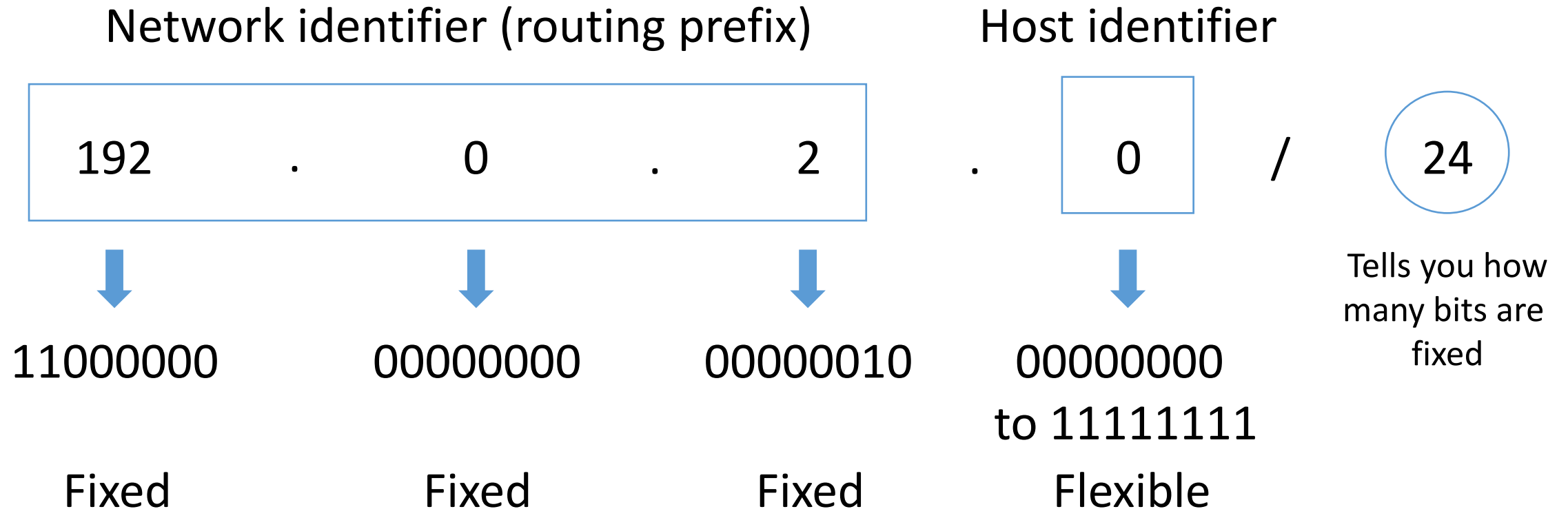
00000000

IPv4 and IPv6 addresses

IPv4 (32-bit) address: 192.0.2.0

IPv6 (128-bit) address: 2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF

Classless Inter-Domain Routing (CIDR)



Amazon VPC, Networking and Security



Amazon VPC

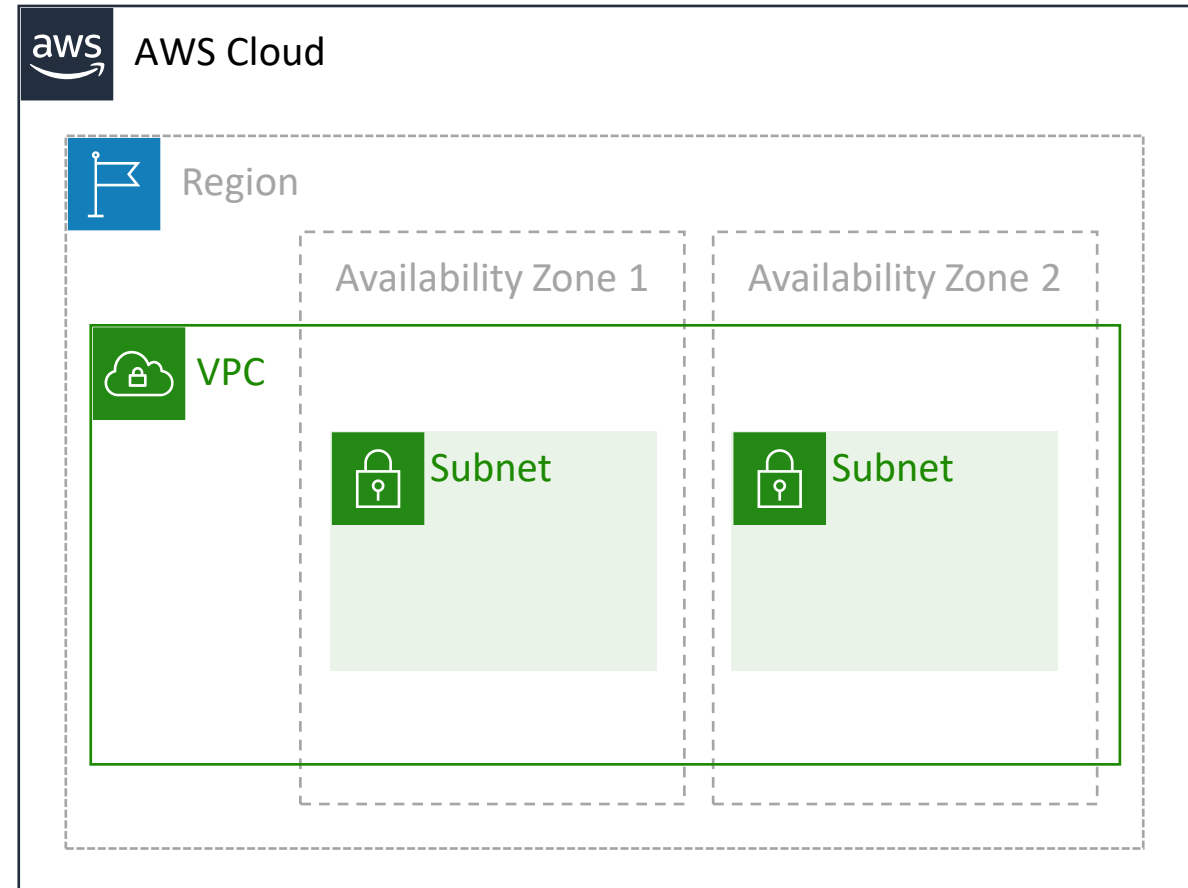
- Enables you to provision a **logically isolated** section of the AWS Cloud where you can launch AWS resources in a virtual network that you define
- Gives you **control over your virtual networking resources**, including:
 - Selection of IP address range
 - Creation of subnets
 - Configuration of route tables and network gateways
- Enables you to **customize the network configuration** for your VPC
- Enables you to use **multiple layers of security**



Amazon
VPC

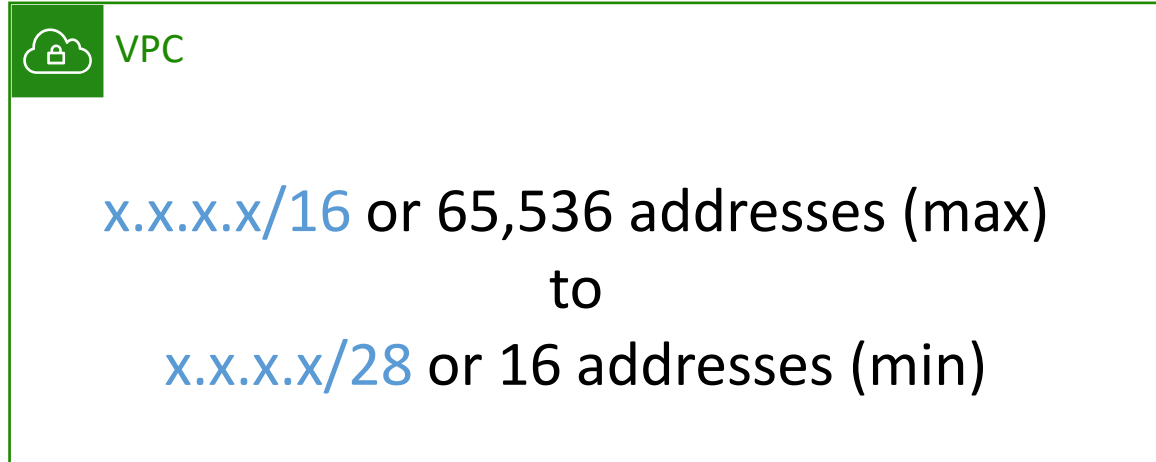
VPCs and subnets

- VPCs:
 - **Logically isolated** from other VPCs
 - **Dedicated** to your AWS account
 - Belong to a single **AWS Region** and can span multiple Availability Zones
- Subnets:
 - **Range of IP addresses** that divide a VPC
 - Belong to a single **Availability Zone**
 - Classified as **public** or **private**



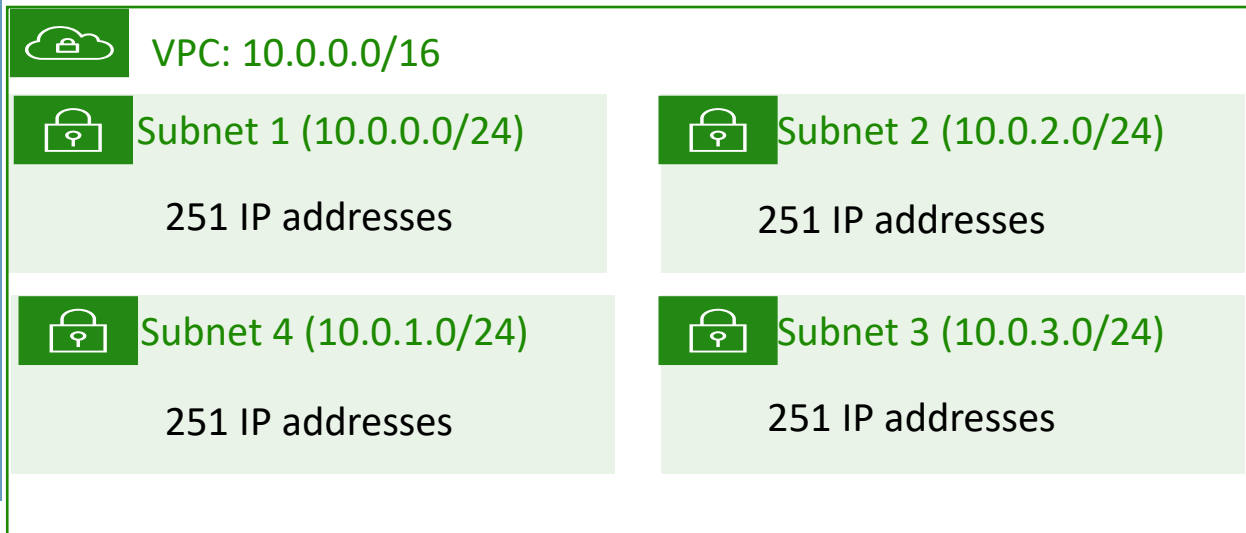
IP addressing

- When you create a VPC, you assign it to an IPv4 **CIDR block** (range of **private** IPv4 addresses).
- You **cannot change the address range** after you create the VPC.
- The **largest** IPv4 CIDR block size is **/16**.
- The **smallest** IPv4 CIDR block size is **/28**.
- IPv6 is also supported (with a different block size limit).
- CIDR blocks of subnets **cannot overlap**.



Reserved IP addresses

Example: A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication (VPC local router)
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

Public IP address types

Public IPv4 address

- Manually assigned through an Elastic IP address
- Automatically assigned through the auto-assign public IP address settings at the subnet level

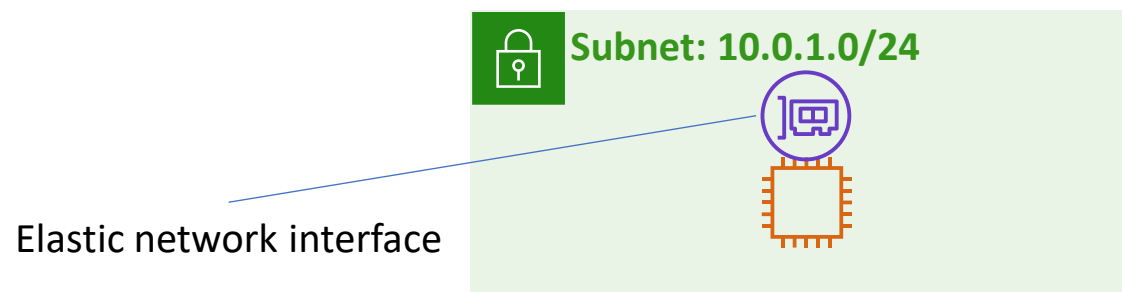
Elastic IP address

- a static and public IPv4 address
- Associated with an AWS account
- associate with any [instance](#) or [network interface](#) for any VPC in your account
- Can be allocated and remapped anytime
- Additional costs might apply



Elastic network interface

- An elastic network interface is a **virtual network interface** that you can:
 - Attach to an instance.
 - Detach from the instance and attach to another instance to redirect network traffic.
- Its **attributes follow** when it is reattached to a new instance.
- Each instance in your VPC has a **default network interface** that is assigned a private IPv4 address from the IPv4 address range of your VPC.



Route tables and routes

- A **route table** contains a set of rules (or routes) that **you can configure** to direct network traffic from your subnet.
- Each **route** specifies a destination and a target.
- By default, every route table contains a **local route** for communication within the VPC.
- Each **subnet must be associated with a route table** (at most one).

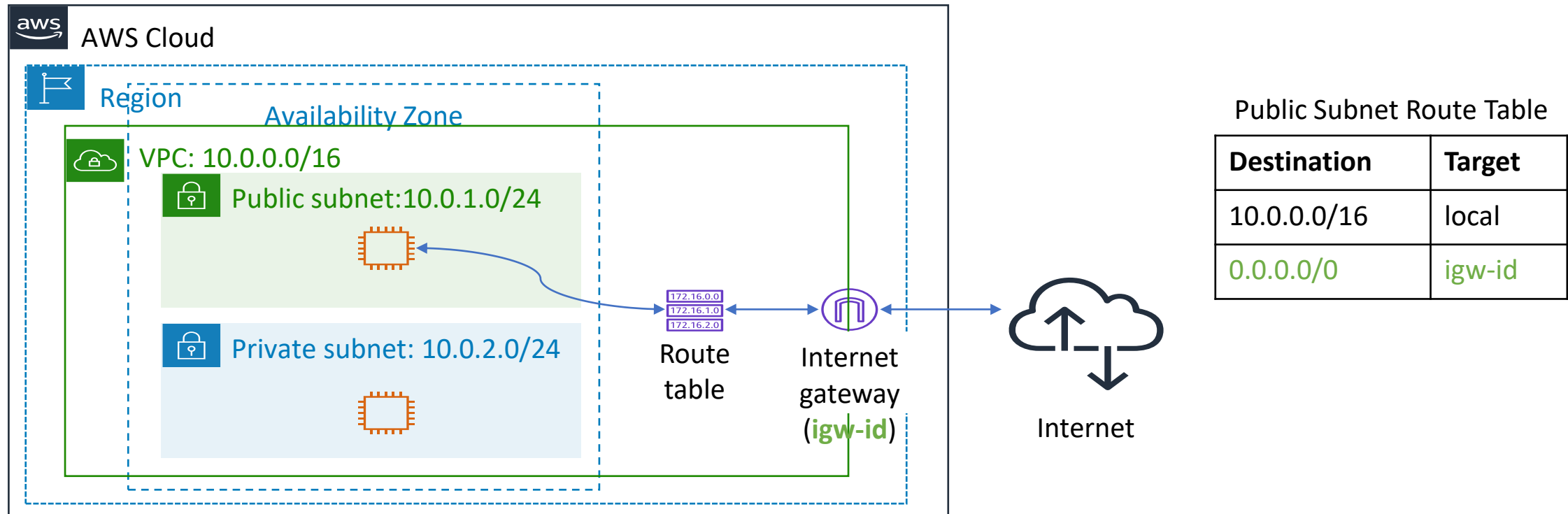
Main (Default) Route Table

Destination	Target
10.0.0.0/16	local

VPC CIDR block

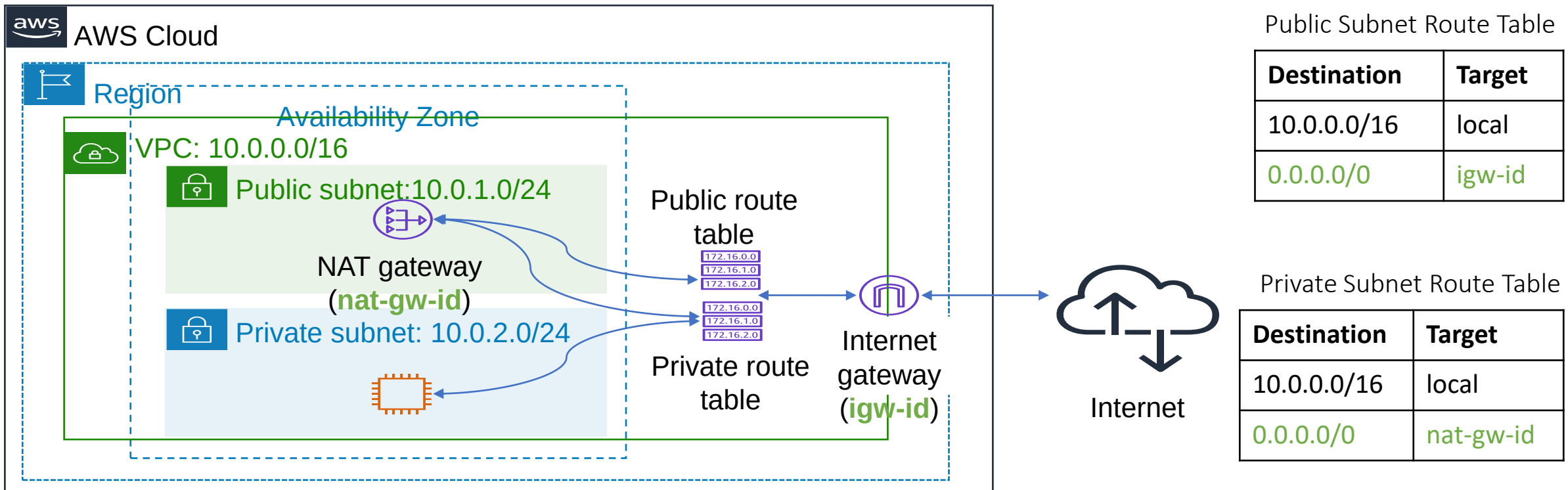
Internet gateway

An **internet gateway** is a scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet



Network address translation (NAT) gateway

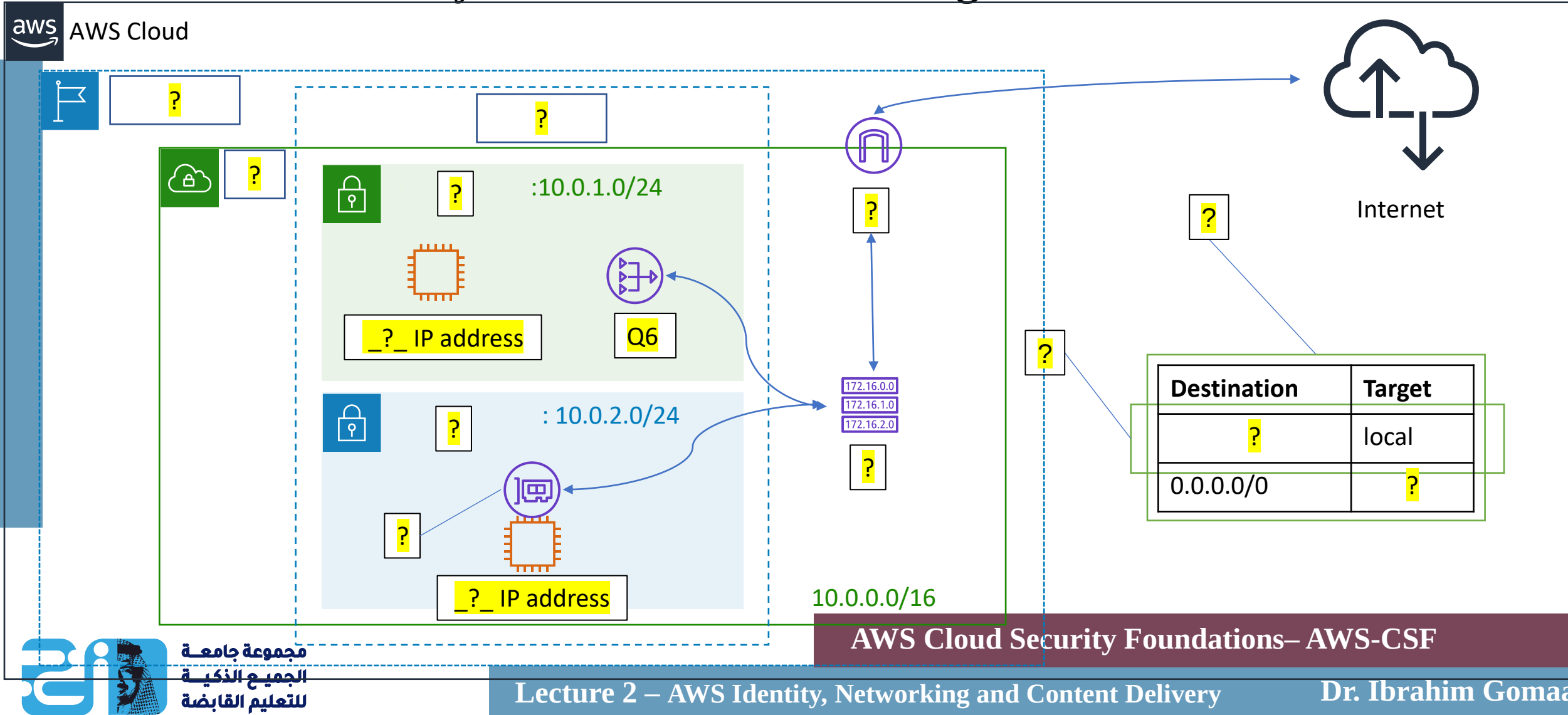
A **network address translation (NAT) gateway** enables instances in a private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a connection with those instances.



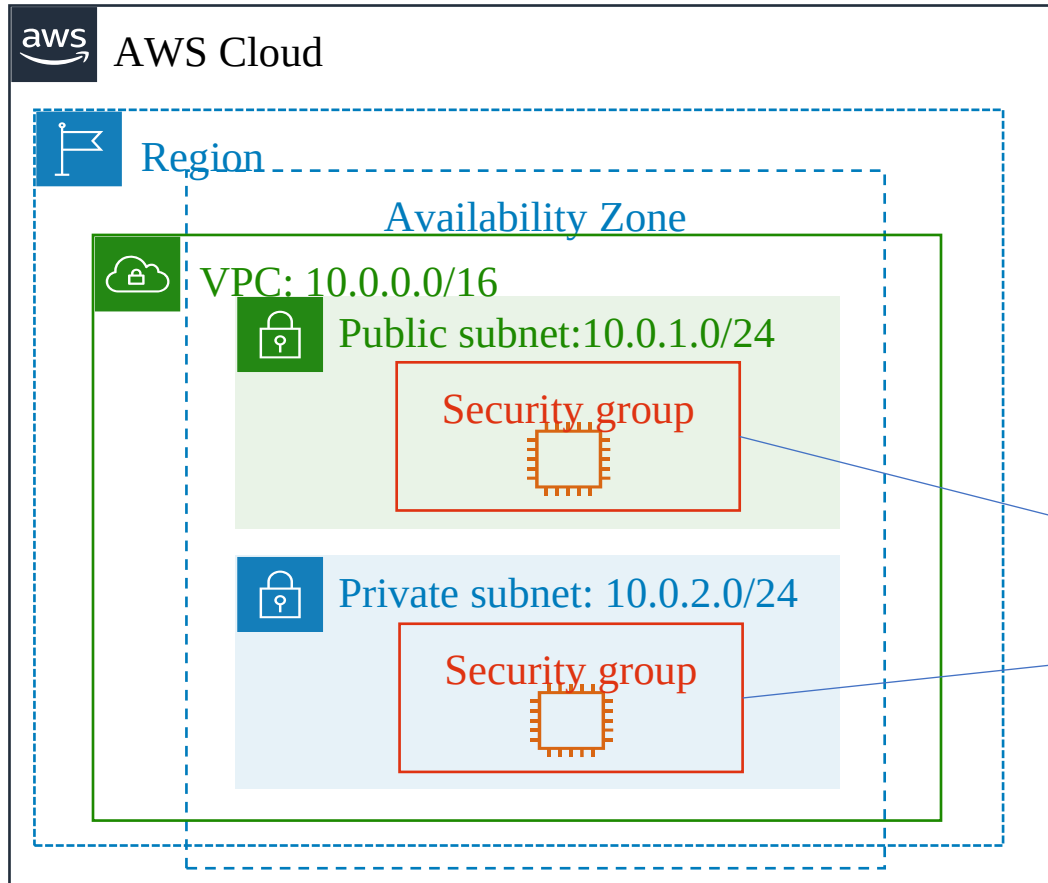
Lecture 2

Amazon VPC, networking and security

Activity: Label this network diagram



Security groups



A **security group** acts as a **virtual firewall** for your **instance**, and it controls inbound and outbound traffic. Security groups act at the **instance level**, not the subnet level. Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups.

Security groups act at the **instance level**.

Security groups

Inbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-xxxxxxx	
Outbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-xxxxxxx	

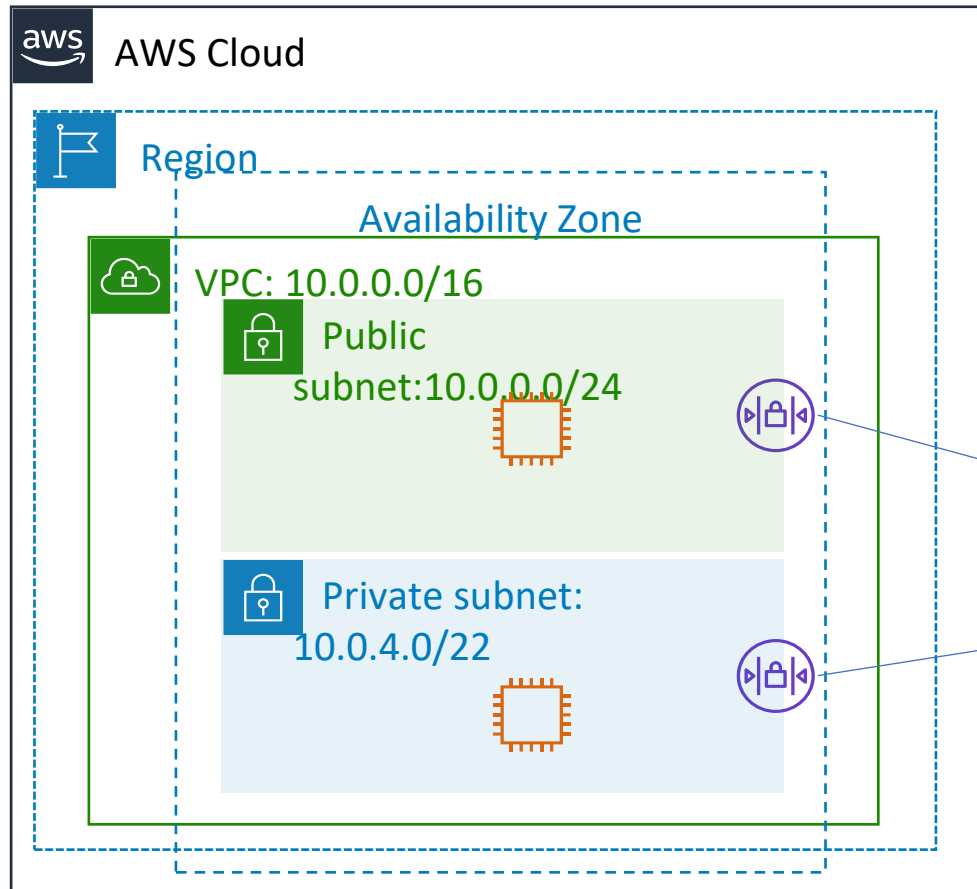
- Security groups have **rules** that control inbound and outbound instance traffic.
- Default security groups **deny all inbound** traffic and **allow all outbound** traffic.
- Security groups are **stateful**.

Custom security groups

Inbound				
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	All web traffic
HTTPS	TCP	443	0.0.0.0/0	All web traffic
SSH	TCP	22	54.24.12.19/32	Office address
Outbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

- You can **specify allow** rules, but not deny rules.
- **All rules are evaluated** before the decision to allow traffic.

Network access control lists (network ACLs)



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

Network ACLs act at the **subnet level**.

Network access control lists (network ACLs)

Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

- A network ACL has **separate inbound and outbound rules**, and each rule can either **allow or deny traffic**.
- **Default** network ACLs **allow** all inbound and outbound IPv4 traffic.
- Network ACLs are **stateless**.

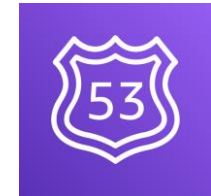
Security groups versus network ACLs

Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated before decision to allow traffic	Rules are evaluated in number order before decision to allow traffic

Amazon Route 53



- Is a highly available and scalable Domain Name System (DNS) web service
- Is used to route end users to internet applications by translating names (like www.example.com) into numeric IP addresses (like *192.0.2.1*) that computers use to connect to each other
- Is fully compliant with IPv4 and IPv6
- Connects user requests to infrastructure running in AWS and also outside of AWS
- Is used to check the health of your resources
- Features traffic flow
- Enables you to register domain names



Amazon
Route 53

Amazon Route 53 supported routing

Amazon Route 53 supports several types of **routing policies**, which determine how Amazon Route 53 responds to queries:

- **Simple routing (round robin)** – Use in single-server environments
- **Weighted round robin routing** – Assign weights to resource record sets to specify the frequency
- **Latency routing** – Help improve your global applications
- **Geolocation routing** – Route traffic based on location of your users
- **Geoproximity routing** – Route traffic based on location of your resources
- **Failover routing** – Fail over to a backup site if your primary site becomes unreachable
- **Multivalue answer routing** – Respond to DNS queries with up to eight healthy records selected at random

DNS failover for a multi-tiered web application

Record Sets

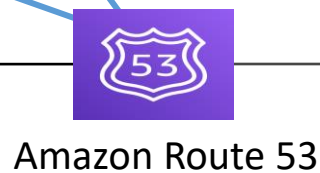
CNAME www

lastic_load_balancer
Routing Policy = Failover
Record Type = Primary

Amazon S3 website
Routing Policy = Failover
Record Type = Secondary



User

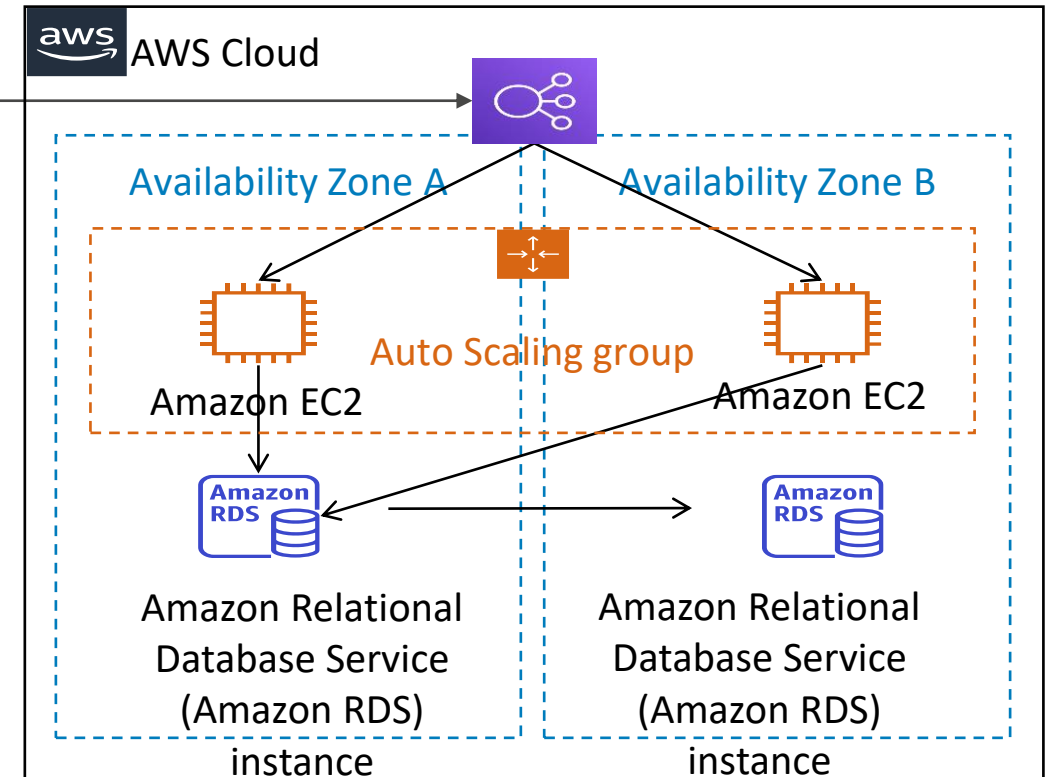


Primary

Secondary



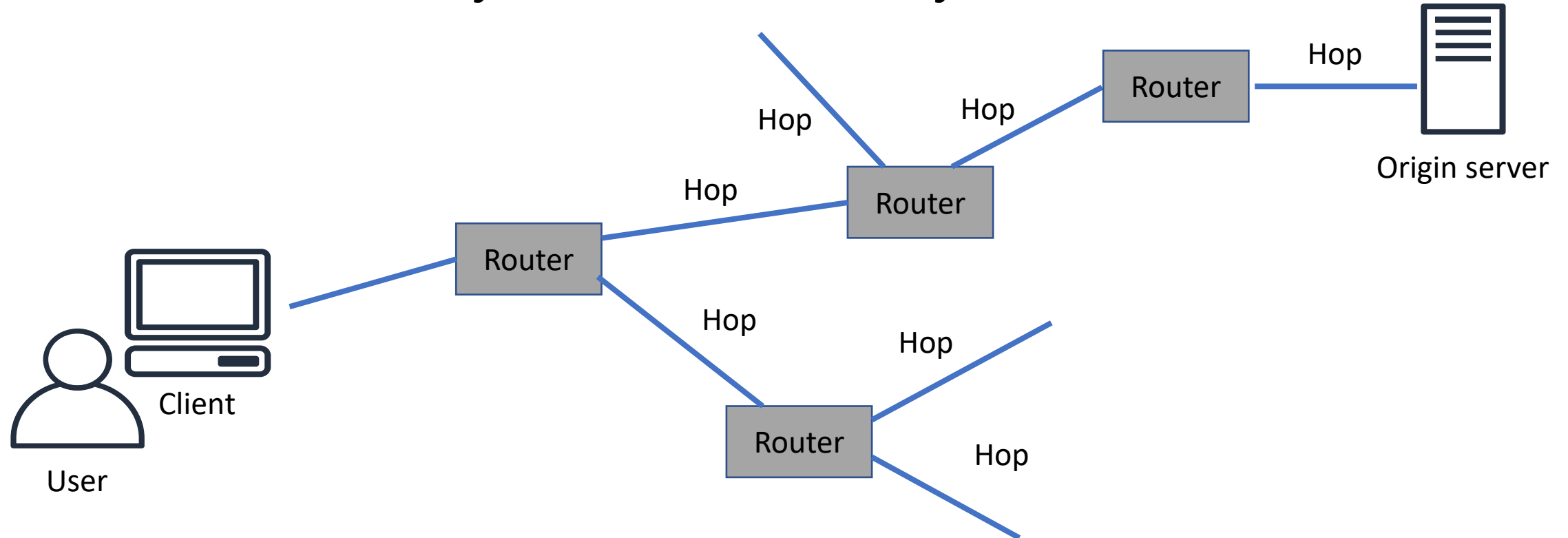
Amazon S3
static website



Amazon CloudFront



Content delivery and network latency



Content delivery network (CDN)

- Is a globally distributed system of caching servers
- Caches copies of commonly requested files (static content)
- Delivers a local copy of the requested content from a nearby cache edge or Point of Presence
- Accelerates delivery of dynamic content
- Improves application performance and scaling

Amazon CloudFront

- Fast, global, and secure CDN service
- Global network of edge locations and Regional edge caches
- Self-service model
- Pay-as-you-go pricing

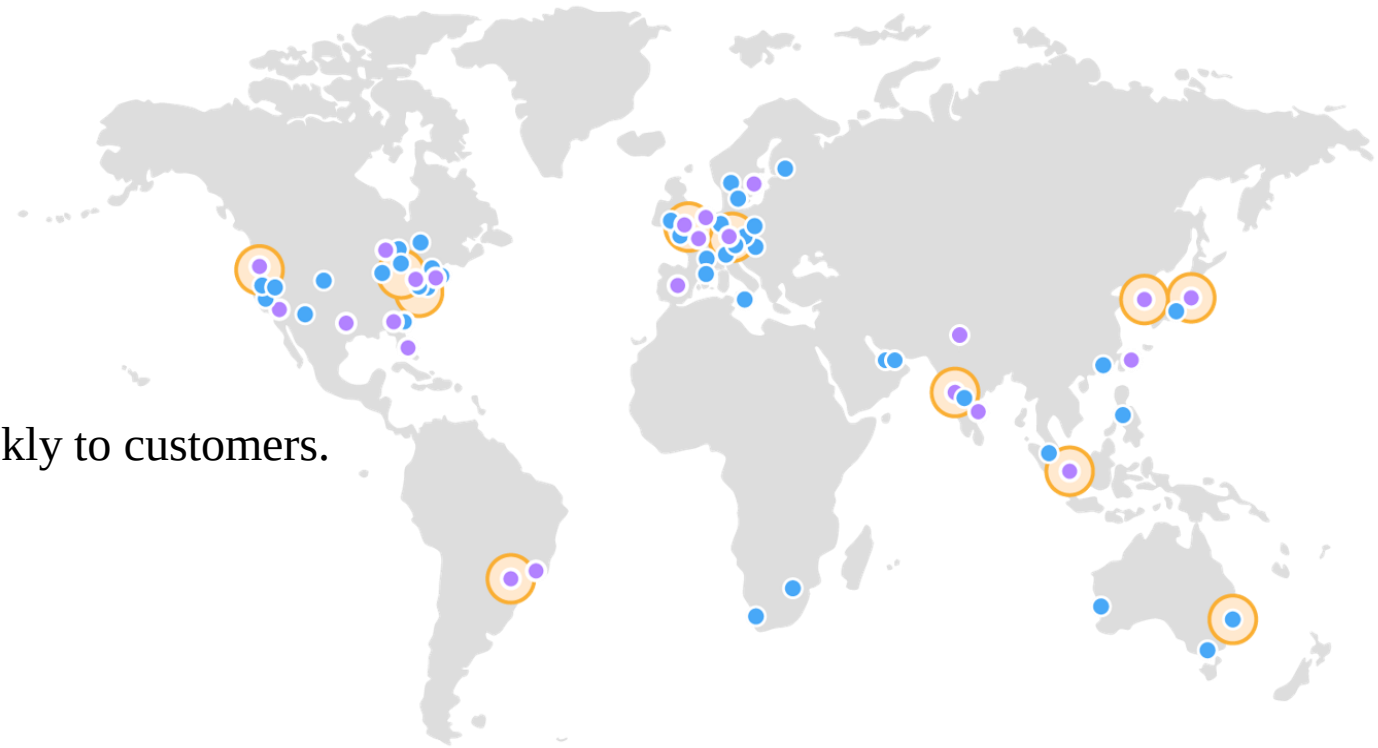


Amazon
CloudFront

Amazon CloudFront infrastructure

- Edge locations
- Multiple edge locations
- Regional edge caches

- **Edge locations** – Network of data centers that CloudFront uses to serve popular content quickly to customers.
- **Regional edge cache** – CloudFront location that caches content that is not popular enough to stay at an edge location. It is located between the origin server and the global edge location.



Amazon CloudFront benefits

- Fast and global
- Security at the edge
- Highly programmable
- Deeply integrated with AWS
- Cost-effective

Amazon CloudFront pricing

Data transfer out

- Charged for the volume of data transferred out from Amazon CloudFront edge location to the internet or to your origin.

HTTP(S) requests

- Charged for number of HTTP(S) requests.

Invalidation requests

- No additional charge for the first 1,000 paths that are requested for invalidation each month. Thereafter, \$0.005 per path that is requested for invalidation.

Dedicated IP custom SSL

- \$600 per month for each custom SSL certificate that is associated with one or more CloudFront distributions that use the Dedicated IP version of custom SSL certificate support.

Thanks