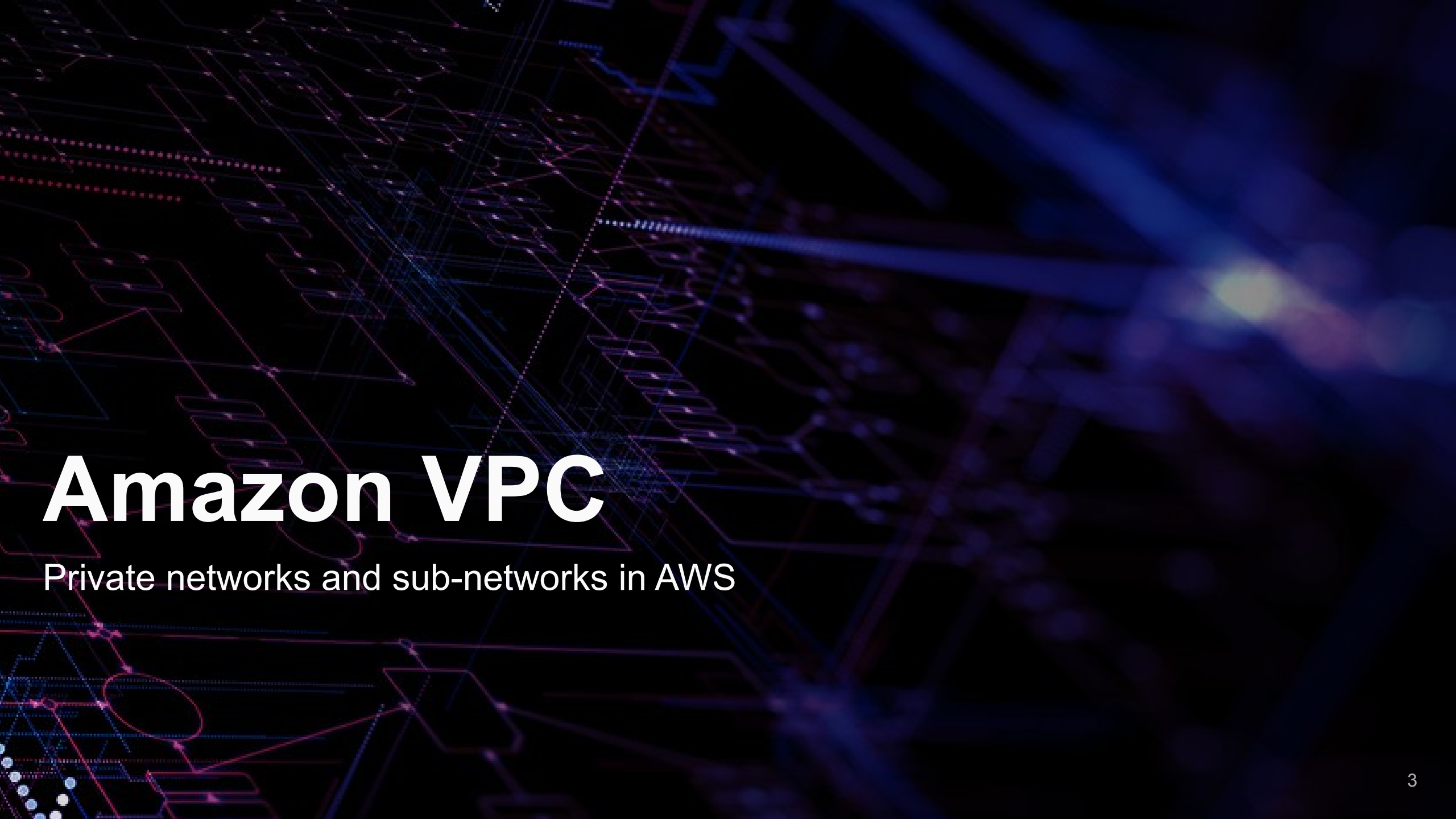
AWS Architecting and SysOps

Networking and Content Delivery, Part 1

June-July 2019

Contents



Amazon VPC

Private networks and sub-networks in AWS

Networking & Content Delivery

Amazon VPC

Elastic Load Balancing

AWS VPN

AWS Direct Connect

Amazon CloudFront

Route 53

Amazon API Gateway

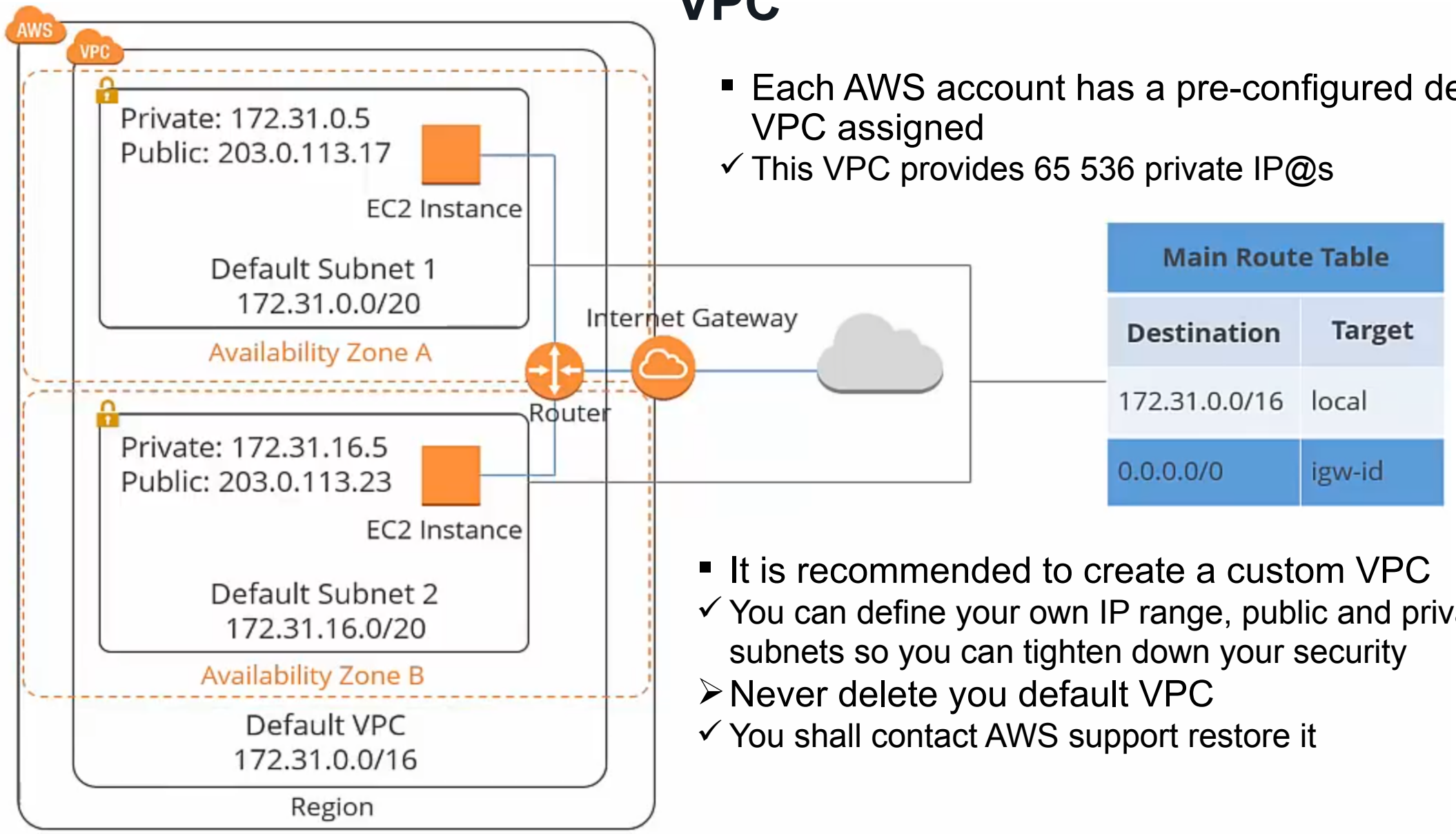
Virtual Private Cloud (VPC)

- VPC lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define
- Complete control over your virtual networking environment:
  - Selection of your own IP (IPv4 or IPv6) address range
  - Creation of subnets
  - Configuration of route tables
  - Configuration of network gateways
- Easily customize the network configuration for you VPC
  - You can create a public-facing subnet for your web servers that has access to the Internet, and place the backend systems in a private-facing subnet with no Internet access
- Multiple layers of security controls
  - Ability to allow / deny specific Internet and internal traffic
- Integration with AWS services

VPC terms

- VPC
  - A logically isolated virtual network in the AWS cloud defined by VPC's IP address range selected
  - You can have multiple VPCs per AWS account, and each VPC you create lives with a region
- Subnet
  - A segment of a VPC's IP address range where you can place groups of isolated resources
- Route table
  - Controls traffic between subnets and going out of the subnets
- Internet Gateway (IGW)
  - The Amazon VPC side of a connection to the public Internet
  - Allows access to the Internet from Amazon VPC
- NAT (Network Address Translation) Gateway
  - Managed service for your resources in a private subnet to access the Internet
  - Allows private subnet resources to access Internet

VPC



- Each AWS account has a pre-configured default VPC assigned

✓ This VPC provides 65 536 private IP@s

- It is recommended to create a custom VPC

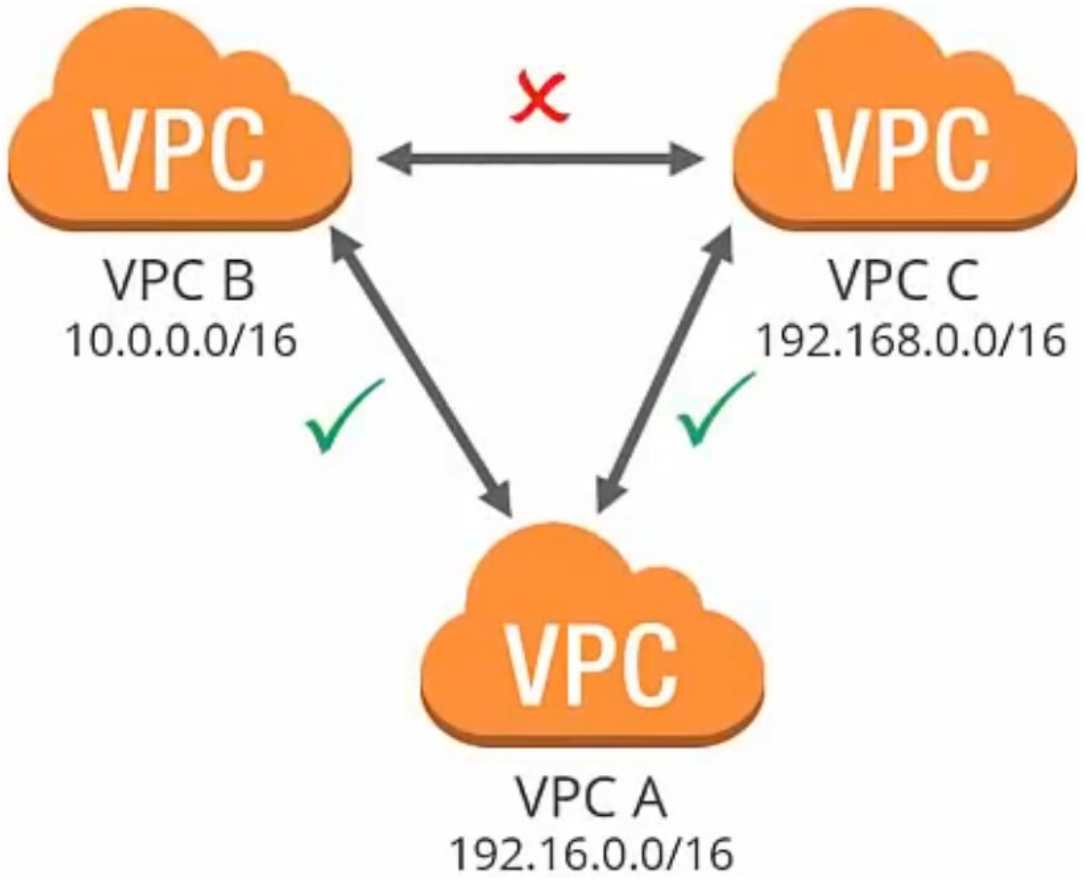
✓ You can define your own IP range, public and private subnets so you can tighten down your security

➢ Never delete you default VPC

✓ You shall contact AWS support restore it

VPC Peering Connection

- A peering connection can be made between your own VPC or with a VPC in another AWS account as long as it is in the same region



✓ Instances in VPC A cannot communicate with instances in VPC B and VPC C, unless you establish a peering connection

- Peering is a one-to-one relationship

✓ VPC can have multiple connections to other VPCs but transitive peering is not supported

- VPC with overlapping CIDRs cannot be paired

✓ Non-overlapping IP ranges can be paired

VPC IP addresses

- Private IP address
  - Used as communication between instances in the same network
  - When you launch a new instance, it is given a private IP address and an internal DNS hostname that resolves to that private IP address
- Public IP addresses
  - Used for communication between your instances and the Internet
  - Each instance that receives a public IP address, it also receives an external DNS hostname
  - Associated with your instances from the Amazon pool of public IP addresses
  - When you stop your instance, the public IP address is released and a new one is associated when the instance starts. The IP address is not persistent.
- Elastic IP addresses
  - Static or persistent public IP address allocated to your account
  - Until you release it, it can be associated to and from your instances as required
  - There is charge for any elastic IP that it is not associated or allocated to an instance

VPC Subnets

- A range of IP addresses in your VPC
  - Used to divide your VPC into multiple zones so you can launch AWS resources
  - VPC can span multiple AZs but a subnet is always associated to a single AZ
  - By default, subnets within a VPC can communicate to each other
  - The netmask for the default subnet in your VPC is always 20 which provides up to 4096 addresses per subnet minus the reserved addresses by AWS
- The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance
  - In a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:
    - 10.0.0.0: Network address
    - 10.0.0.1: Reserved by AWS for the VPC router
    - 10.0.0.2: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; so, AWS reserves the base of each subnet range plus two
    - 10.0.0.3: Reserved by AWS for future use
    - 10.0.0.255: Network broadcast address. AWS reserve it although it does not support broadcast yet

VPC Subnets

Easy subnet calculator: http://jodies.de/ipcalc

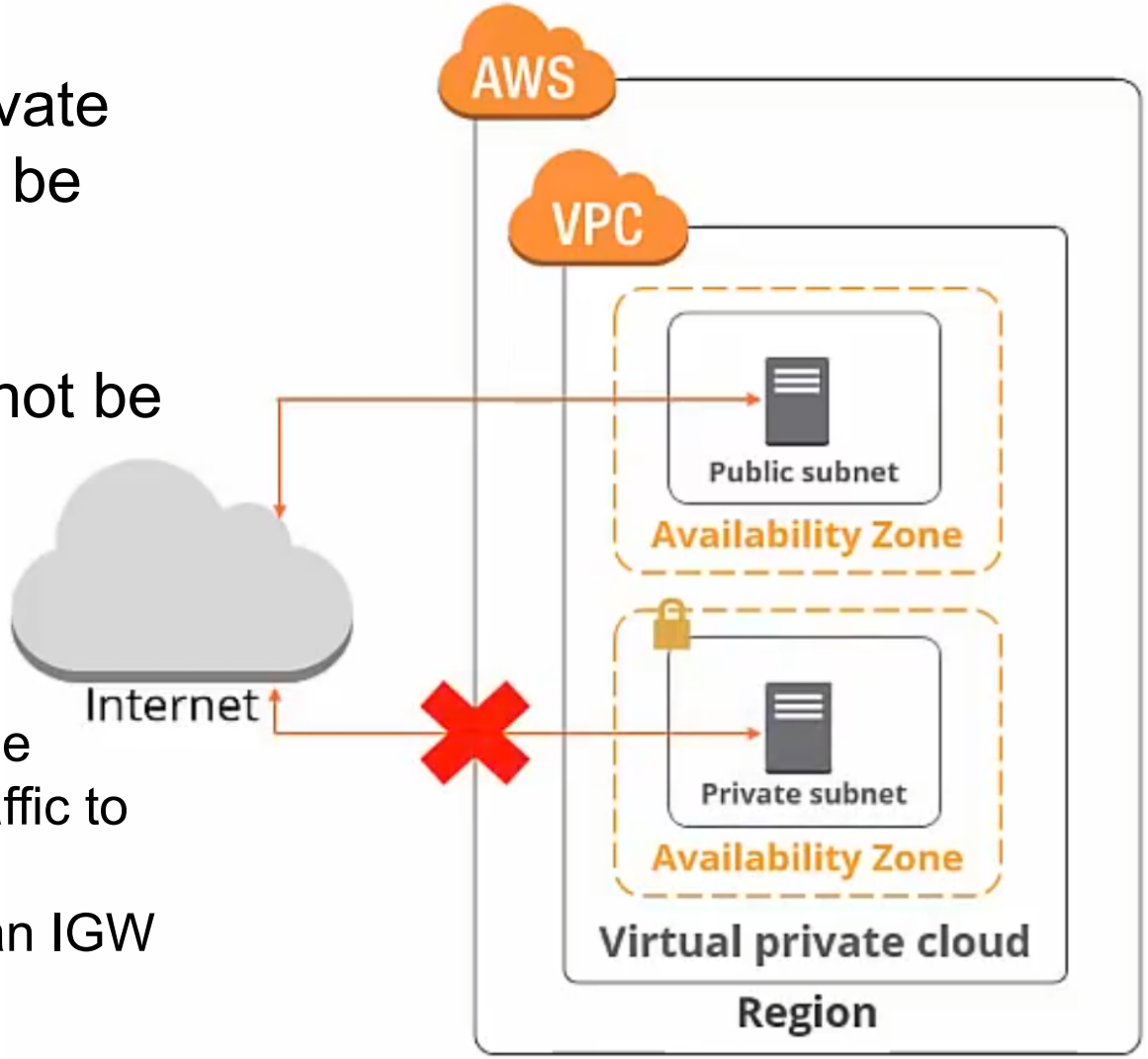
- CIDR block : 10.0.0.0/16
  HostMin: 10.0.0.1
  HostMax: 10.0.255.254
  Hosts/Net: 65534
- CIDR block : 10.0.0.0/18
  HostMin: 10.0.0.1
  HostMax: 10.0.63.254
  Hosts/Net: 16382
- CIDR block: 124.25.0.0/20
  HostMin: 124.25.0.1
  HostMax: 124.25.15.254
  Hosts/Net: 4094
- CIDR block: 124.25.0.0/22
  HostMin: 124.25.0.1
  HostMax: 124.25.3.254
  Hosts/Net: 1022
- CIDR block : 192.168.123.0/24
  HostMin: 192.168.123.1
  HostMax: 192.168.123.254
  Hosts/Net: 254
- CIDR block : 192.168.123.0/28
  HostMin: 192.168.123.1
  HostMax: 192.168.123.14
  Hosts/Net: 14

VPC subnets types

- There are 2 types of subnets: public and private
- Use a public subnet for resources that must be connected to the Internet
  - Web services
- Use a private subnet for resources that will not be connected to the Internet
  - Database instances

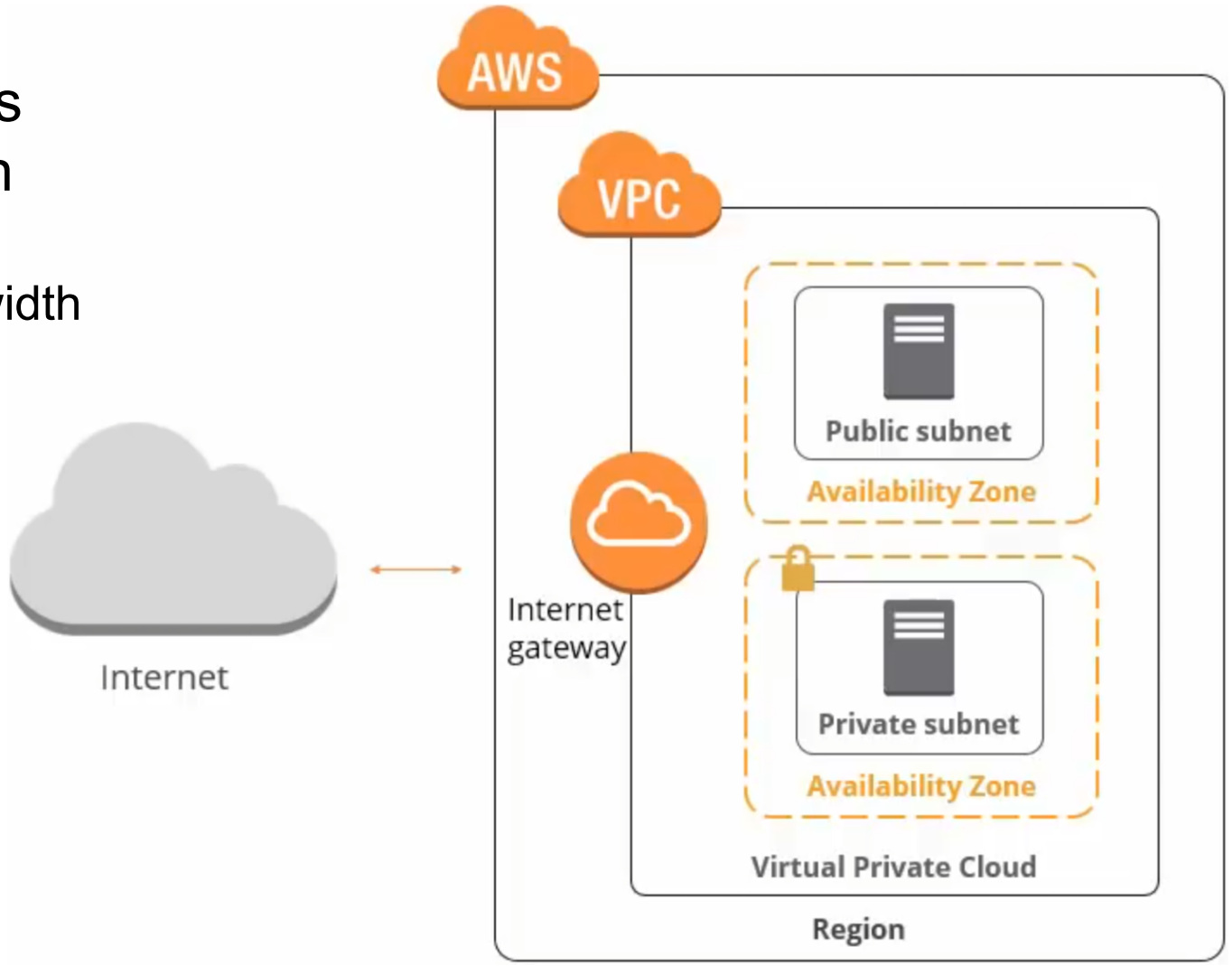
- For a subnet to be public
- We need to attach an IGW to the VPC and update the route tablet of the public subnet to send non-local traffic to the IGW

➢ EC2 instances need a public IP address to route to an IGW



VPC Internet Gateway

- A scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet
  - It imposes no availability risks or bandwidth constraints on your network traffic
- Serves two purposes:
  - Provide a target in your VPC route tables for internet-routable traffic
  - Perform NAT for instances that have been assigned public IPv4 addresses
- Supports IPv4 and IPv6 traffic



VPC Internet Gateway

- To enable access to or from the internet for instances in a VPC subnet, you must do the following:
  1. Attach an Internet Gateway to your VPC
  2. Ensure that your subnet's route table points to the IGW
  3. Ensure that instances in your subnet have a globally unique IP address
     - ✓ Public IPv4 address
     - ✓ Elastic IP address
     - ✓ IPv6 address
  4. Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance.

- Note: You can set the route to all destinations (0.0.0.0/0 for IPv4 or ::/0 for IPv6), or you can scope the route to a narrower range of IP addresses (public IPv4 addresses of your company's public endpoints outside of AWS, or the Elastic IP addresses of other EC2 instances outside your VPC)

➢ If your subnet is associated with a route table that has a route to an IGW, it is then a public subnet

VPC route tables

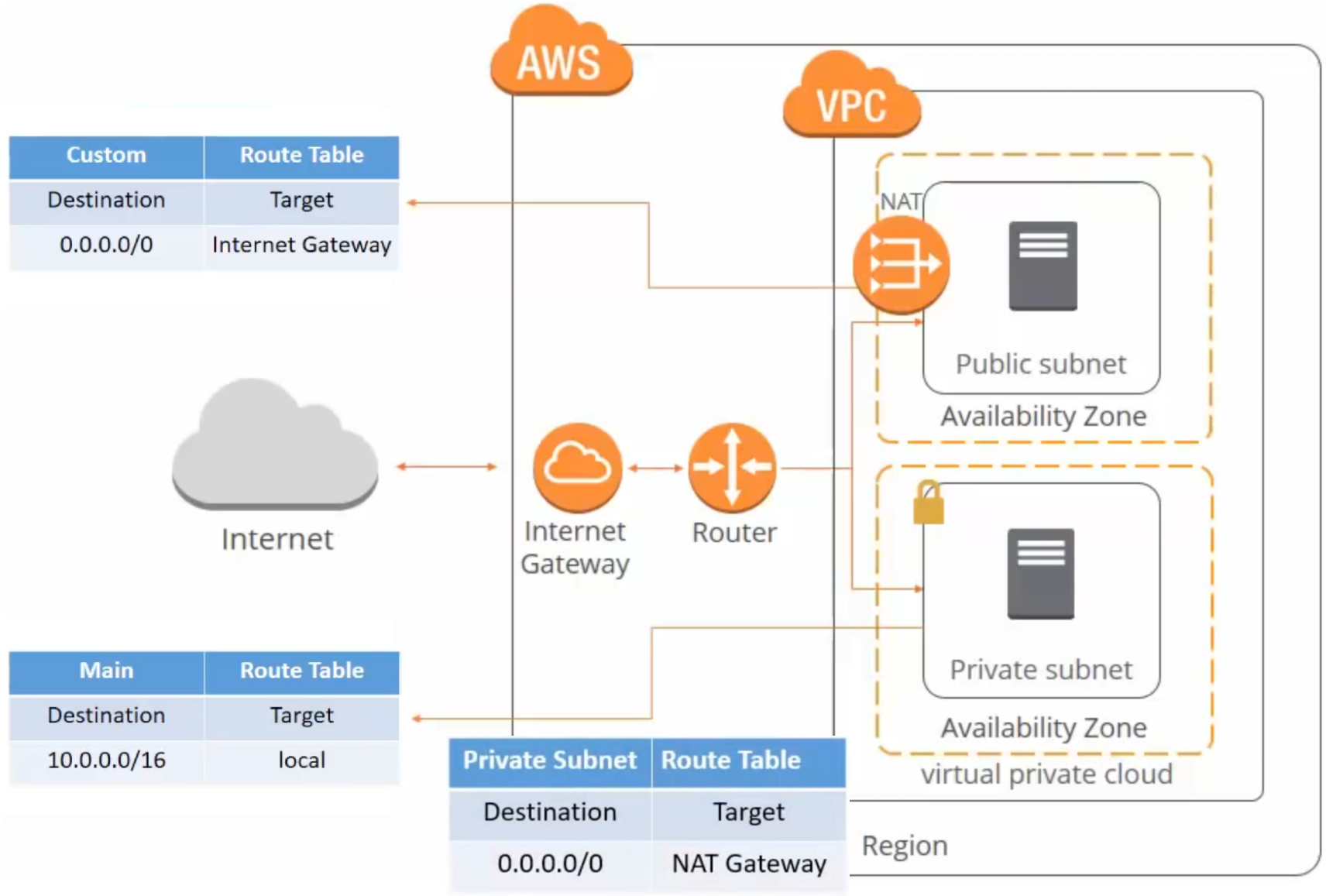
- A route table contains several routes
  - ❖Routes: a set of rules used to determine where network traffic is directed
- Each subnet must be associated with a route table
  - The route table controls the routing for the subnet
- A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table
- When you create a VPC, it automatically creates a main route table
- Your VPC can have custom route tables
  - One way to protect your VPC is to leave the main route table in its original default state (with only the local route), and explicitly associate each new subnet you create with one of the custom route tables you've created
  - This ensures that you explicitly control how each subnet routes outbound traffic

NAT Gateway

- Used to enable instances in a private subnet to connect to the Internet, but prevent the Internet from initiating a connection with those instances

➢ You are charged for creating and using a NAT gateway in your account

- NAT gateway hourly usage and data processing rates apply
- Amazon EC2 charges for data transfer also apply



Security Groups

- A security group acts as a virtual firewall that controls traffic from one or more instances
- You add inbound and outbound rules to each security group that allow traffic to or from its associated instances

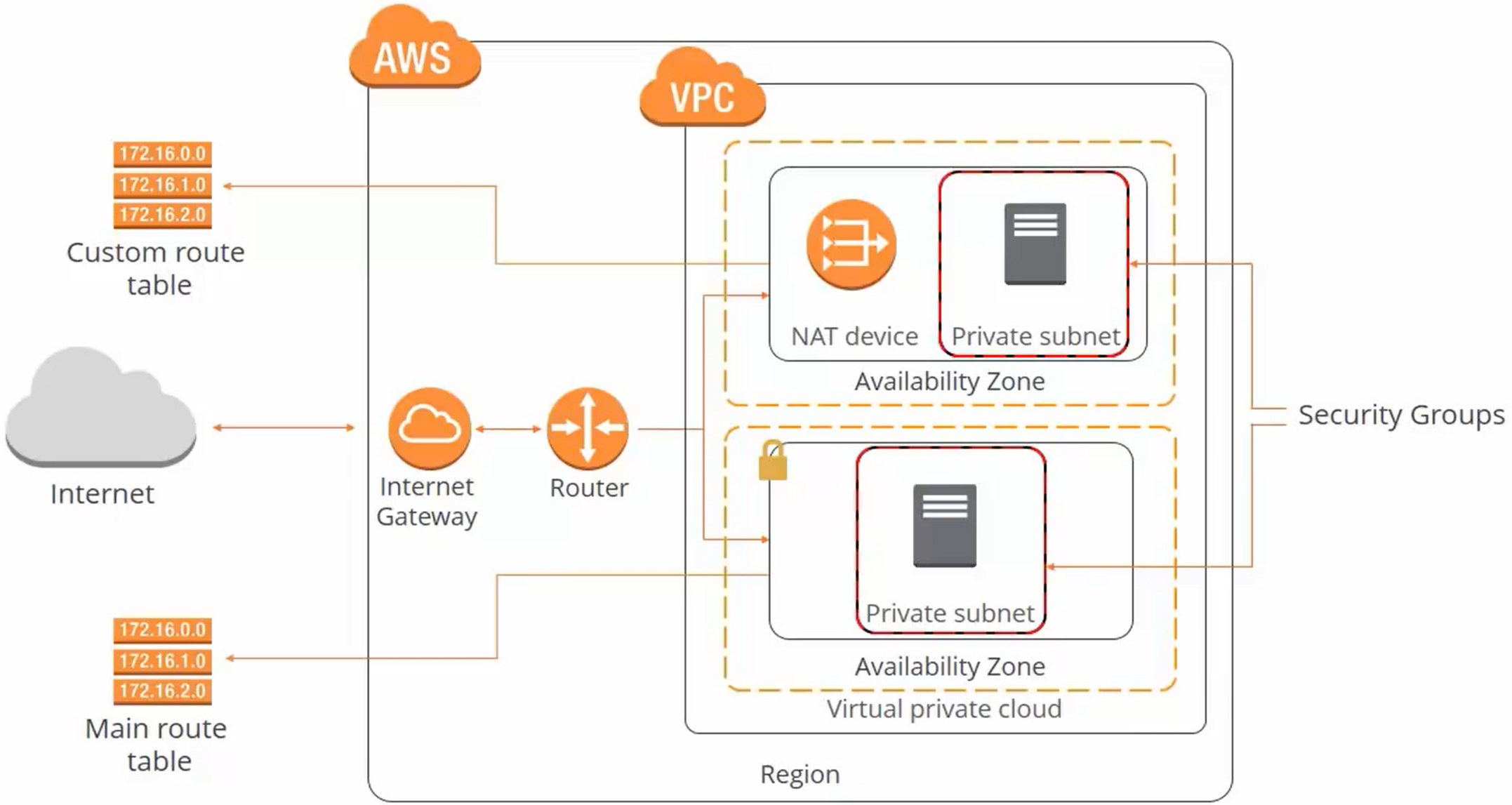
✓You can modify the rules at any time and they are applied immediately

✓Security group rules are always permissive (allow)

✓By default, security groups allow all outbound traffic and they do not have any inbound rule

- Your VPC comes with a default security group
- Al launch time of a new instance, the default security group is assigned to the instance unless you specify another security group/s
  - You can modify the security group/s assigned to any instance any time

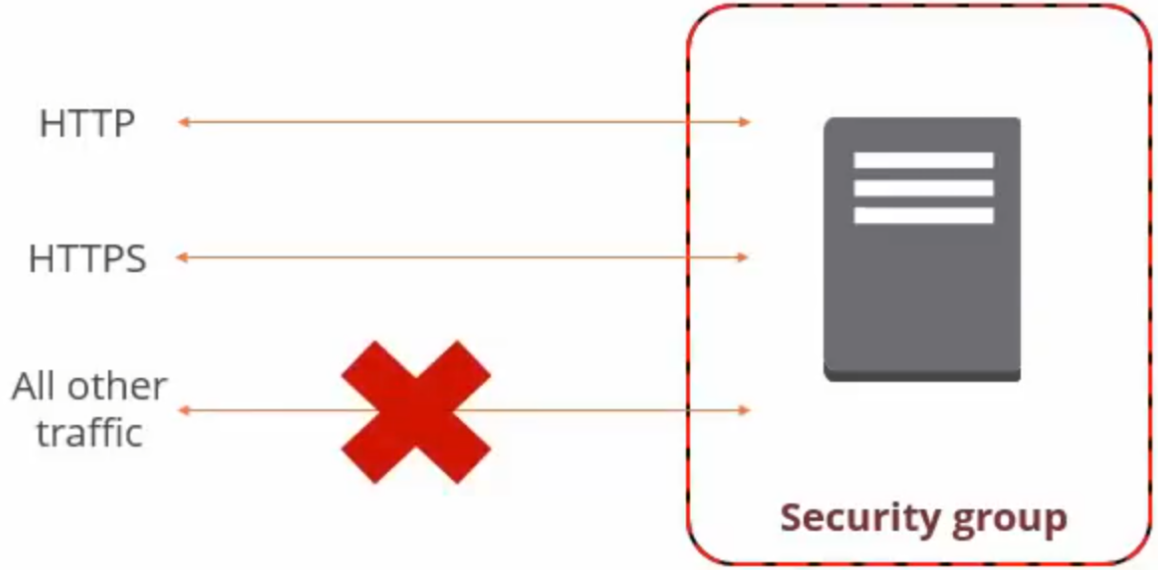
Security Groups



Security groups for webservers

- A webserver need HTTP and HTTPS traffic as a minimum to be able to access it
  - All traffic to specific ports (80 for HTTP, 443 for HTTPS) is allowed
  - All other traffic is not allowed to access any instance inside that security group

Type	Protocol	Port Range	Source
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0



Network Access Control List (Network ACL)

- A network ACL is an additional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets
- You add inbound at outbound rules to each network ACL to allow or deny traffic to or from its associated subnets

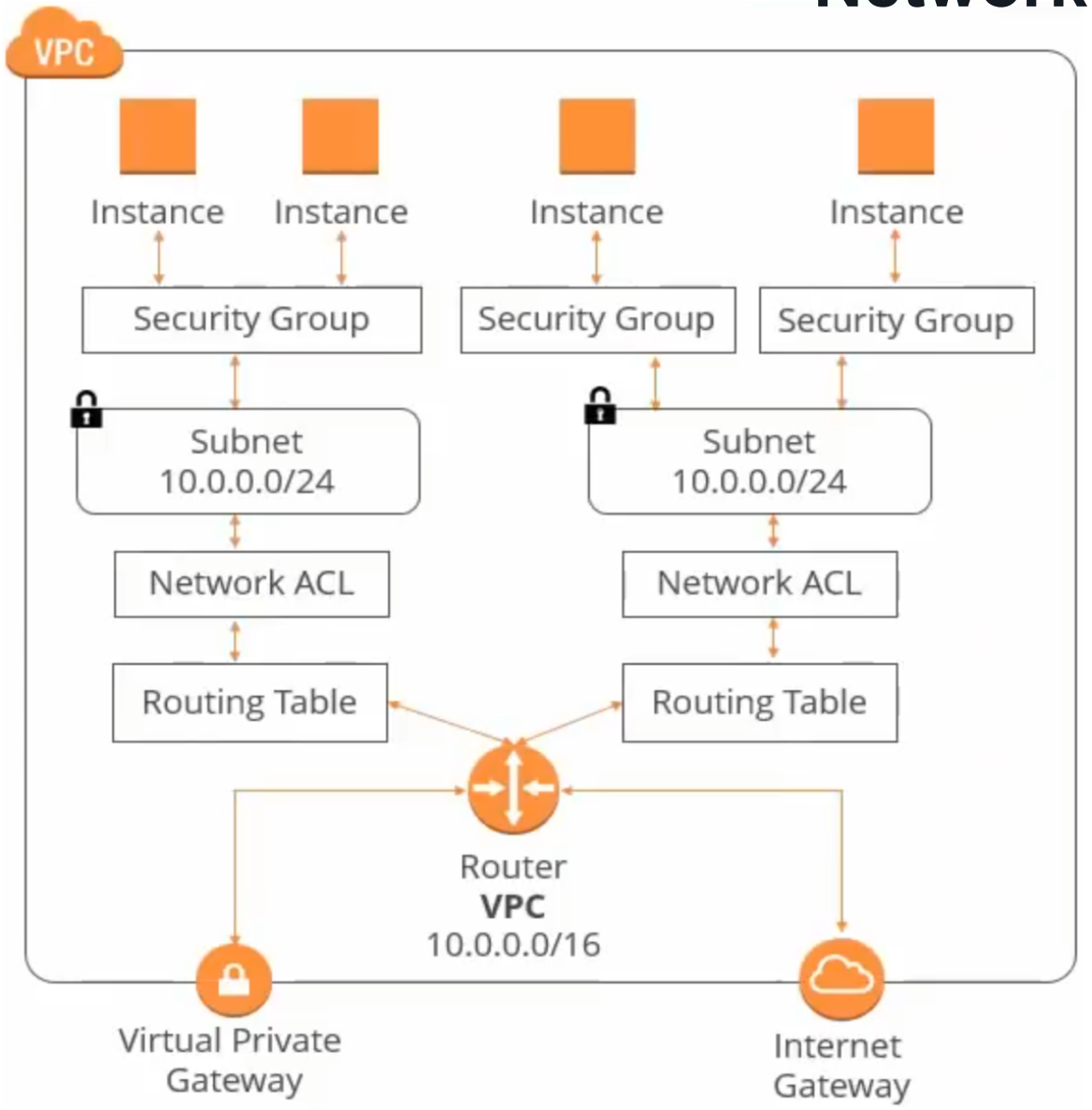
✓ Network ACL rules can be both permissive or non-permissive

  - ➢ Deny allows to blacklist traffic to the subnet

✓ By default, a new network ACLs deny all inbound and outbound traffic and it is not associated to any subnet

- Your VPC comes with a default network ACL
- Each subnet in your VPC must be associated with a network ACL (and only one)
  - The subnet is associated with the default network ACL, unless another network ACL is specified
  - Rules are evaluated in order, starting with the lowest numbered rule, to determine if traffic is allowed in and out of any subnet associated with the network ACL

Network ACL



Default network ACL

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

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

- Configured to allow all traffic in and out of the subnet to which is associated
- Each network ACL includes a rule (rule number = “*”) that cannot be modified or removed
- If a packet does not match any of the other rules, it is denied

VPC Best Practices

- Span you VPC across multiple AZ inside your region, using subnets
- Always use private and public subnets
  - Use private subnets to secure resources that need to be available to the Internet
- Use a NAT gateway to provide secure Internet access to the instances that reside in your private subnet
- Choose your CIDR blocks carefully
  - VPC can contain from 16 to 65536 IP addresses so you should choose you CIDR according how many addresses you may need for your organization
  - You can choose to create separate VPCs for development, staging, and production environments or create one VPC with separate subnets
- Be careful!! VPC has various limitations on the VPC components
- Use security groups and network ACLs to secure the traffic coming in and out of your VPC
  - Use security groups for white listing traffic and network ACLs for black listing traffic

VPC limits

- There is a limit for VPC resources per region in you AWS account
  - You can request an increase of these limits raising a ticket with AWS support

Resource	Default limit
VPCs per Region	5
Subnets per VPC	200

Resource	Default limit
Route tables per VPC	200
Routes per route table (non-propagated routes)	50

Resource	Default limit
Network ACLs per VPC	200
Rules per network ACL	20

Resource	Default limit
VPC security groups per Region	2500
Inbound or outbound rules per security group	60

- Other limits: https://docs.aws.amazon.com/vpc/latest/userguide/amazon-vpc-limits.html

VPC pricing

- Creation of most of the VPC components are free of charge
- Site-to-site VPN connection
  - Duration: charged for each VPN connection-hour that your VPN connection is provisioned and available. Partial VPN connection-hours are billed as a full hour
  - Data processing: data transfer for all data transferred via the VPN connection
- NAT Gateway
  - ✓ Duration: charged for each NAT gateway-hour that your gateway is provisioned and available. Partial NAT connection-hours are billed as a full hour
  - ✓ Data processing: for each Gbps processed through the NAT gateway regardless of the traffic's source or destination

https://aws.amazon.com/vpc/pricing/

Amazon ELB

Distributing load between AWS resources

Elastic Load Balancing (ELB)

- ELB automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions
- It can handle the varying load of your application traffic in a single AZ or across multiple AZs
- ELB offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your application fault tolerant:

✓Application load balancer (L7)
✓Network load balancer (L4)
✓Classic load balancer (L4 & L7)

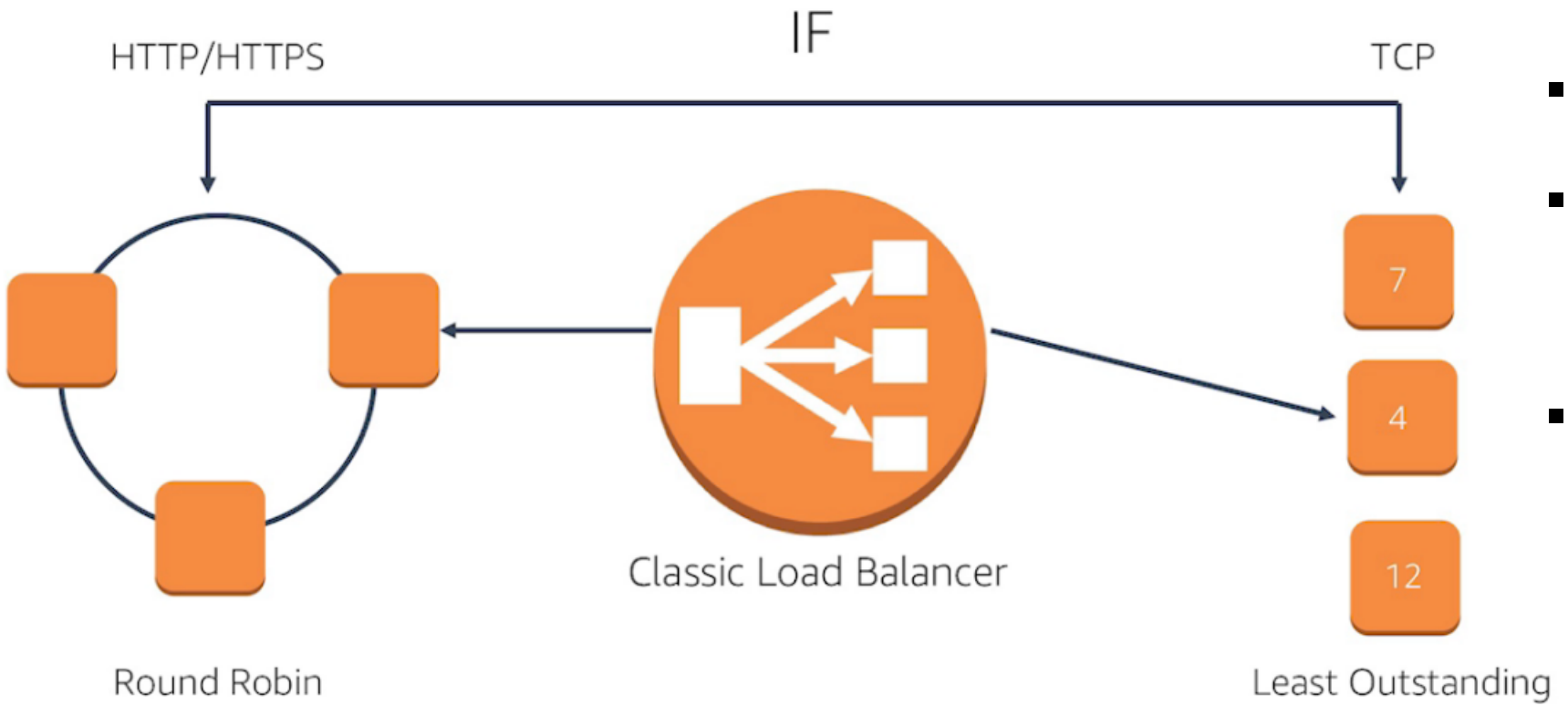
  - AWS is now recommending to use Application Load Balancer for L7 and Network Load Balancer for L4 when using VPC

Classic Load Balancer

- Distributes incoming application traffic across multiple EC2 instances
  - If set to balance load across multiple AZs, it increases the fault tolerance of your applications
- Serves as a single point of contact for clients
  - Increases availability of your application because you can add / remove instances from your load balancer, without disrupting the overall flow of requests to your application
- Operates at both the request level (L7) and connection level (L4)
- A listener checks for connection requests from clients, using the protocol and port you configure, and forwards requests to one or more registered instances
- You can configure health checks, used to monitor the health of the registered instances so that the load balancer only sends requests to the healthy instances
- Allows you monitoring:
  - View of HTTP responses
  - View number of healthy / unhealthy EC2 hosts behind the load balancer

Traffic distribution

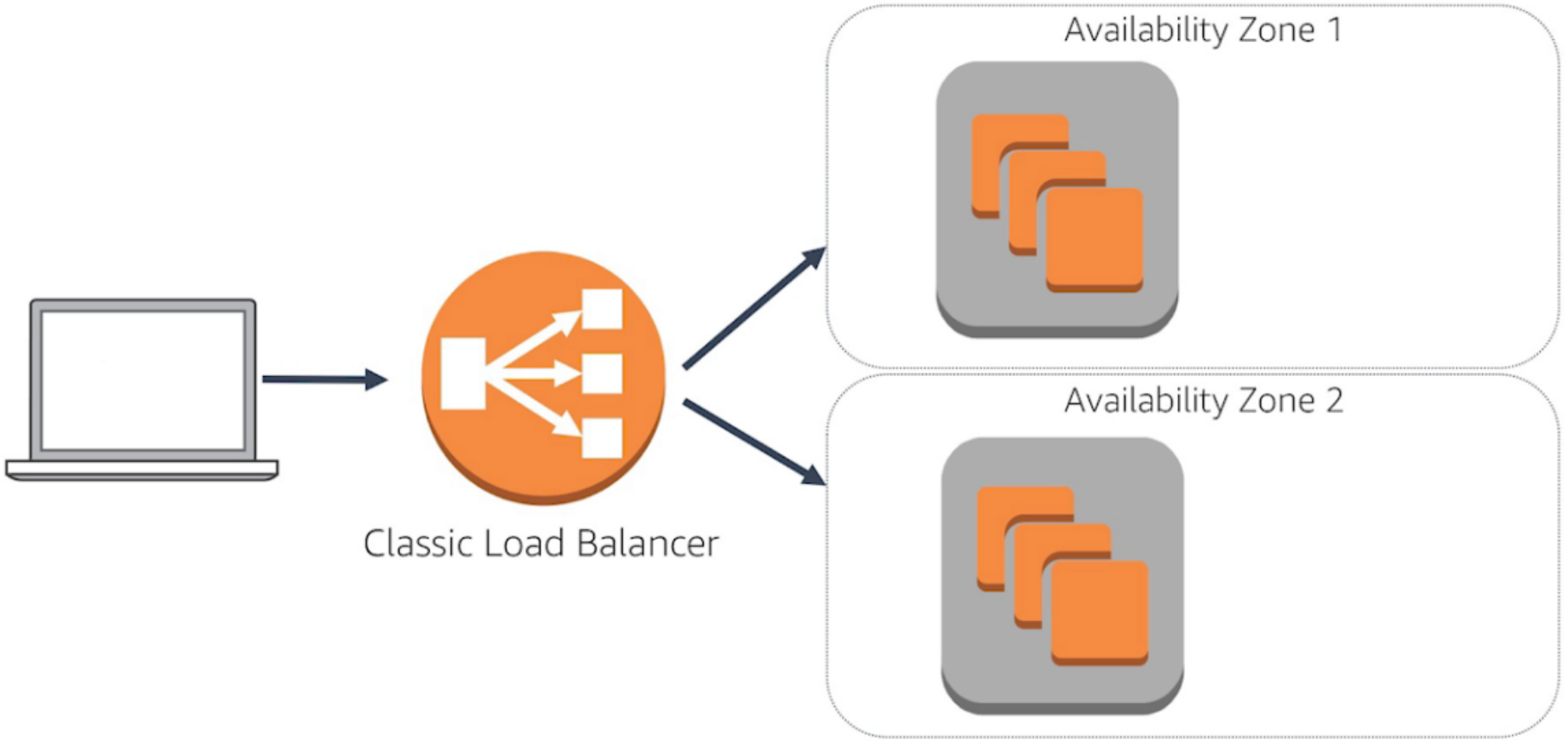
- The ability to distribute traffic depends on the type of request you are distributing



- Algorithms used:
- Round robin: instances are chosen in a rotating sequential way
- Least outstanding: the backend instance chosen is the one that, at that moment, has the lowest number of outstanding (pending, unfinished) requests

Traffic distribution

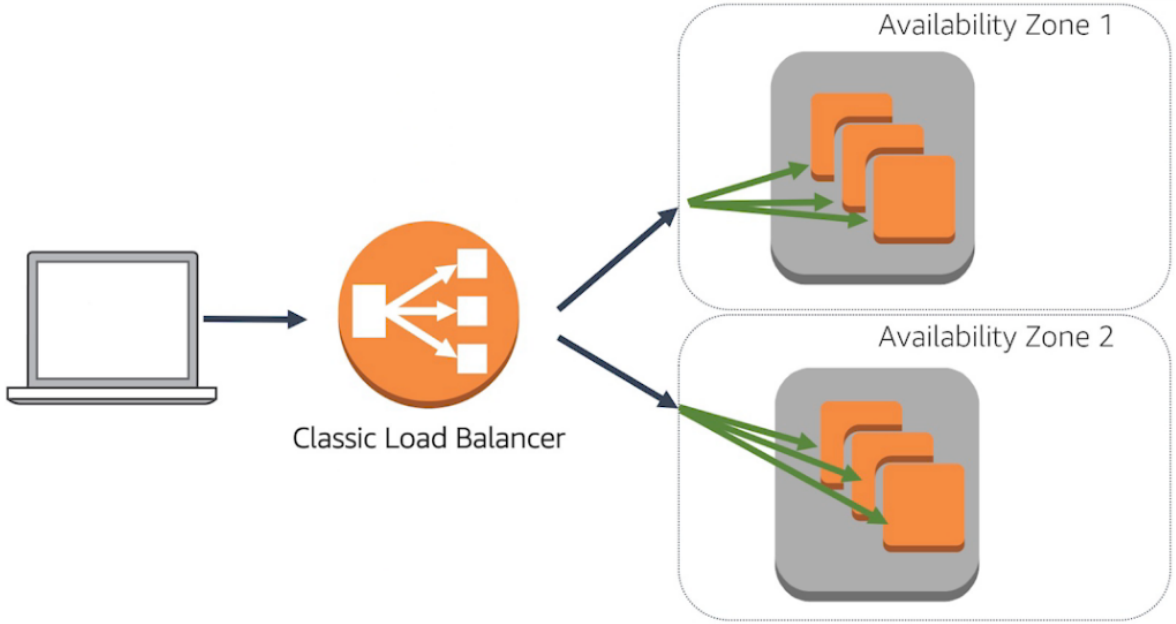
- Helps distributing traffic across multiple AZs

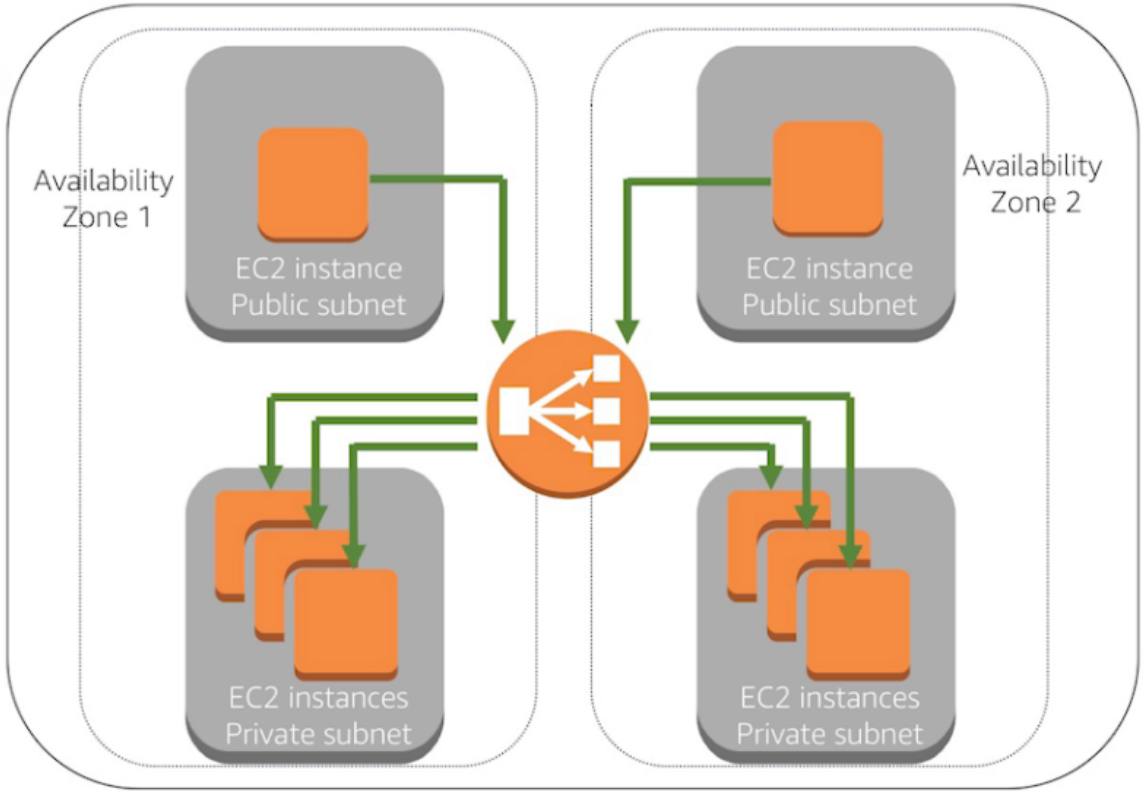


- It is important to keep approximately the same number of instances in each AZ, to better distribute traffic
- Enabled by default if you launch the load balancer using AWS management console
- It is necessary to enable it if you start load balancer via CLI or SDKs

Types of load balancers

- Internet-facing load balancer
  - Internal load balancer





Network Load Balancer

- Best suited for load balancing of TCP and TLS traffic where extreme performance is required (high throughput, low latency)
- Operating at the connection level (L4), it routes traffic to targets within VPC and is capable of handling millions of requests per second
- Targets:

✓EC2 instances: you use EC2 instances to serve TCP / TLS traffic

✓IP addresses: you can use the IP@ of the application backend

- Optimized to handle sudden and volatile traffic patterns
- The load balancer selects a target using flow hash algorithm, that routes traffic based on the flow of traffic and its origin
  - It uses the protocol, sources IP@ + port, destination IP @ + port, and TCP sequence number
  - Each individual TCP connection is routed to a single target for the life of the connection

Application Load Balancer

- Best suited for load balancing of HTTP and HTTPS traffic
- Operating at the individual request level (L7), it routes traffic to targets within VPC based on the content of the request
- Targets:

✓EC2 instances: you use EC2 instances to serve HTTP(S) requests

✓IP addresses: you can use the IP@ of the application backend

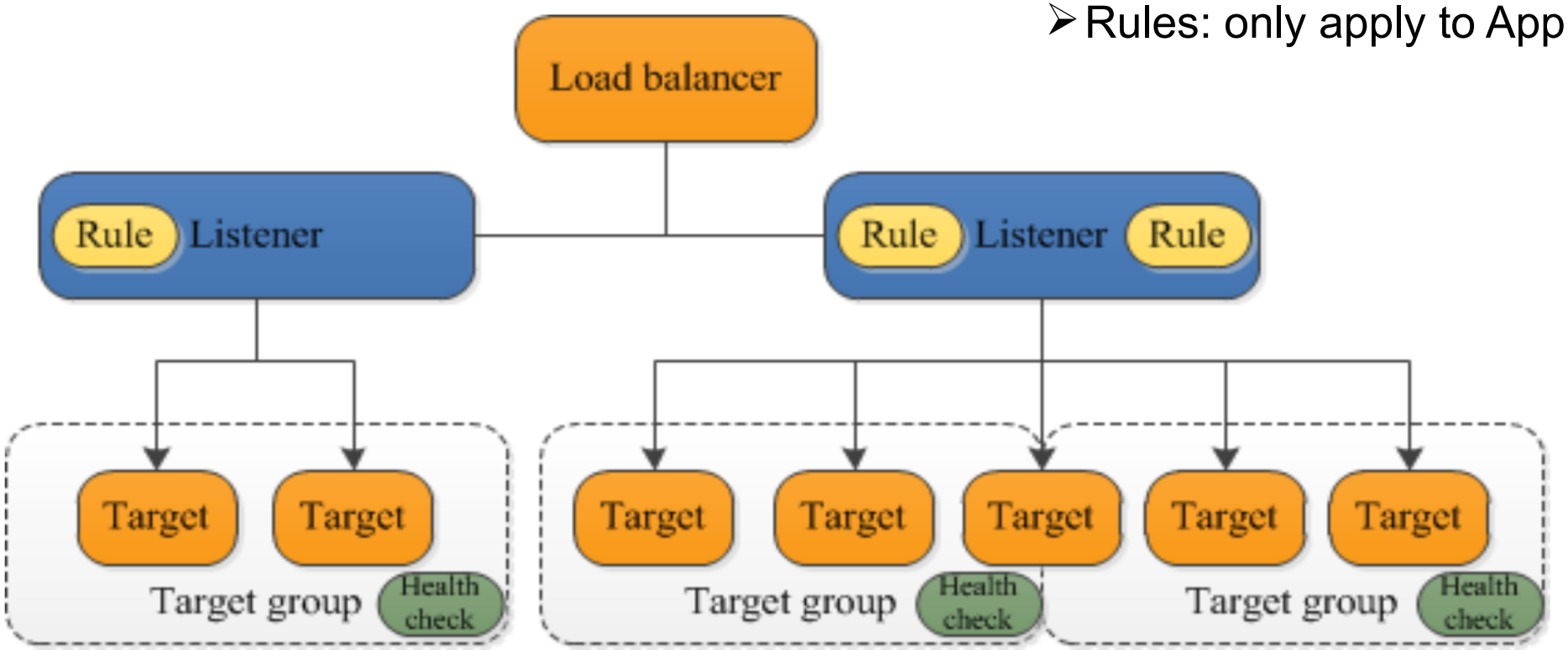
✓Lambda functions: you can use them to serve HTTP(S) requests, enabling users to access serverless applications from any HTTP client, including web browsers

- Provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers
- The load balancer selects a target using least outstanding algorithm

Net & App LB Components

- The load balancer distributes incoming traffic across multiple targets
- A listener checks for connection requests from clients and forwards requests to target groups
- Each target group routes requests to one or more registered targets

➢ Rules: only apply to Application LBs



Net & App LB Components

- Load balancer
  - Distributes incoming application traffic across multiple targets
  - You add one or more listeners to your load balancer
- Listener
  - Checks for connection requests from clients and forwards requests to one or more target groups. In the case of Application Load Balancer, based on the rules that you define
  - ✓ Each rule specifies a target group, condition, and priority. When the condition is met, the traffic is forwarded to the target group
- Target group
  - Routes requests to one or more registered targets. You can register a target with multiple target groups
  - You can configure health checks on a per target group basis. Health checks are performed on all targets registered to a target group that is specified in a listener rule for your load balancer

ELB pricing

- Application Load Balancer
  - You are charged for each hour or partial hour that an Application Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used per hour
- Network Load Balancer
  - You are charged for each hour or partial hour that a Network Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used by Network Load Balancer per hour
- Classic Load Balancer
  - You are charged for each hour or partial hour that a Classic Load Balancer is running and for each GB of data transferred through your load balancer

❖LCU (Load Balancer Capacity Unit) measures the dimensions on which the LB processes your traffic (averaged over an hour) using a combination of connections and bytes processed

https://aws.amazon.com/elasticloadbalancing/pricing/

Amazon API Gateway

Enabling access to code via APIs

API Gateway

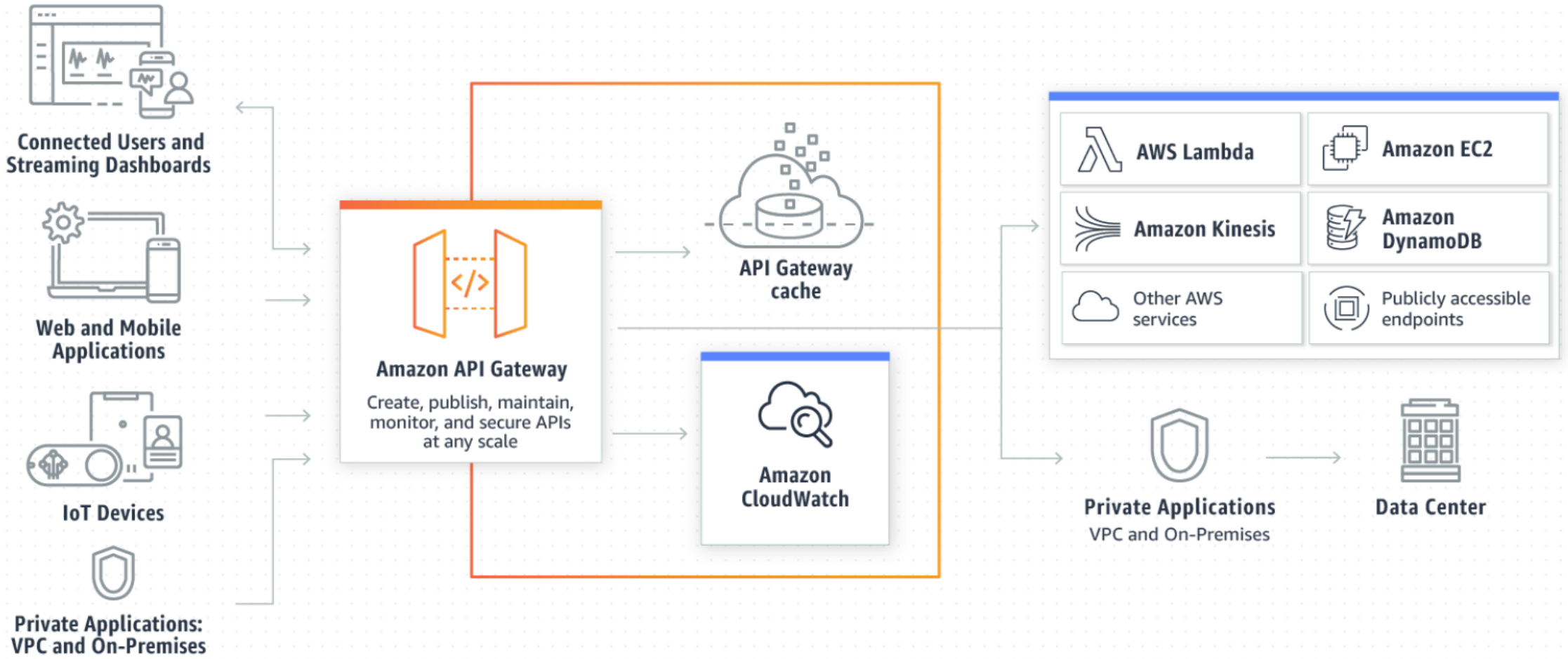
- A fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale
- With API Gateway you can create REST and WebSocket APIs to act as a “front door” for applications to:
  - access data
  - business logic
  - functionality from your backend services:
  - ✓ workloads running on EC2
  - ✓ code running on Lambda
  - ✓ any web application or real-time communication applications
- It handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management

API Gateway

- API developers can create APIs that access AWS or other web services as well as data stored in the AWS Cloud
- As an API Gateway API developer, you can create APIs for use in your own client applications (apps). Or you can make your APIs available to third-party app developers
- API Gateway creates REST APIs that:
  - Are HTTP-based
  - Adhere to the REST protocol, which enables stateless client-server communication
  - Implement standard HTTP methods such as GET, POST, PUT, PATCH and DELETE
- API Gateway creates WebSocket APIs that:
  - Adhere to the WebSocket protocol, which enables stateful, full-duplex communication between client and server
  - Route incoming messages and based on message content

API Gateway: how it works

- API Gateway works with a large set of applications and AWS services



API Gateway benefits

- Efficient API development
  - Run multiple versions of the same API simultaneously allowing you to quickly iterate, test, and release new versions
- Easy monitoring
  - Monitor performance metrics and information on API calls, data latency, and error rates from the API Gateway dashboard, which allows you to visually monitor calls to your services using CloudWatch
- Performance at any scale:
  - Provide end users with the lowest possible latency for API requests and responses by taking advantage of our edge locations using CloudFront
  - Throttle traffic and cache the output of API calls to ensure that backend operations withstand traffic spikes and backend systems are not unnecessarily called
- Flexible security controls
  - Authorize access to your APIs with IAM or Cognito (User Sign-Up, Sign-In, and Access Control)
  - If you use OAuth tokens or other authorization mechanisms, API Gateway can help you verify incoming requests by executing a Lambda authorizer from Lambda

API Gateway benefits

- Restful API endpoints
  - Create resource-based APIs and use API Gateway's data transformation capabilities to generate the requests in the language target services expect
  - Protect your existing services by enforcing throttling rules to ensure that your backend can withstand unpredictable spikes in traffic
- Serverless APIs
  - Create REST APIs using API Gateway that mobile and web applications can use to call publicly available AWS services through code running in Lambda
  - Lambda runs your code on a high-availability compute infrastructure, eliminating the need to provision, scale, or manage any servers
- WebSocket APIs
  - Build real-time two-way communication applications, such as chat apps and streaming dashboards, without having to provision or manage any servers or worry about connected users and devices
  - API Gateway maintains a persistent connection between clients, handles message transfer, and pushes data through backend servers.

API Gateway pricing

- HTTP / REST APIs
  - Pay for the API calls you receive and the amount of data transferred out
  - There are no data transfer out charges for private APIs, except PrivateLink charges
  - Optional data caching charged at an hourly rate, depending on the cache memory size
  - API Gateway provides a tiered pricing model, where you can reduce your cost as your API usage scales
- WebSocket APIs
  - Pay for messages sent and received and the total number of connection minutes
  - You may send messages up to 128KB and are metered by 32KB increments
- Additional charges apply if you use API Gateway in conjunction with other AWS services

https://aws.amazon.com/api-gateway/pricing/