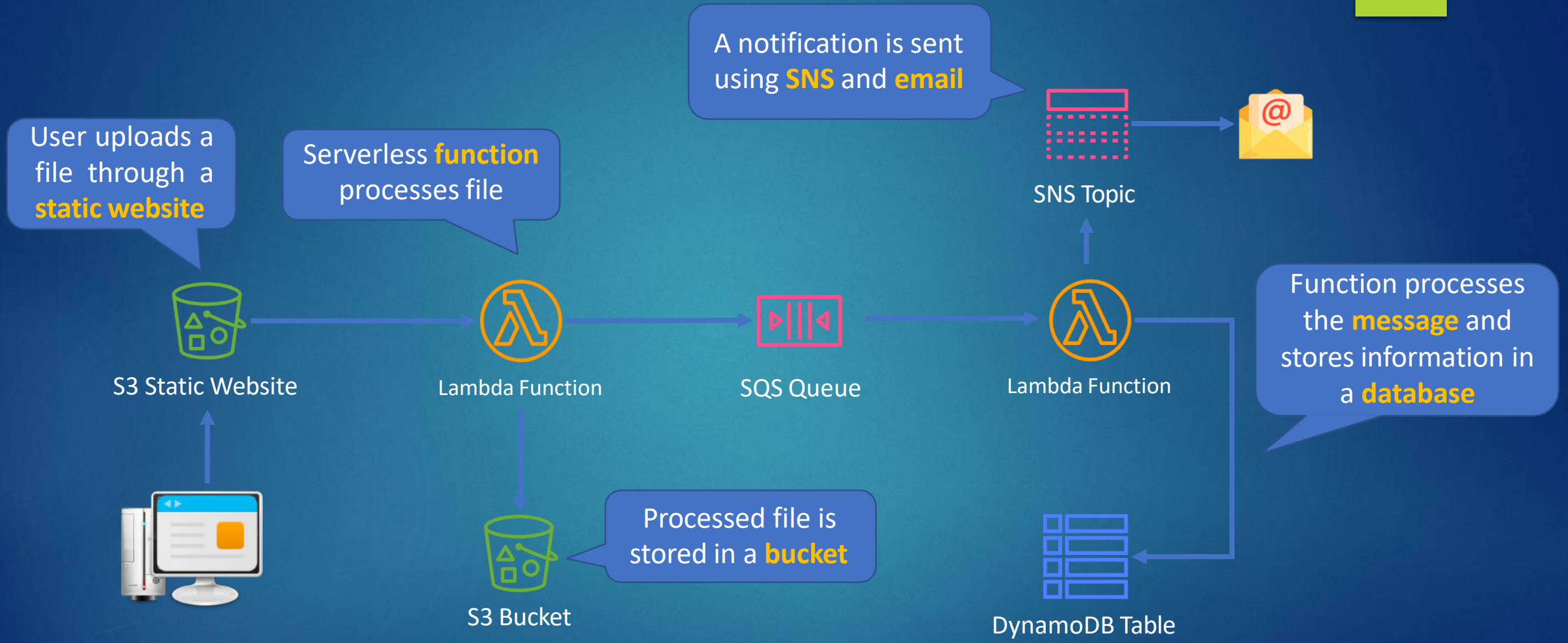
Serverless Services

Serverless Services



Serverless Services

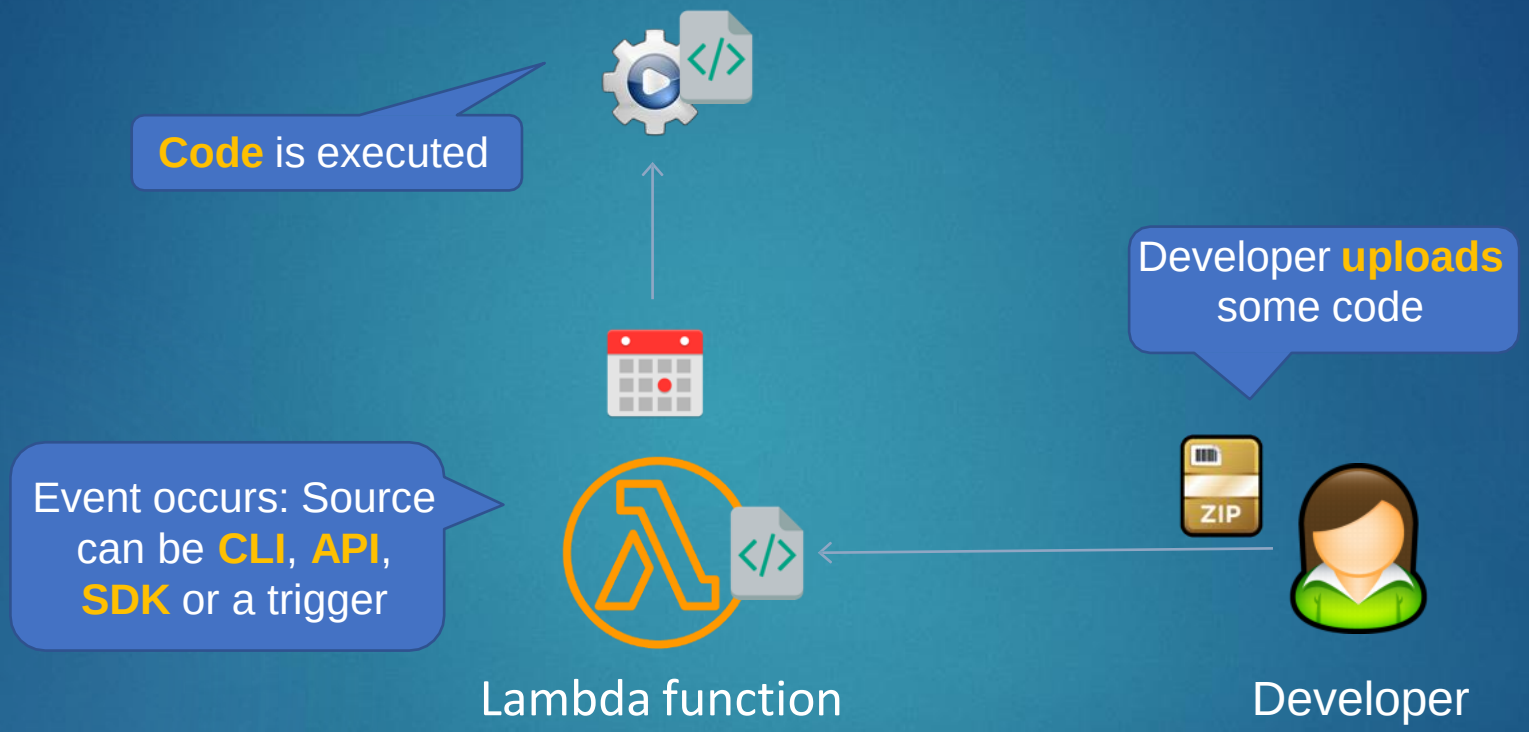
- With serverless there are **no instances** to manage
- You don’t need to provision hardware
- There is no management of operating systems or software
- Capacity provisioning and patching is handled automatically
- Provides automatic scaling and high availability
- Can be very cheap!

Serverless Services

- Serverless services include:
  - AWS Lambda
  - AWS Fargate
  - Amazon EventBridge
  - AWS Step Functions
  - Amazon SQS
  - Amazon SNS
  - Amazon API Gateway
  - Amazon S3
  - Amazon DynamoDB

AWS Lambda Functions

AWS Lambda Functions



AWS Lambda Functions

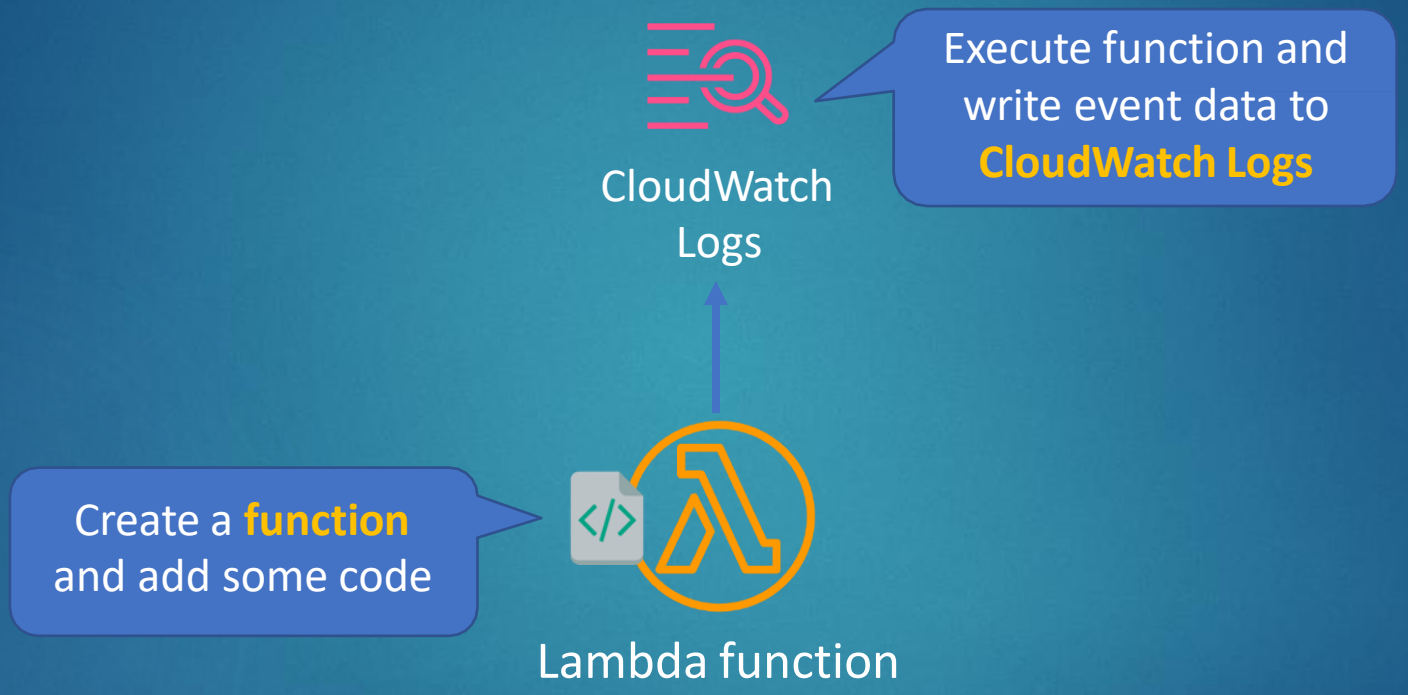
- AWS Lambda executes code only when needed and scales automatically
- You pay only for the compute time you consume (you pay nothing when your code is not running)
- Benefits of AWS Lambda:
  - No servers to manage
  - Continuous scaling
  - Millisecond billing
  - Integrates with almost all other AWS services

AWS Lambda Functions

- Primary use cases for AWS Lambda:
  - Data processing
  - Real-time file processing
  - Real-time stream processing
  - Build serverless backends for web, mobile, IOT, and 3rd party API requests

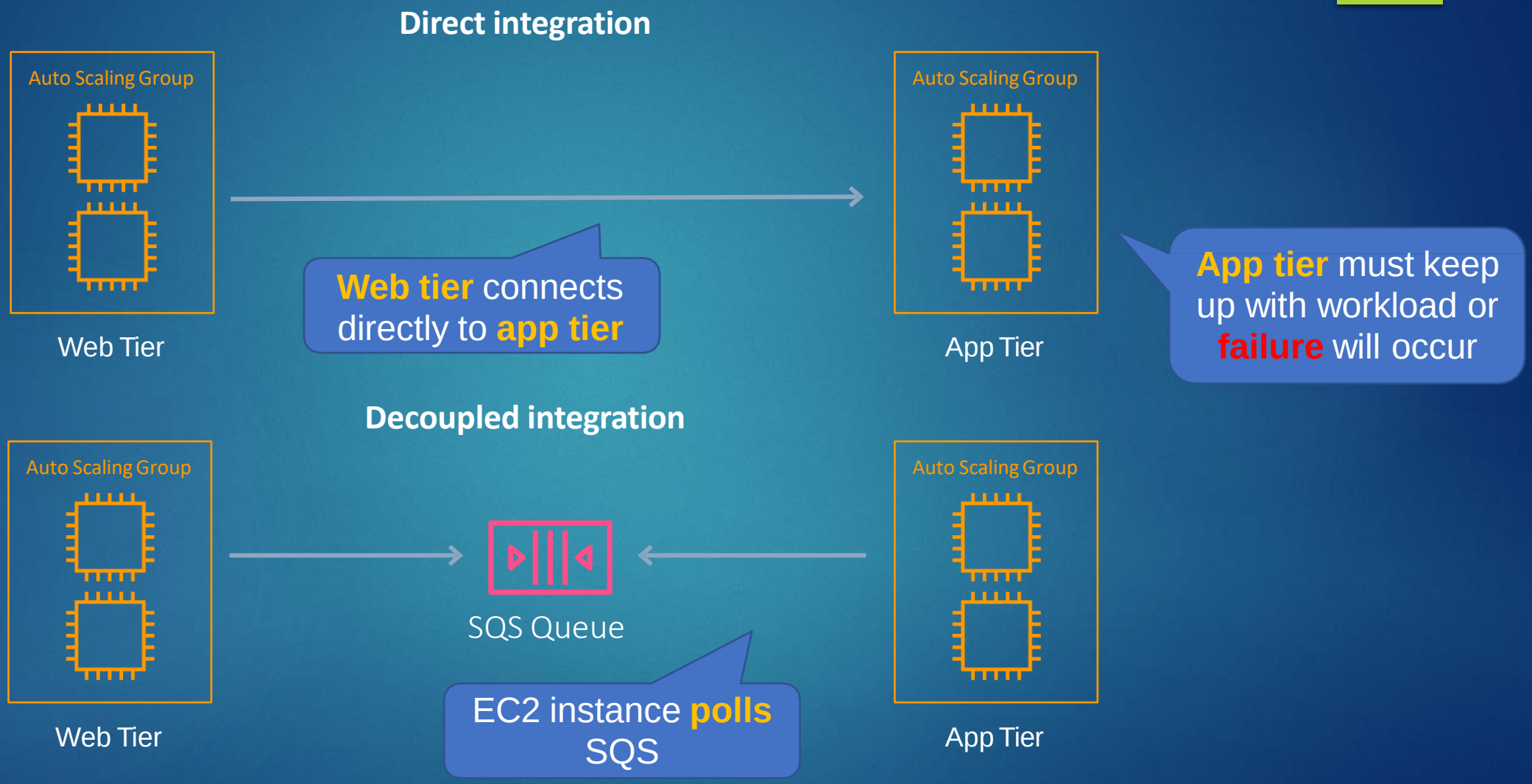
Create a Simple Lambda Function

Create a Simple Lambda Function



Application Integration Services

Amazon Simple Queue Service (SQS)



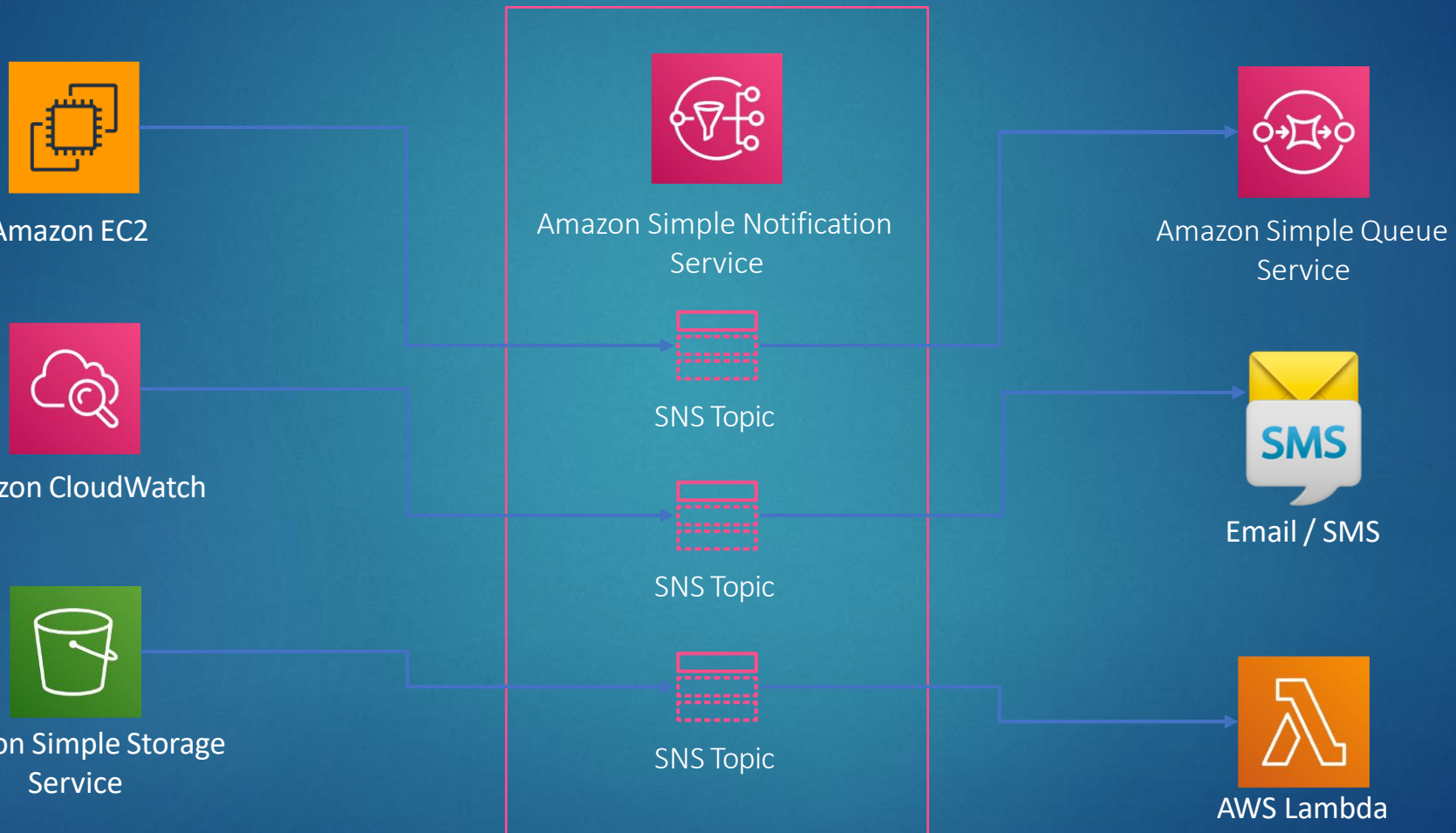
Amazon SQS

- SQS offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers
- SQS is used for distributed/decoupled applications
- SQS uses a message-oriented API
- SQS uses pull based (polling) not push based

Amazon MQ

- Message broker service
- Similar to Amazon SQS
- Based on Apache Active MQ and RabbitMQ
- Used when customers require industry standard APIs and protocols
- Useful when migrating existing queue-based applications into the cloud

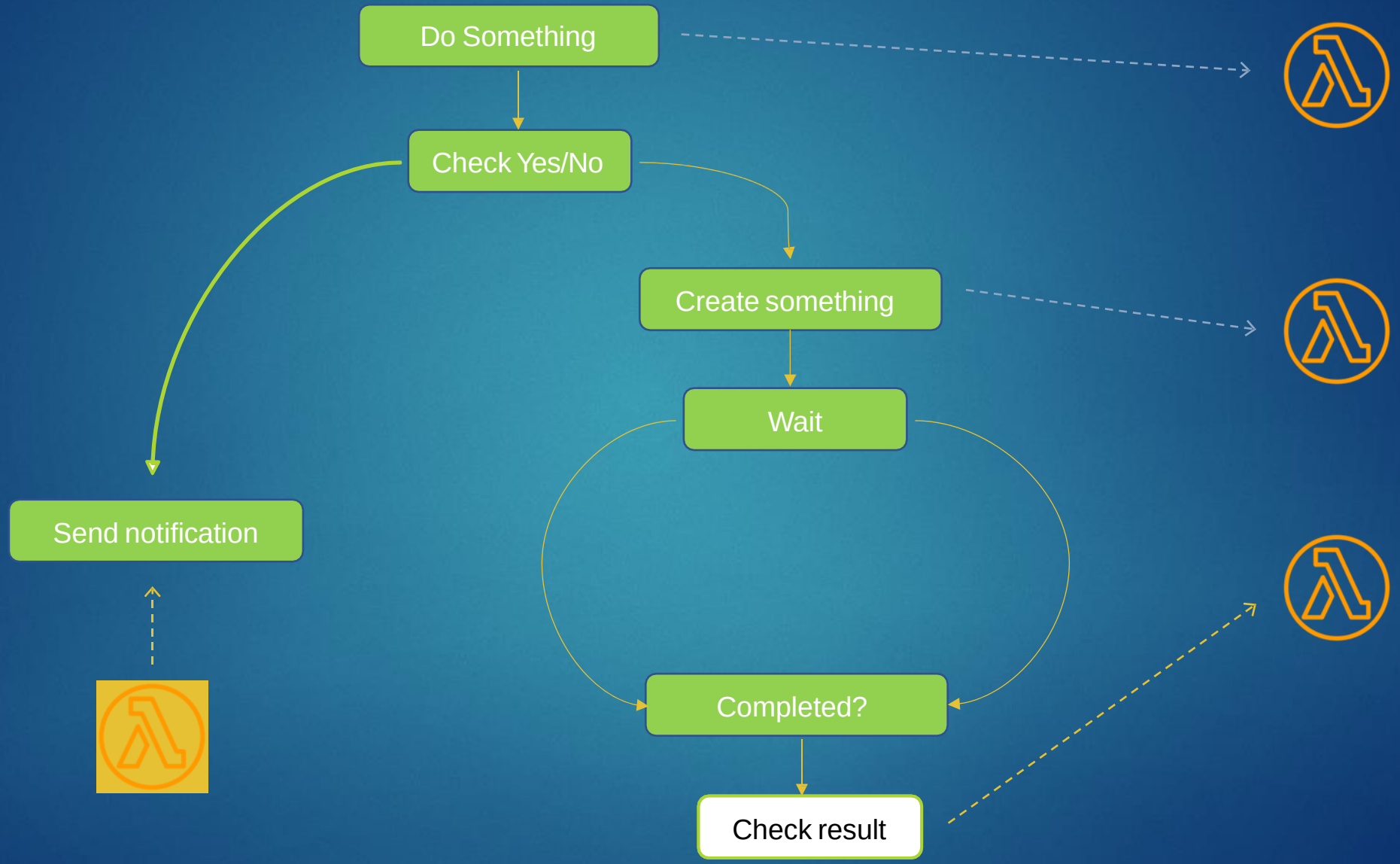
Amazon Simple Notification Service (SNS)



Amazon SNS

- Amazon SNS is used for building and integrating loosely-coupled, distributed applications
- Provides instantaneous, push-based delivery (no polling)
- Uses simple APIs and easy integration with applications
- Offered under an inexpensive, pay-as-you-go model with no up-front costs

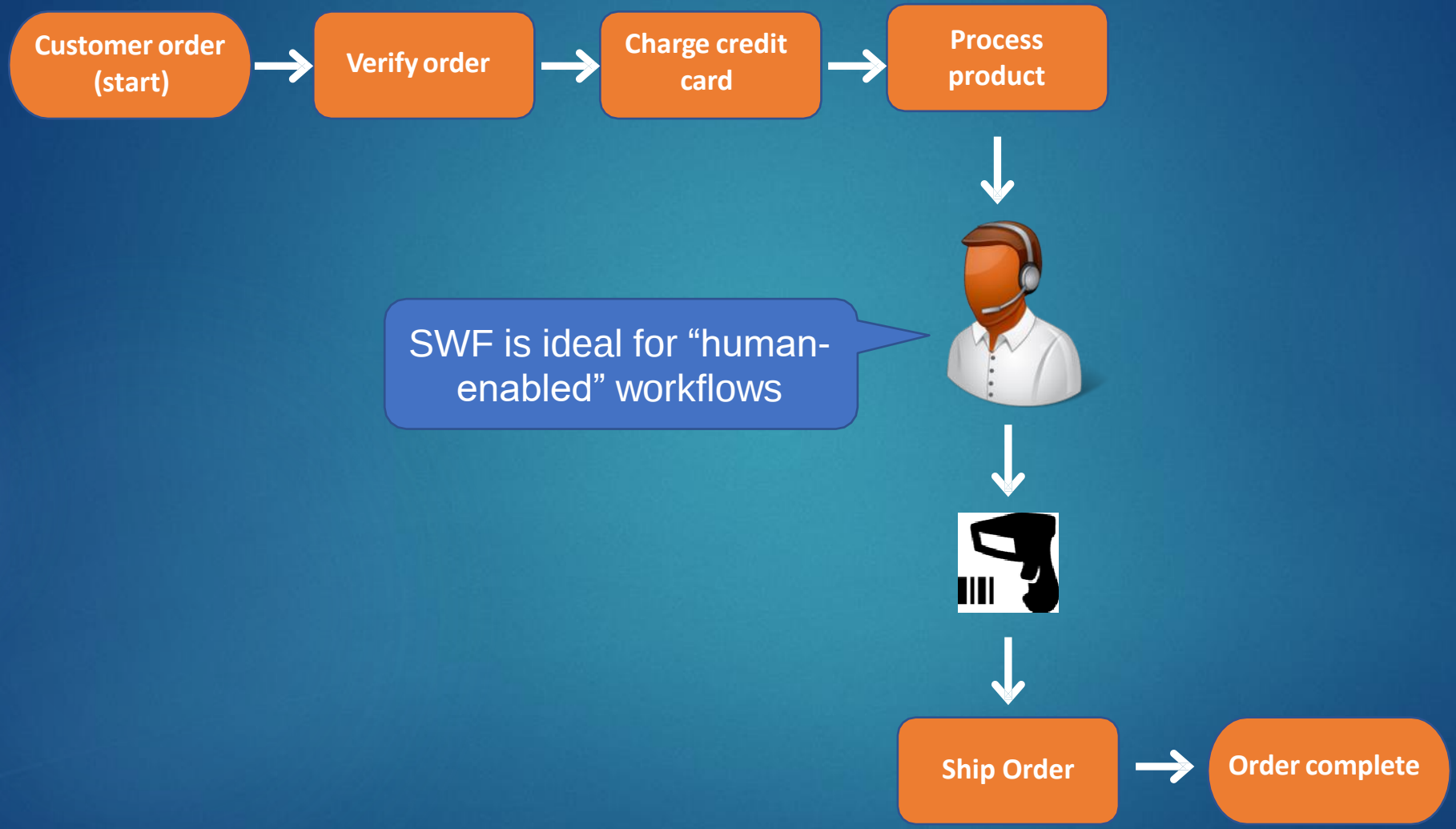
AWS Step Functions



AWS Step Functions

- AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow
- You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion

AWS Simple Workflow Service (SWF)



Amazon SWF

- Amazon Simple Workflow Service (SWF) is a web service that makes it easy to coordinate work across distributed application components
- Create distributed asynchronous systems as workflows
- Best suited for human-enabled workflows like an order fulfilment system or for procedural requests
- AWS recommends that for new applications customers consider Step Functions instead of SWF

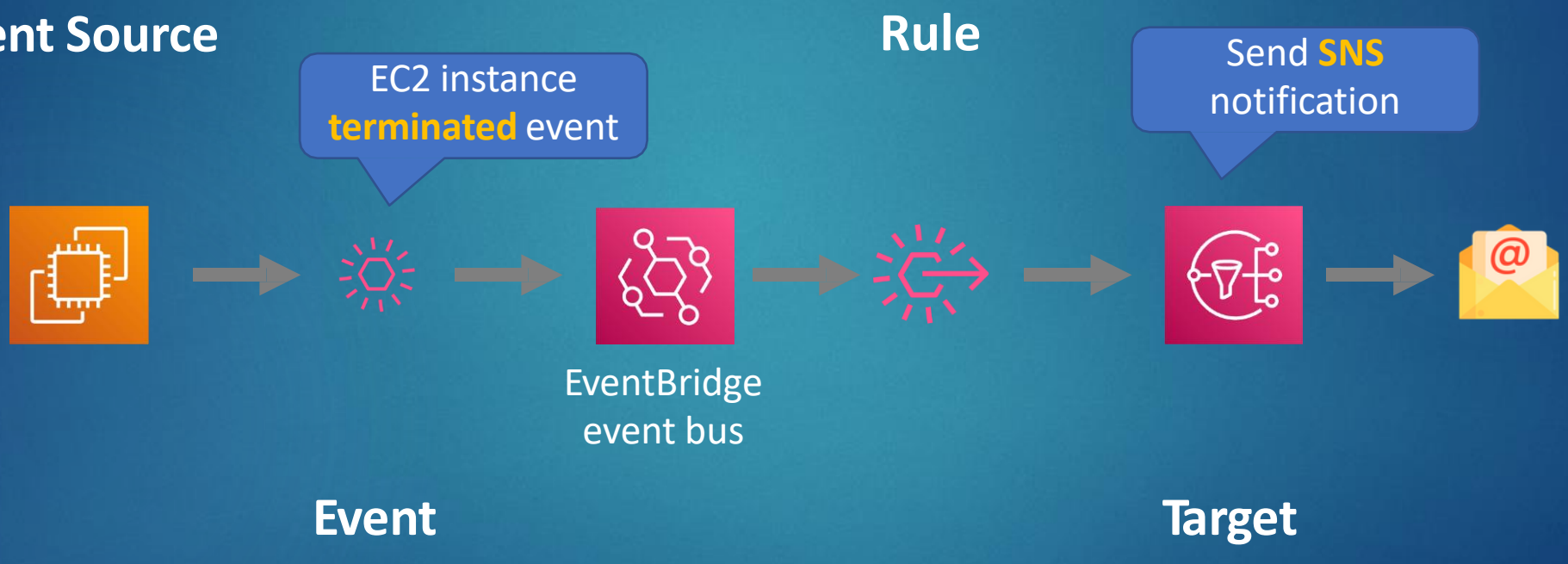
Application Integration Services Comparison

Service	What it does	Example use cases
Simple Queue Service	Messaging queue; store and forward patterns	Building distributed / decoupled applications
Simple Notification Service	Set up, operate, and send notifications from the cloud	Send email notification when CloudWatch alarm is triggered
Step Functions	Out-of-the-box coordination of AWS service components with visual workflow	Order processing workflow
Simple Workflow Service	Need to support external processes or specialized execution logic	Human-enabled workflows like an order fulfilment system or for procedural requests Note: AWS recommends that for new applications customers consider Step Functions instead of SWF
Amazon MQ	Message broker service for Apache Active MQ and RabbitMQ	Need a message queue that supports industry standard APIs and protocols; migrate queues to AWS

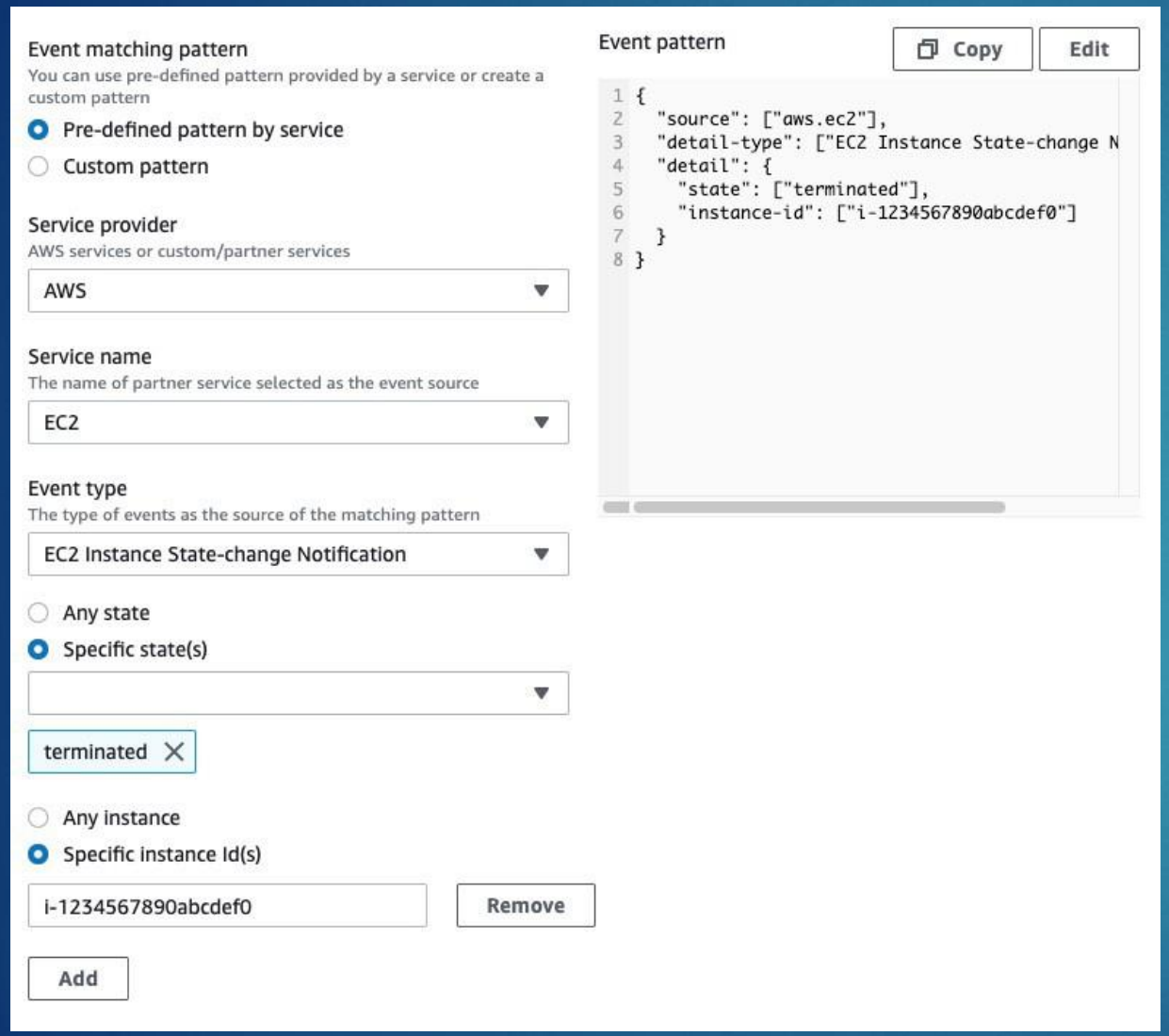
Amazon EventBridge / CloudWatch Events

Amazon EventBridge

EventBridge used to be known as **CloudWatch Events**



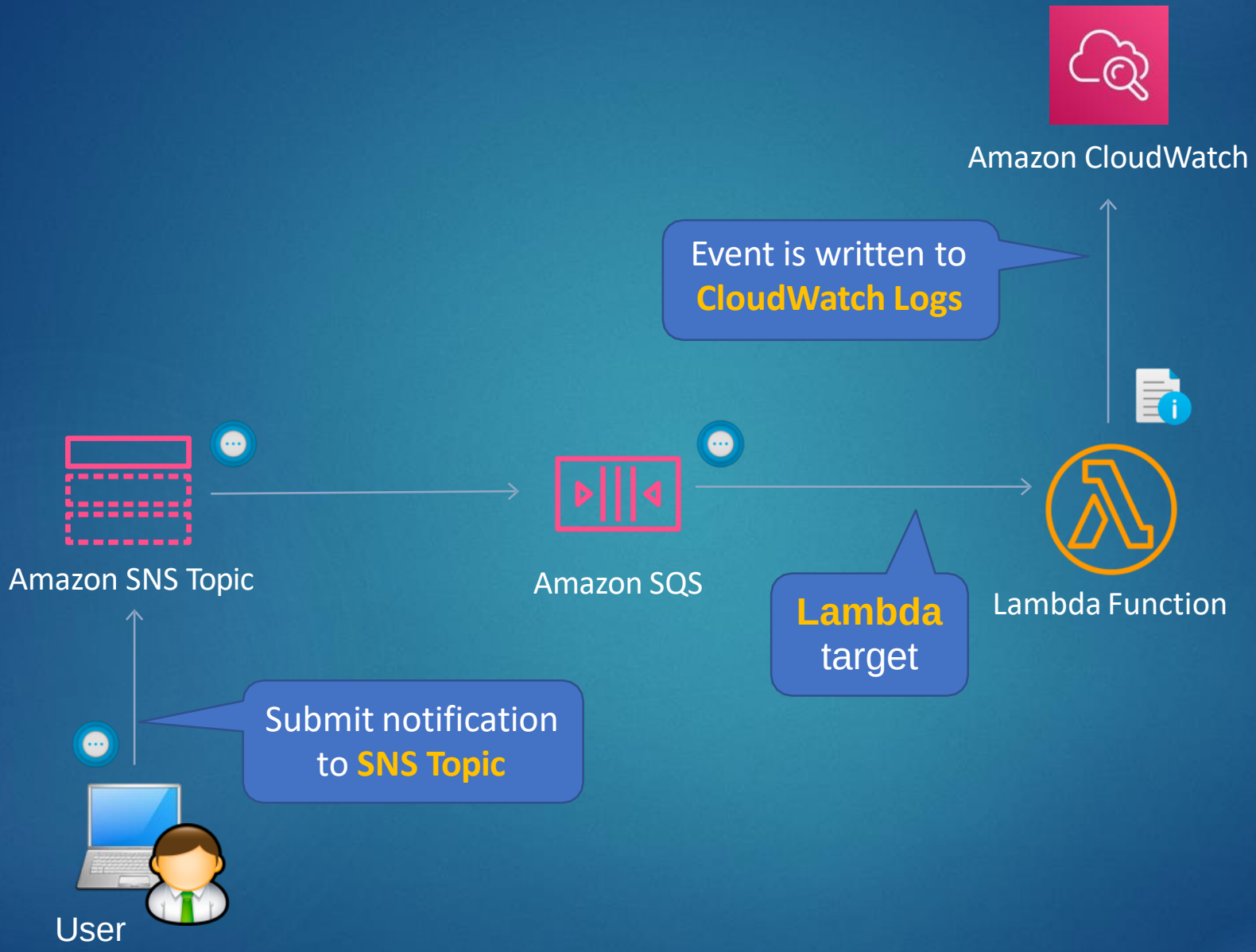
Amazon EventBridge



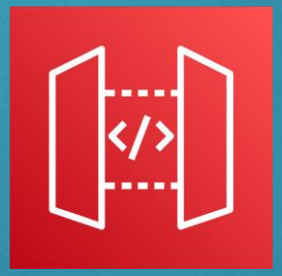

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/i-1234567890abcdef0"
 ],
 "detail": {
   "instance-id": "i-1234567890abcdef0",
   "state": "terminated"
   }
 }
```

Create an Event-Driven Application

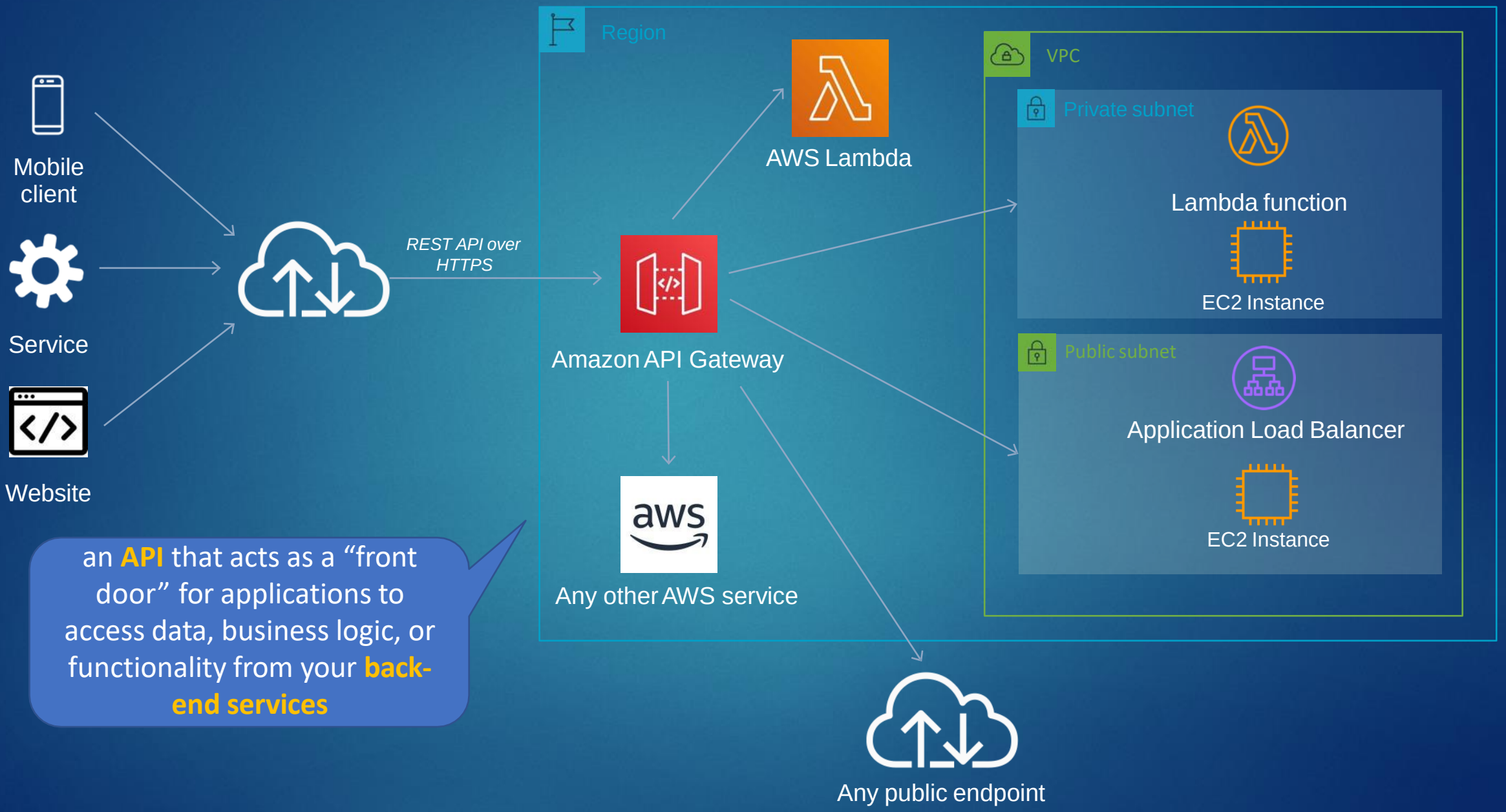
Simple Event-Driven Application



Amazon API Gateway



Amazon API Gateway



Amazon Virtual Private Cloud (VPC)

Amazon VPC

A **VPC** is a logically isolated portion of the AWS cloud within a region

Subnets are created within **AZs**

Main Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

The **route table** is used to configure the VPC router

You can launch **EC2 instances** into your VPC subnets

An **Internet Gateway** is used to connect to the Internet

Amazon VPC


Each VPC has a different block of IP addresses
CIDR stands for Classless Interdomain Routing
Region
VPC
CIDR 10.0.0.0/16
Availability Zone
Public subnet
10.0.0.0/20
Private subnet
10.0.32.0/20
Availability Zone
Public subnet
10.0.16.0/20
Private subnet
10.0.48.0/20
VPC
CIDR 10.1.0.0/16
Availability Zone
Public subnet
10.1.0.0/20
Private subnet
10.1.32.0/20
Availability Zone
Public subnet
10.1.16.0/20
Private subnet
10.1.48.0/20
Each subnet has a block of IP addresses from the CIDR block
You can create multiple VPCs within each region

Amazon VPC

VPC Component	What it is
Virtual Private Cloud (VPC)	A logically isolated virtual network in the AWS cloud
Subnet	A segment of a VPC's IP address range where you can place groups of isolated resources
Internet Gateway/Egress-only Internet Gateway	The Amazon VPC side of a connection to the public Internet for IPv4/IPv6
Router	Routers interconnect subnets and direct traffic between Internet gateways, virtual private gateways, NAT gateways, and subnets
Peering Connection	Direct connection between two VPCs
VPC Endpoints	Private connection to public AWS services
NAT Instance	Enables Internet access for EC2 instances in private subnets managed by you)
NAT Gateway	Enables Internet access for EC2 instances in private subnets (managed by AWS)
Virtual Private Gateway	The Amazon VPC side of a Virtual Private Network (VPN) connection
Customer Gateway	Customer side of a VPN connection
AWS Direct Connect	High speed, high bandwidth, private network connection from customer to aws
Security Group	Instance-level firewall
Network ACL	Subnet-level firewall

Amazon VPC

- A virtual private cloud (VPC) is a virtual network dedicated to your AWS account
- Analogous to having your own DC inside AWS
- It is logically isolated from other virtual networks in the AWS Cloud
- Provides complete control over the virtual networking environment including selection of IP ranges, creation of subnets, and configuration of route tables and gateways
- You can launch your AWS resources, such as Amazon EC2 instances, into your VPC

Amazon VPC

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16
- A VPC spans all the Availability Zones in the region
- You have full control over who has access to the AWS resources inside your VPC
- By default you can create up to 5 VPCs per region
- A default VPC is created in each region with a subnet in each AZ

Create a Custom VPC

Create a Custom VPC

Main Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

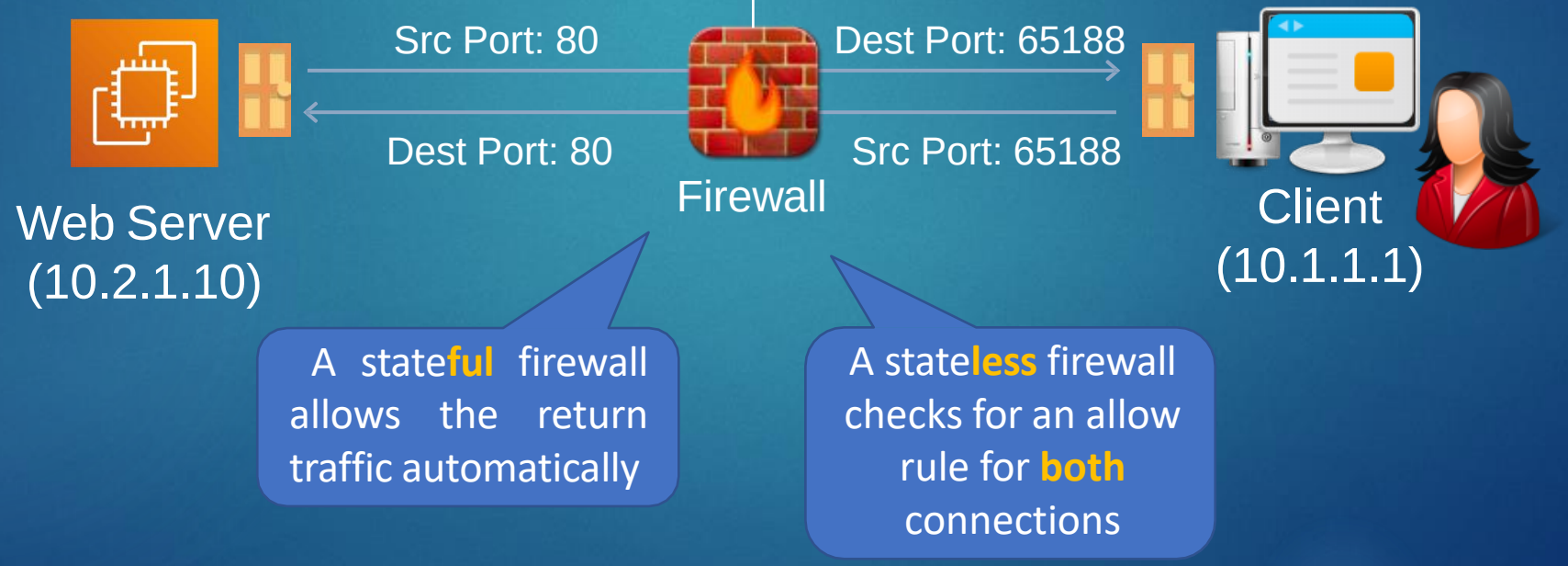
Private Route Table

Destination	Target
10.0.0.0/16	Local

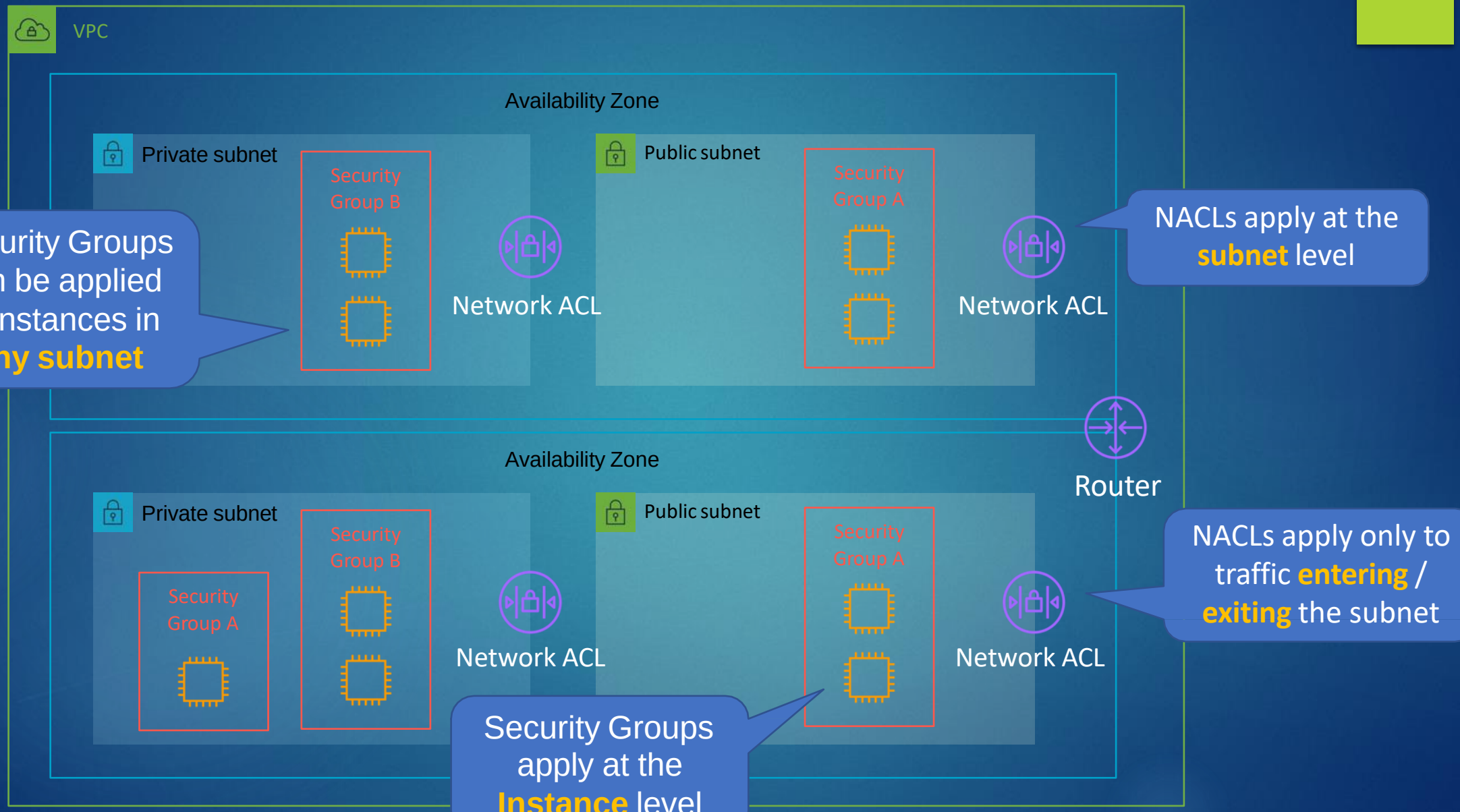
Security Groups and Network ACLs

Stateful vs Stateless Firewalls

PROTOCOL	SOURCE IP	DESTINATION IP	SOURCE PORT	DESTINATION PORT
HTTP	10.1.1.1	10.2.1.10	65188	80
HTTP	10.2.1.10	10.1.1.1	80	65188



Security Groups and Network ACLs



Security Group Rules

Security groups support **allow** rules only

Inbound rules

Separate rules are defined for outbound traffic

Type	Protocol	Port range	Source
SSH	TCP	22	0.0.0.0/0
RDP	TCP	3389	0.0.0.0/0
RDP	TCP	3389	::/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0
All ICMP - IPv4	ICMP	All	0.0.0.0/0

A source can be an **IP address** or **security group ID**

Network ACLs

Inbound Rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

Outbound Rules

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

NACLs have an explicit deny

Rules are processed in order

Configure Security Groups and NACLs

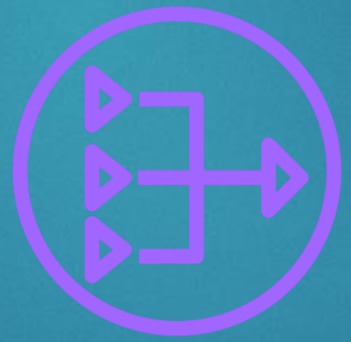
Public, Private and Elastic IP Addresses

Public, Private and Elastic IP addresses

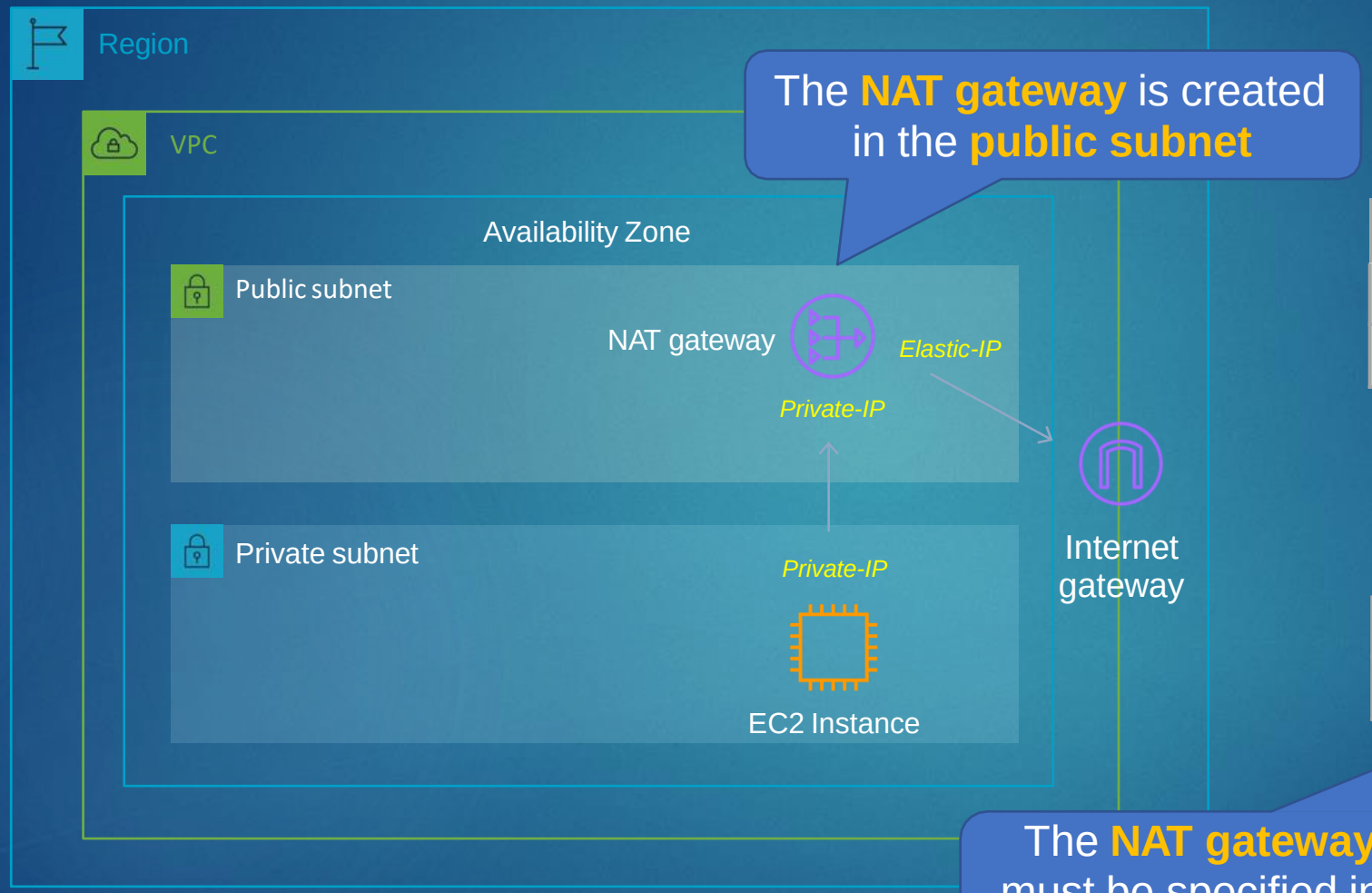
Name	Description
Public IP address	Lost when the instance is stopped Used in Public Subnets No charge Associated with a private IP address on the instance Cannot be moved between instances
Private IP address	Retained when the instance is stopped Used in Public and Private Subnets
Elastic IP address	Static Public IP address You are charged if not used Associated with a private IP address on the instance Can be moved between instances and Elastic Network Adapters

Working with IP Addresses

NAT Gateways and NAT Instances



NAT Gateways



The **NAT gateway** is created in the **public subnet**

Main Route Table

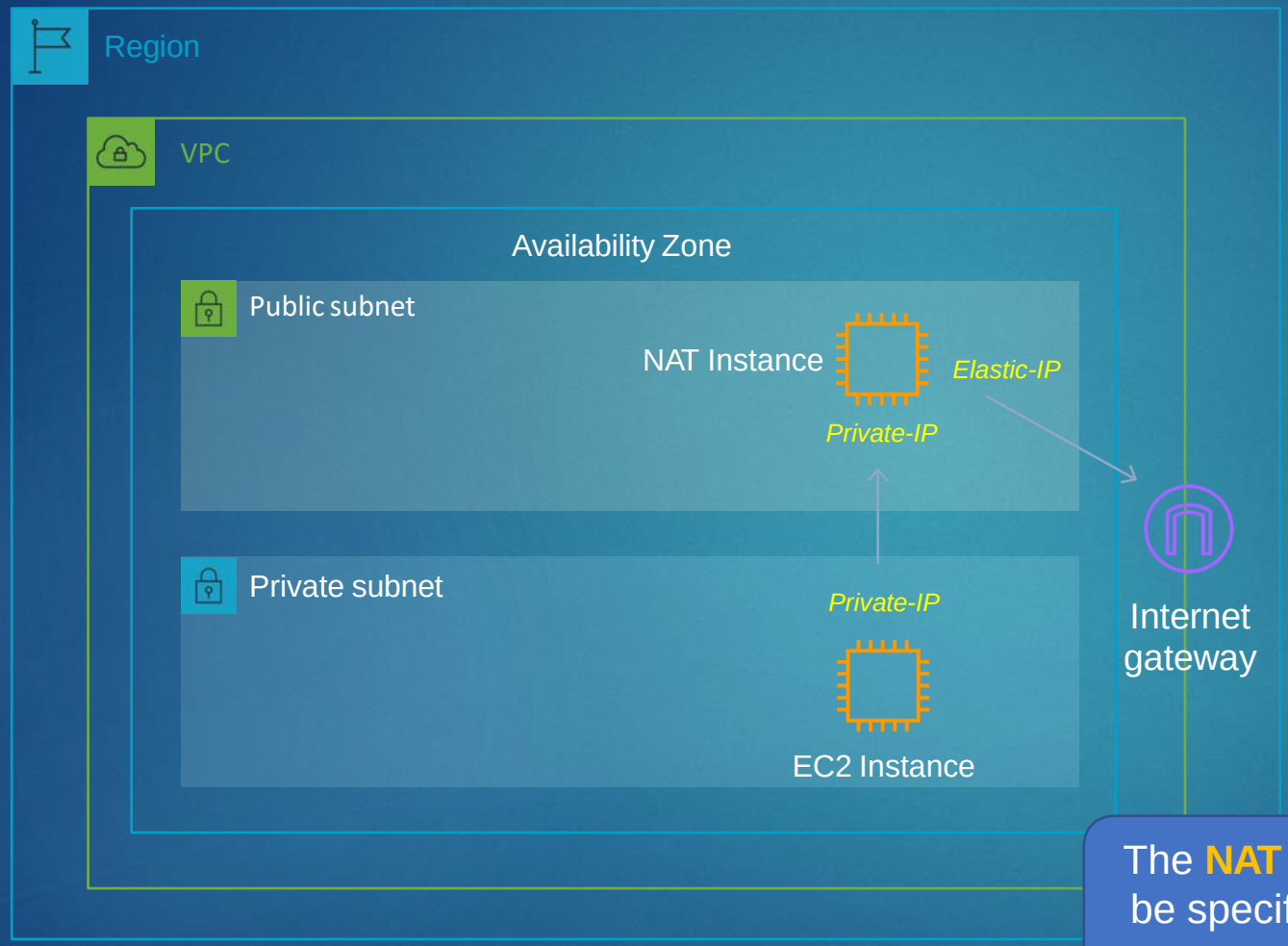
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	nat-gateway-id

The **NAT gateway ID** must be specified in the **private subnet** RT

NAT Instances



Main Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	nat-instance-id

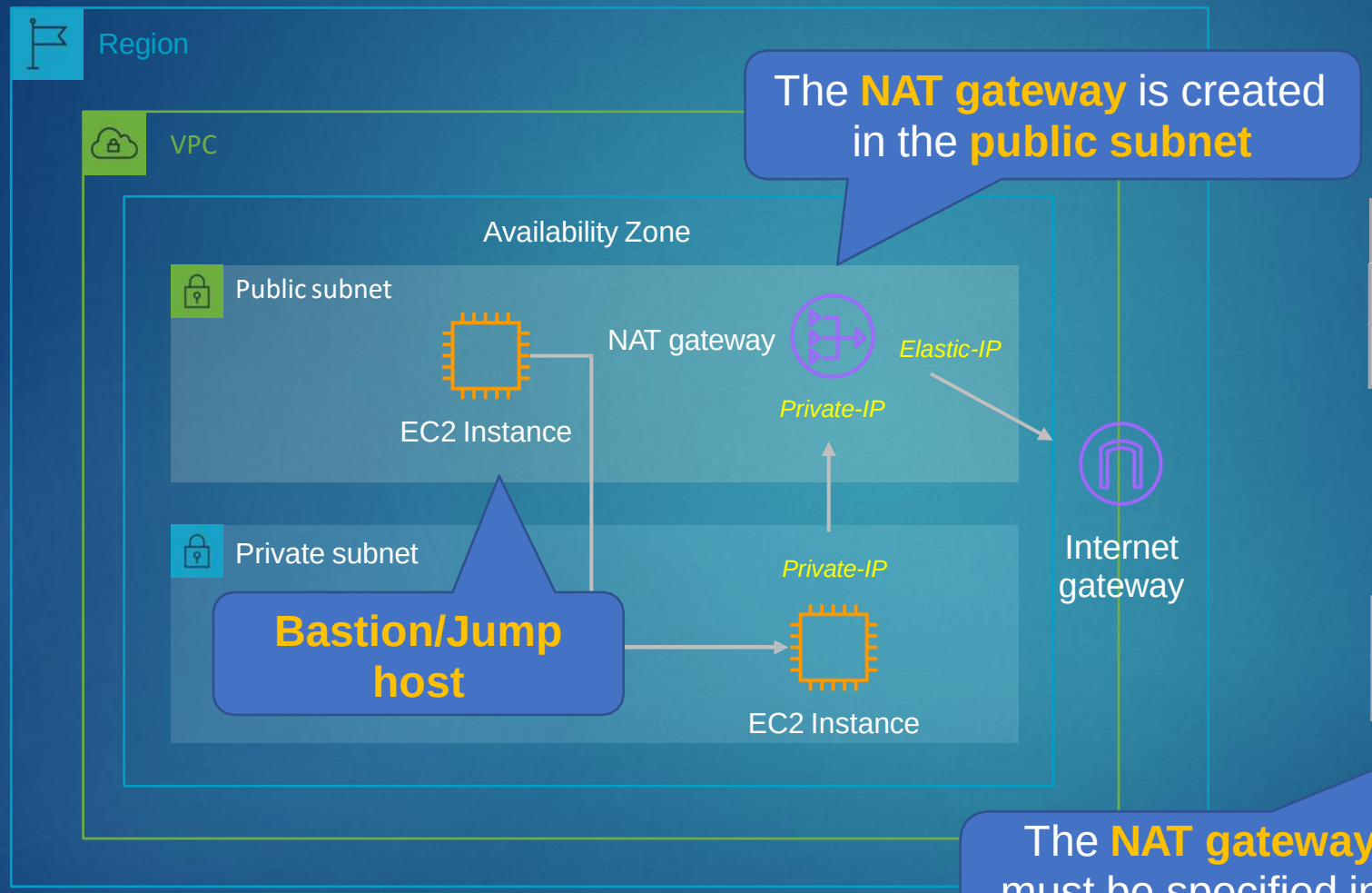
The **NAT instance ID** must be specified in the **private subnet** RT

NAT Instance vs NAT Gateway

NAT Instance	NAT Gateway
Managed by you (e.g. software updates)	Managed by AWS
Scale up (instance type) manually and use enhanced networking	Elastic scalability up to 45 Gbps
No high availability – scripted/auto-scaled HA possible using multiple NATs in multiple subnets	Provides automatic high availability within an AZ and can be placed in multiple AZs

Deploy a NAT Gateway

NAT Gateways



The **NAT gateway** is created in the **public subnet**

Main Route Table

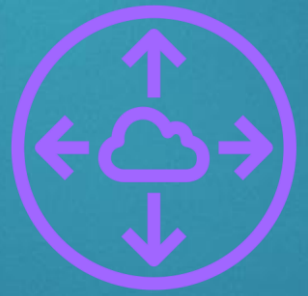
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Private Route Table

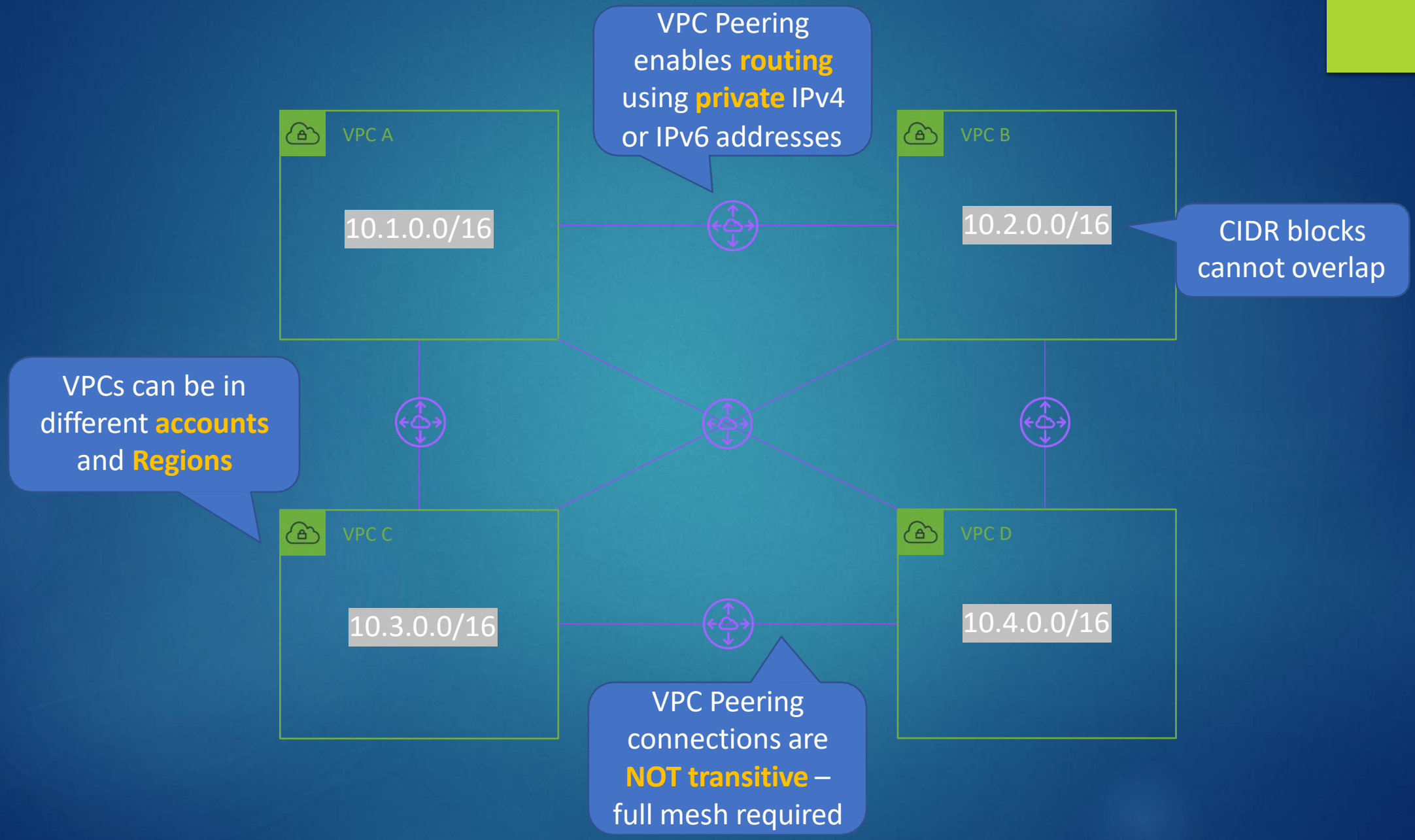
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	nat-gateway-id

The **NAT gateway ID** must be specified in the **private subnet** RT

Amazon VPC Peering



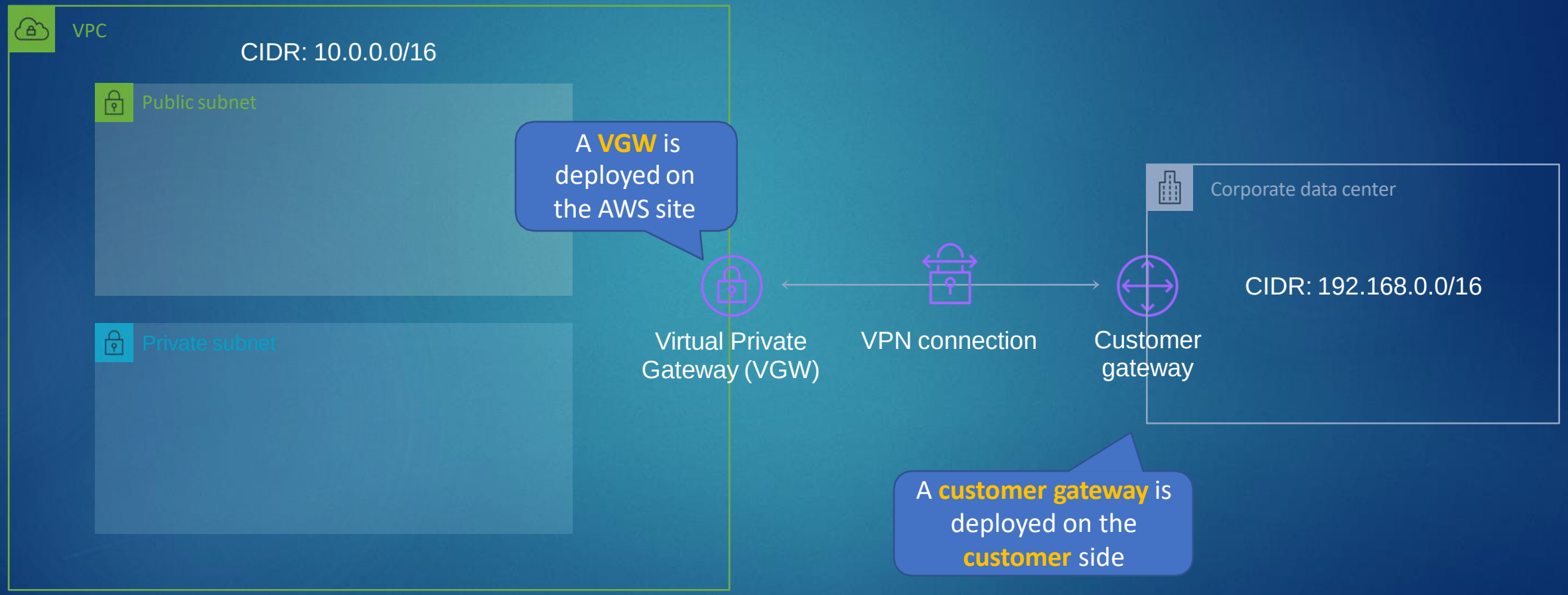
VPC Peering


VPC Peering enables routing using private IPv4 or IPv6 addresses
VPC A
10.1.0.0/16
VPC B
10.2.0.0/16
CIDR blocks cannot overlap
VPCs can be in different accounts and Regions
VPC C
10.3.0.0/16
VPC D
10.4.0.0/16
VPC Peering connections are NOT transitive – full mesh required

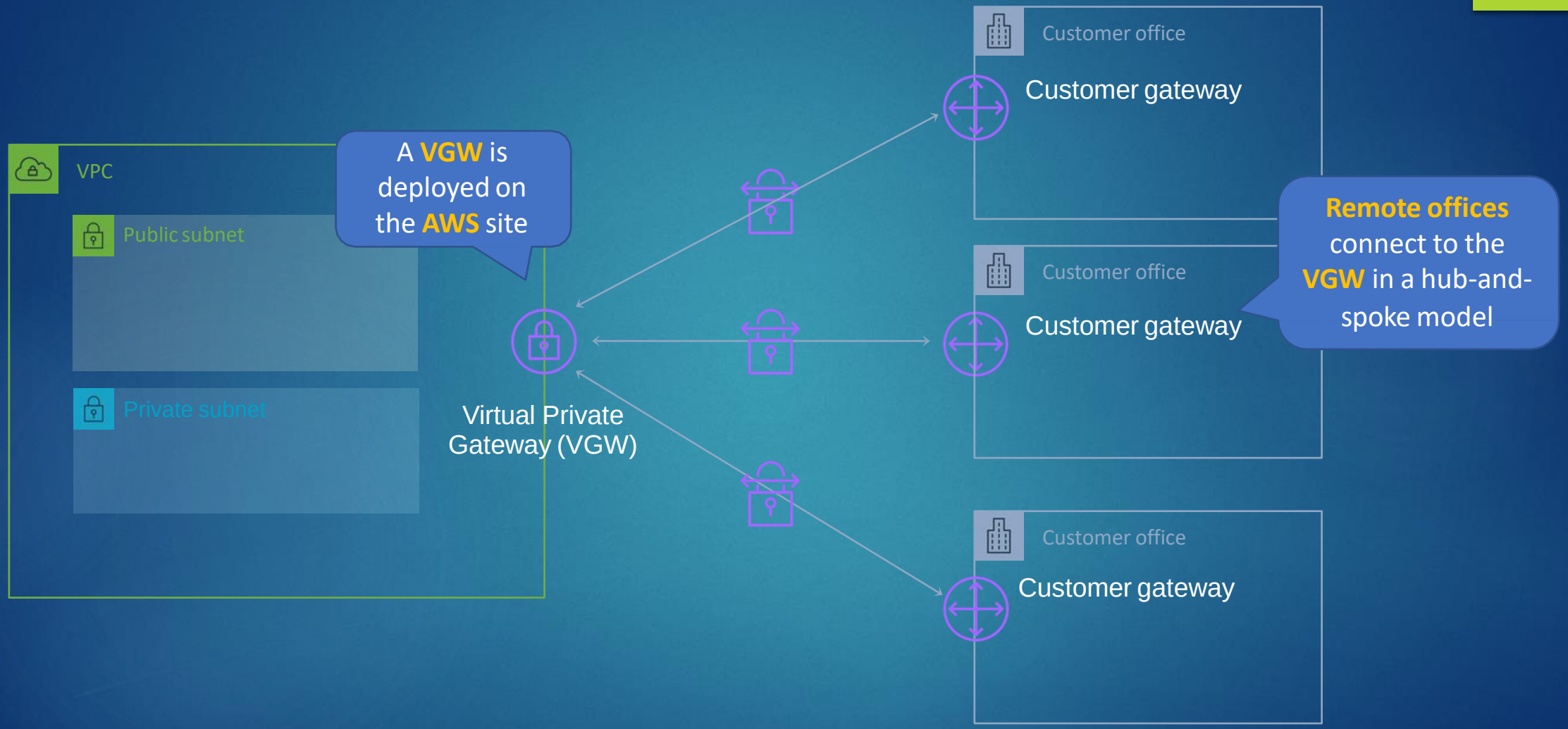
Amazon VPN and AWS Direct Connect

AWS Site-to-Site VPN

AWS VPN is a managed IPSec VPN

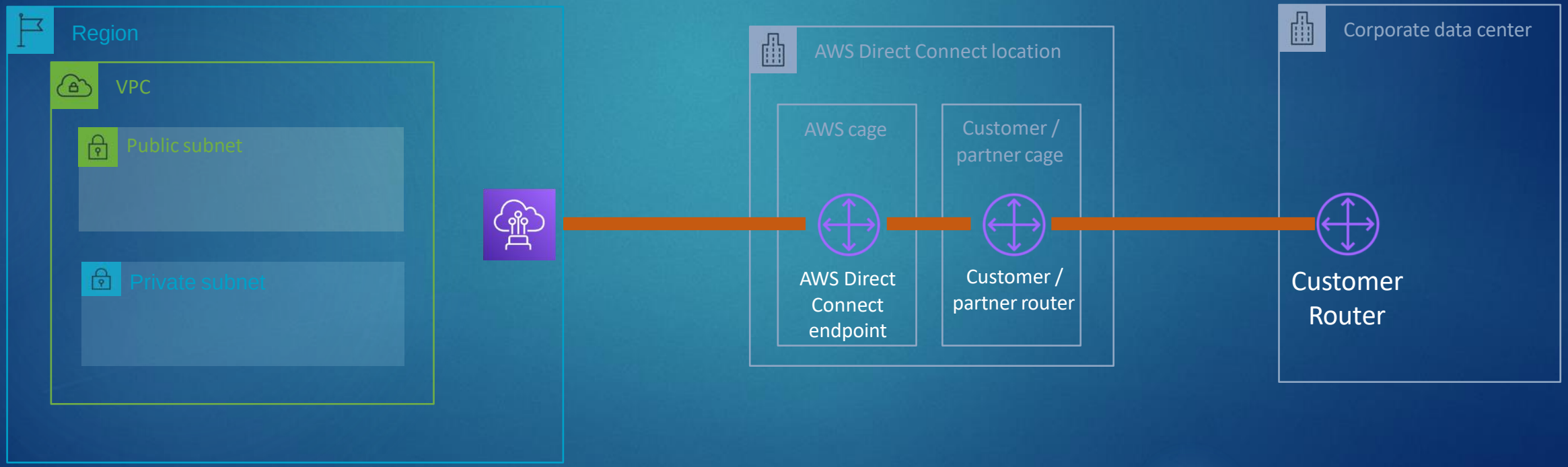


AWS VPN CloudHub



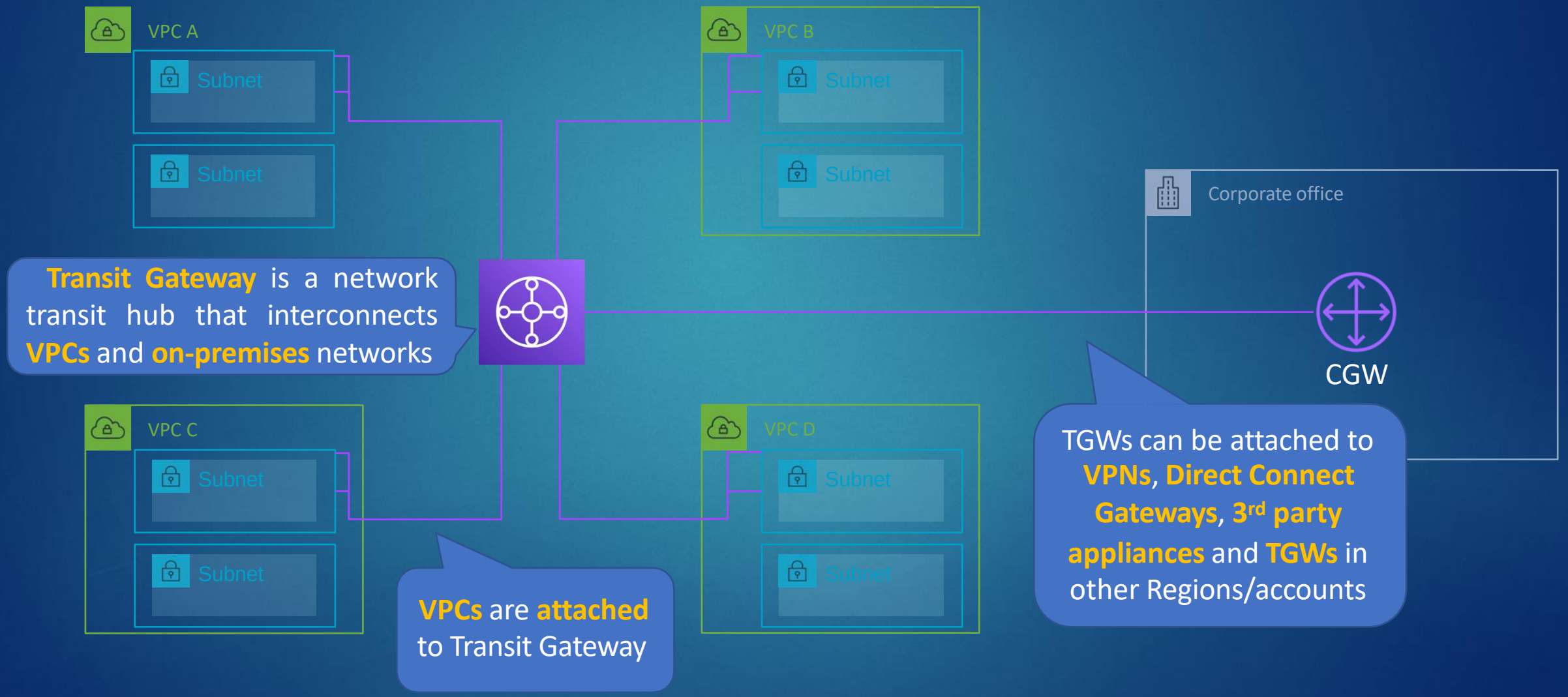
AWS Direct Connect

- **Private** connectivity between AWS and your data center / office
- Consistent network experience – increased **speed/latency** & **bandwidth/throughput**
- Lower costs for organizations that transfer **large** volumes of data



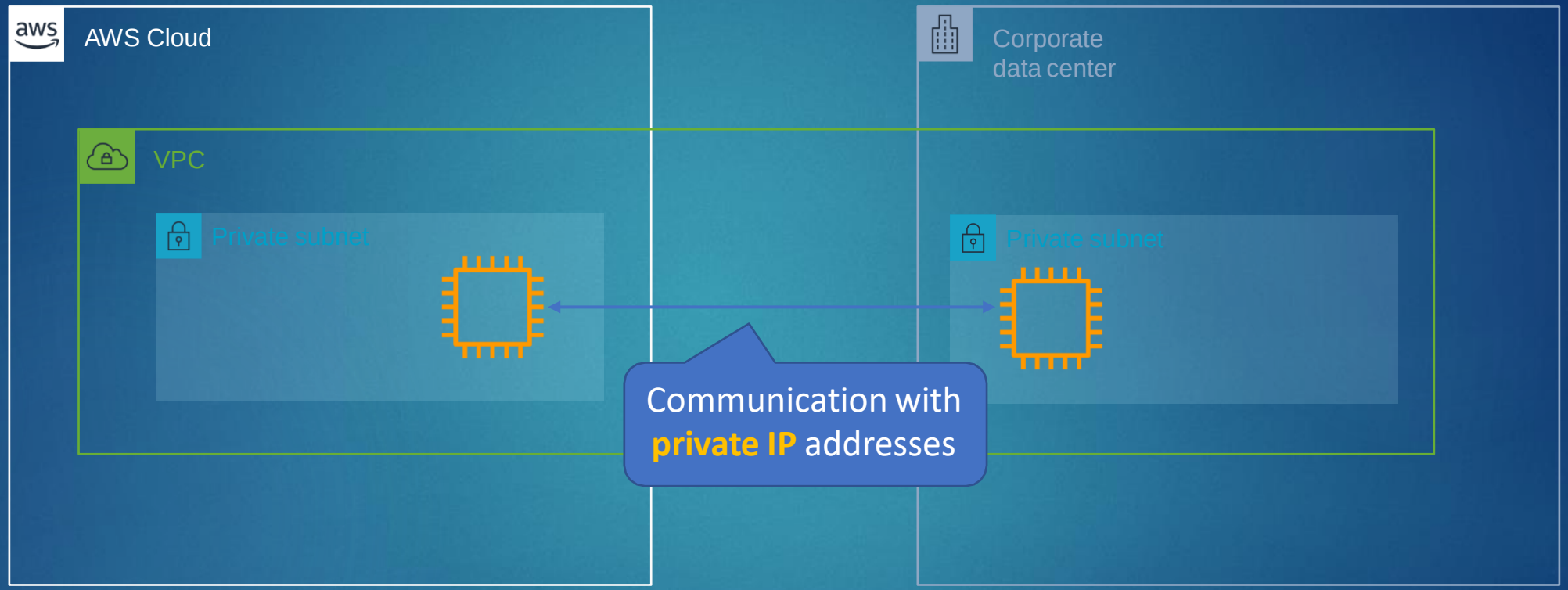
AWS Transit Gateway

AWS Transit Gateway



AWS Outposts

AWS Outposts



AWS Outposts

Services you can run on AWS Outposts include:

- Amazon EC2
- Amazon EBS
- Amazon S3
- Amazon VPC
- Amazon ECS/EKS
- Amazon RDS
- Amazon EMR