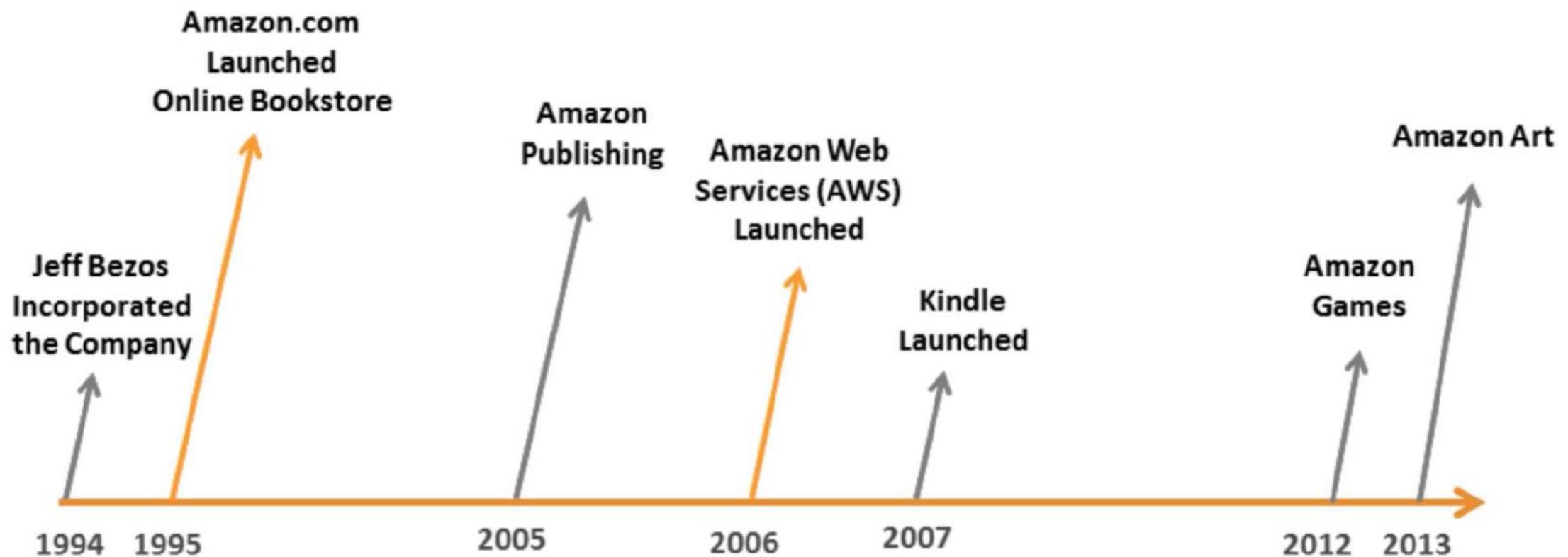


Introduction to AWS

AWS History – Still Being Told



AWS Enterprise Customers



AWS Public Sector Clients: A Sample



AWS Key Cloud Computing Characteristics

On Demand



Uniform



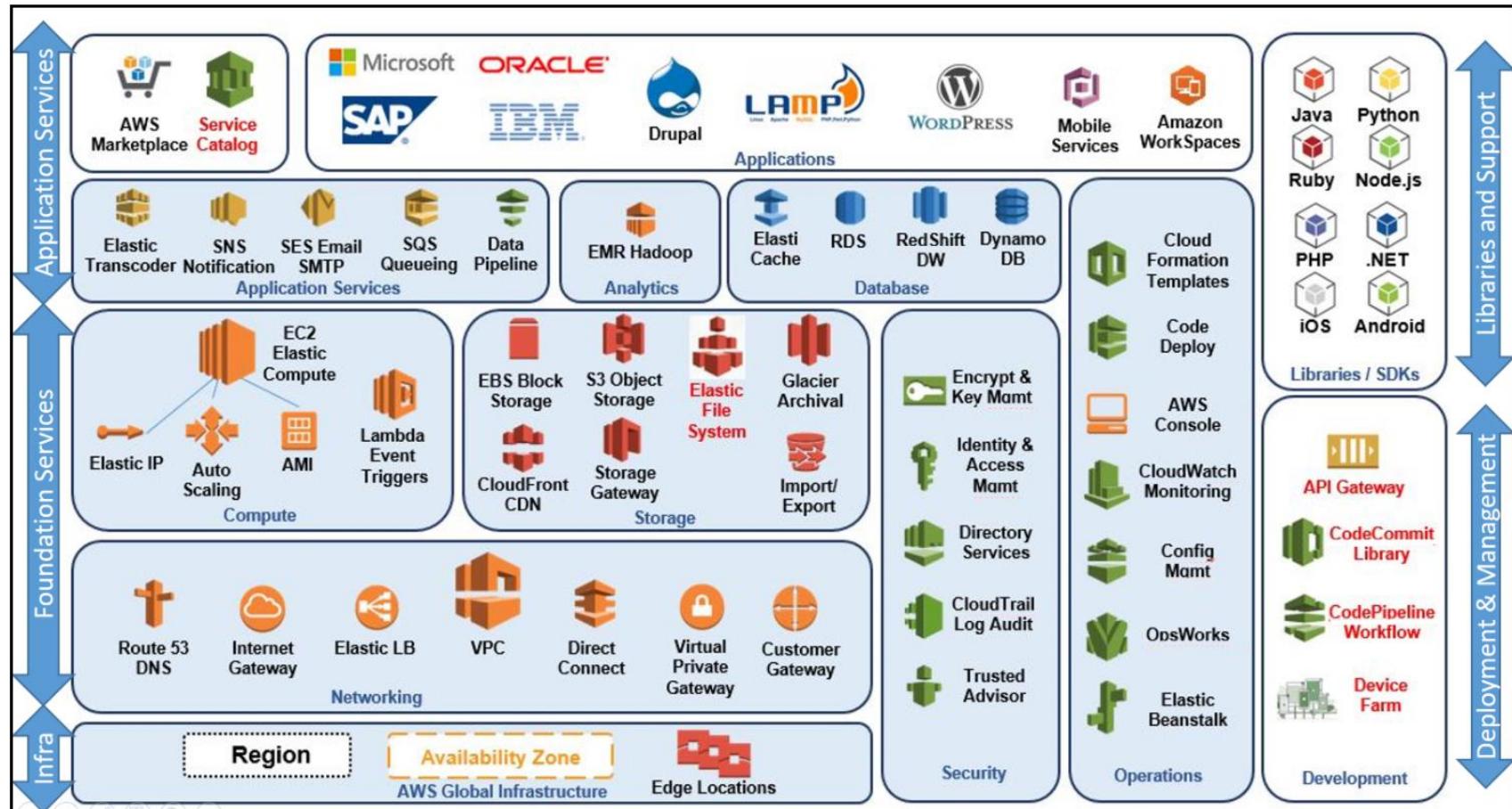
Pay as You Go



Available

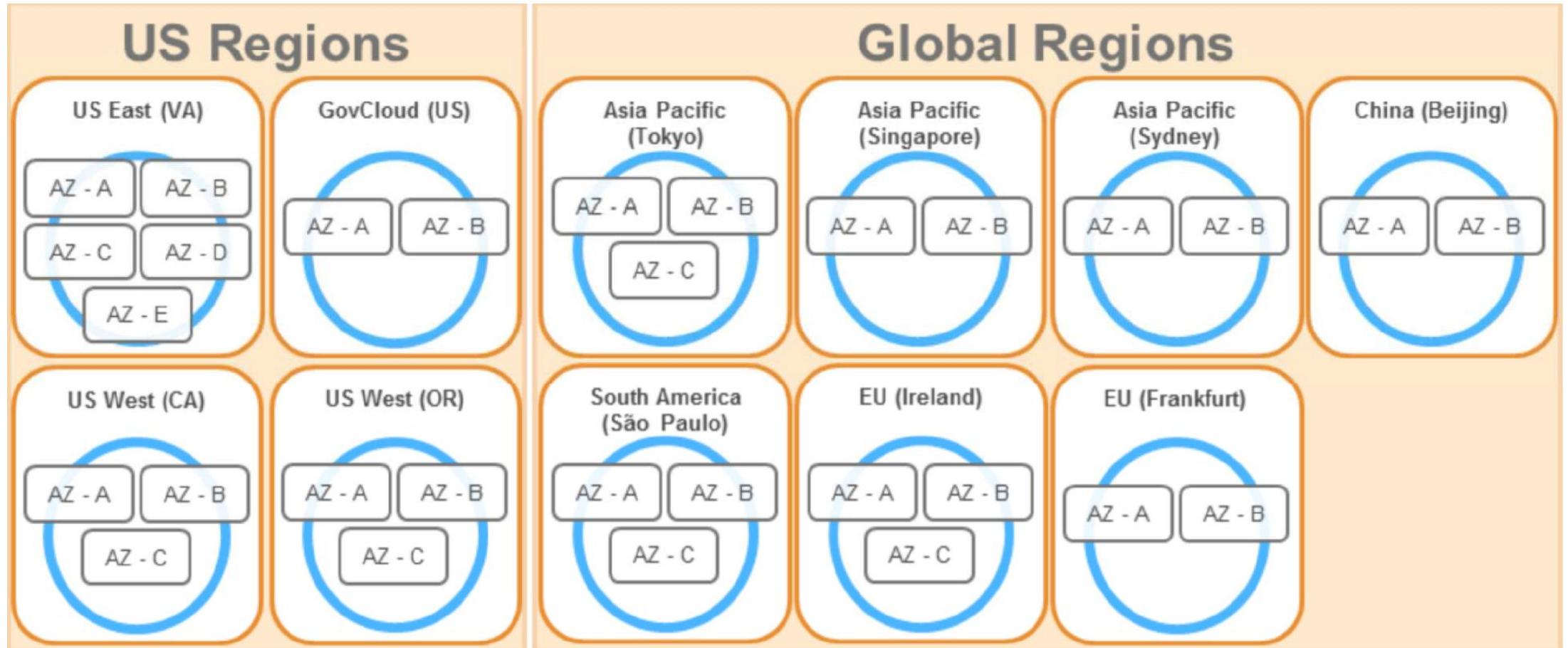


AWS Logical Architecture: Consumer Point of View

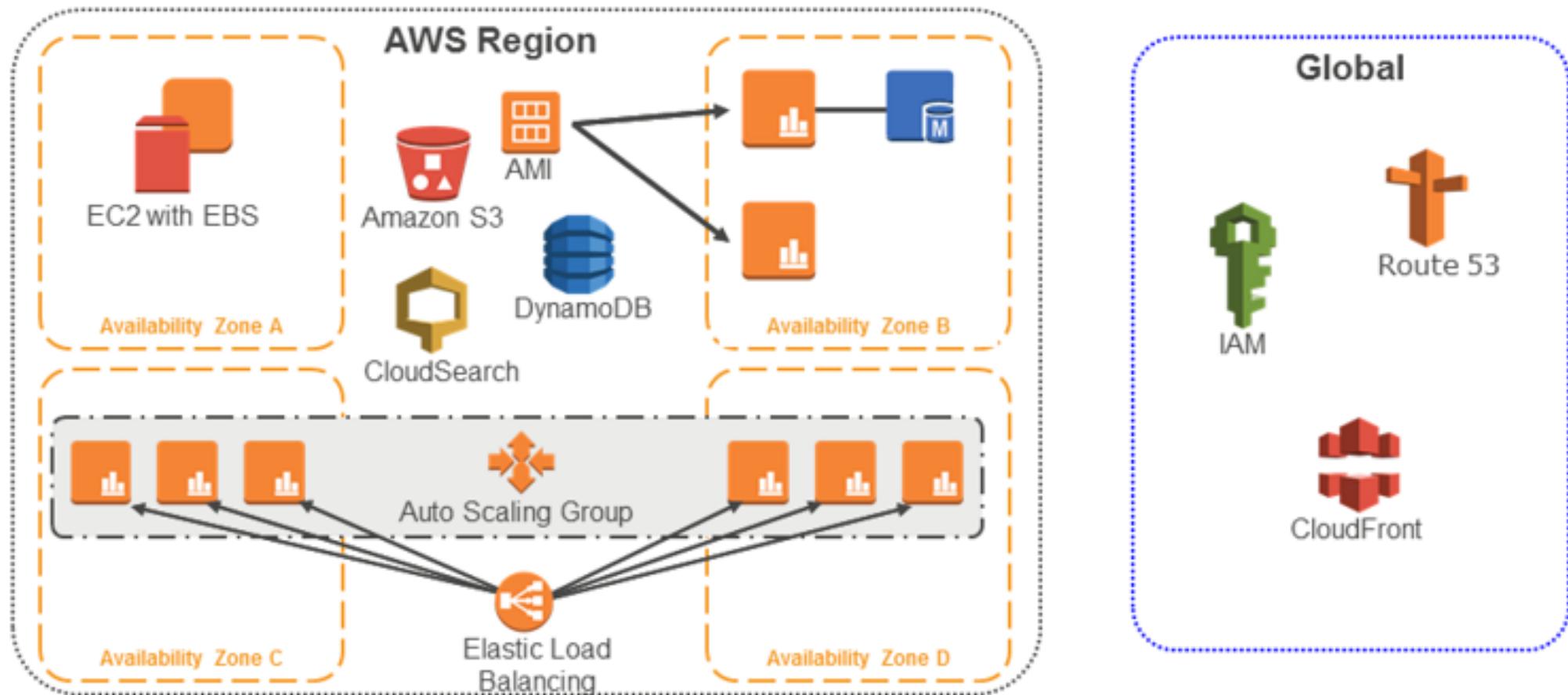


Source: Raja Malleswara Rao Pattamsetti, 2017

AWS Regions and Availability Zones



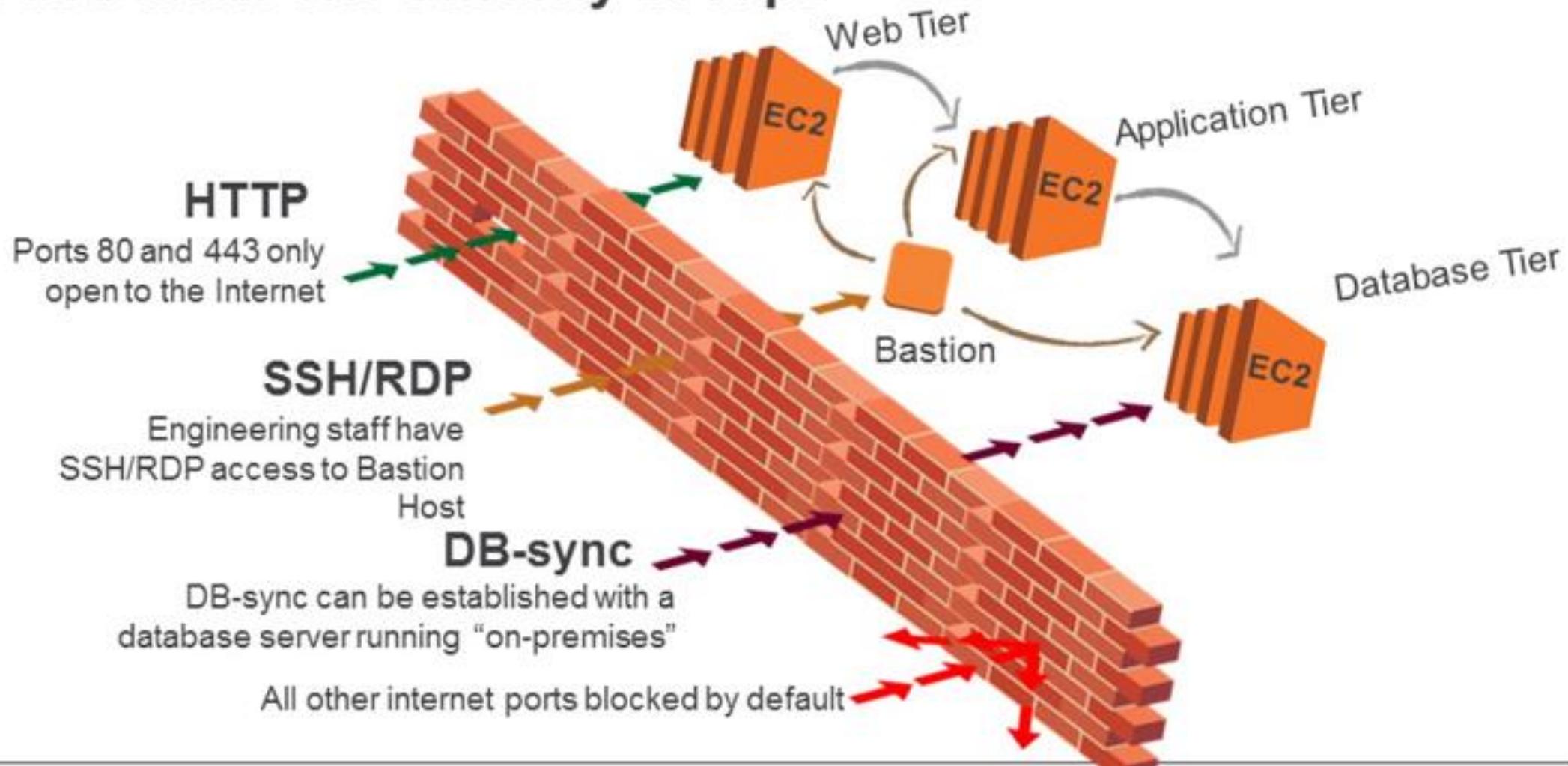
Regions Services vs AZ's Services



AWS Security Capabilities

SSL Endpoints	Security Groups	IAM	VPC
<p>Secure Transmission</p> <p>Establish secure communication sessions (HTTPS) using SSL</p>	<p>Instance Firewalls</p> <p>Configure firewall rules for instances using Security Groups</p>	<p>User Accounts</p> <p>Create individual IAM accounts so that each of your users has their own AWS security credentials</p>	<p>Subnet Control</p> <p>In your Virtual Private Cloud, create low-level networking constraints for resource access, such as public and private subnets, internet gateways, and NATs</p>

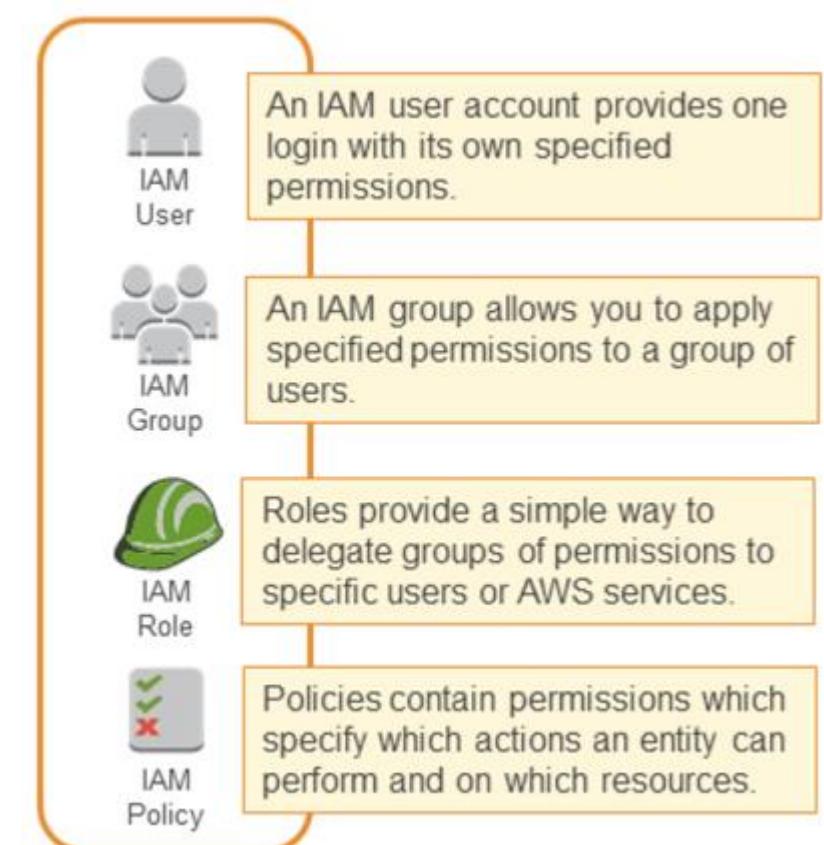
AWS Multi-Tier Security Groups



IAM



The screenshot shows the AWS Identity and Access Management (IAM) homepage. At the top, there's a teal header with the title "AWS Identity and Access Management (IAM)" and a lock icon. Below the header, a main section says "Securely control access to AWS services and resources for your users" with a "Get Started with AWS IAM" button. A navigation bar follows with links for "IAM Home", "Product Details", "Getting Started", "Resources & Tools", "FAQs", and "Partners". The main content area contains a paragraph about IAM enabling secure access to AWS services, followed by sections on IAM being a feature of the AWS account and getting started with IAM best practices.



AWS Console Demo

How to Use IAM

In Our Scenario

The diagram illustrates a scenario within an AWS Account. On the left, a user icon labeled 'Adele' is connected by a line to a central box labeled 'AWS Account'. Inside this box are three service icons: 'Amazon S3' (red block), 'AWS IAM' (play button inside a circle), and 'Amazon EC2' (orange server blocks). Below the box is the text 'Account Services and Resources'. To the right of the box is another user icon labeled 'David'. In the bottom right corner of the main diagram area is a timer displaying '6:44'.

Getting Started with AWS IAM

Homework IAM

- Create 2 users, S3TestUser and EC2TestUser, with strong passwords
- Generate an access key to each user and save them
- Attach AmazonS3FullAccess to S3TestUser
- Create a user group, EC2TestGroup
- Attach AmazonEC2FullAccess to this group
- Add EC2TestUser to this group
- Create a role name, EC2Describe, and attach AmazonEC2ReadOnlyAccess to this role
- Test the users and policies by
 - Login as S3TestUser and try to manipulate EC2 services
 - Then, create a bucket (<classname><yourlastname><youfirstname>) in the east region, and upload simple file into that bucket
 - Finally, login as EC2TestUser and try to access the file you uploaded before

What is a Virtual Private Cloud?

- Your own logically isolated section of the Amazon Web Services (AWS) Cloud
- By default, your VPC has no access to the internet nor are instances addressable from the internet
- You have complete control over your virtual networking environment
- Proven and well-understood networking concepts:
 - User defined IP address range
 - Subnets
 - Route Tables
 - Access Control Lists
 - Network Gateways
- A way to gain agility as well as additional security



What's in the VPC tool box?



VPC - User-defined address space up to /16 (65,536 addresses)



Subnets - 200 user-defined subnets up to /16



Route Tables – Define how traffic should be routed from/to each subnet



Access Control Lists – Stateless network filtering between subnets



Internet Gateway – A **logical** device enabling traffic to be routed to/from the public internet



Managed NAT – Provide Network Address Translation to private instances for 10Gbps traffic

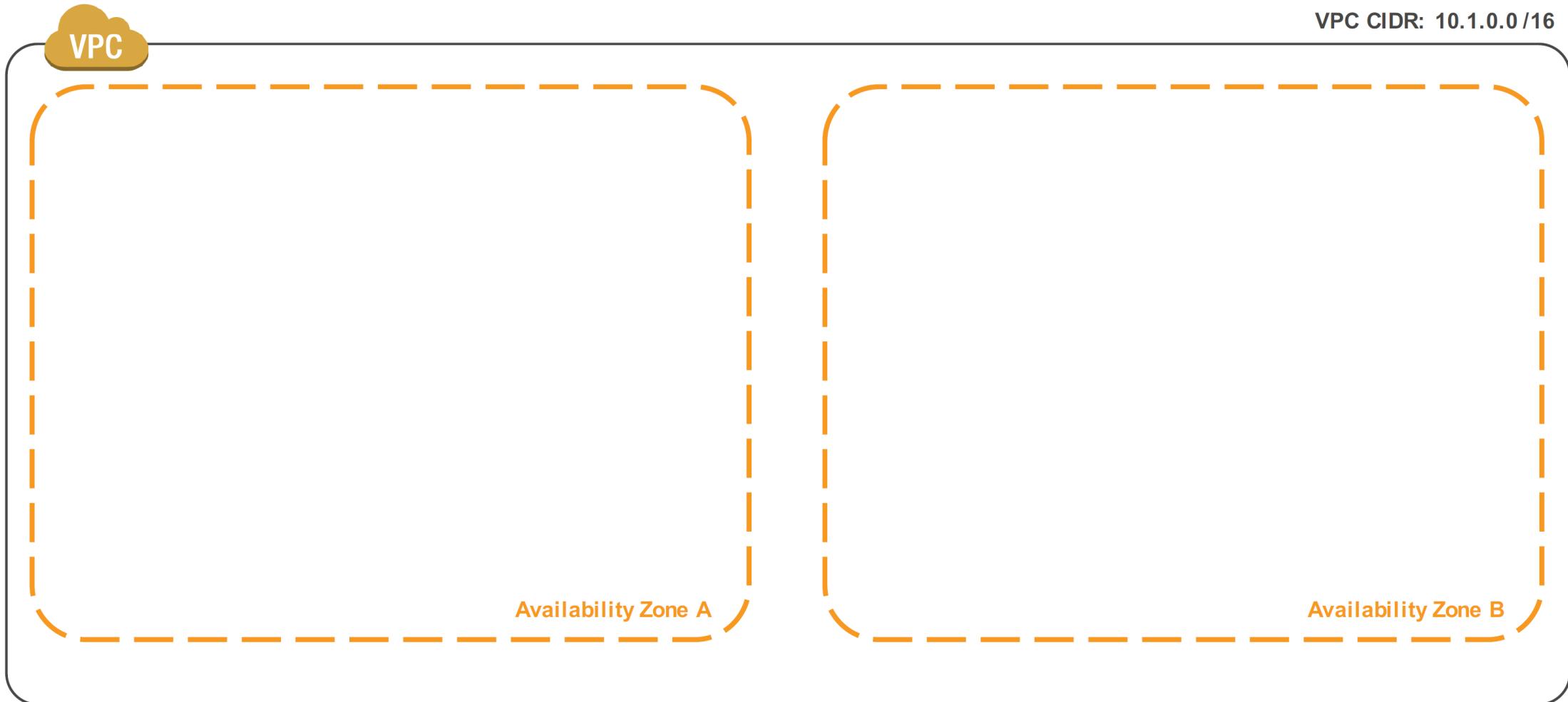


Virtual Private Gateway - The Amazon end of a VPN connection

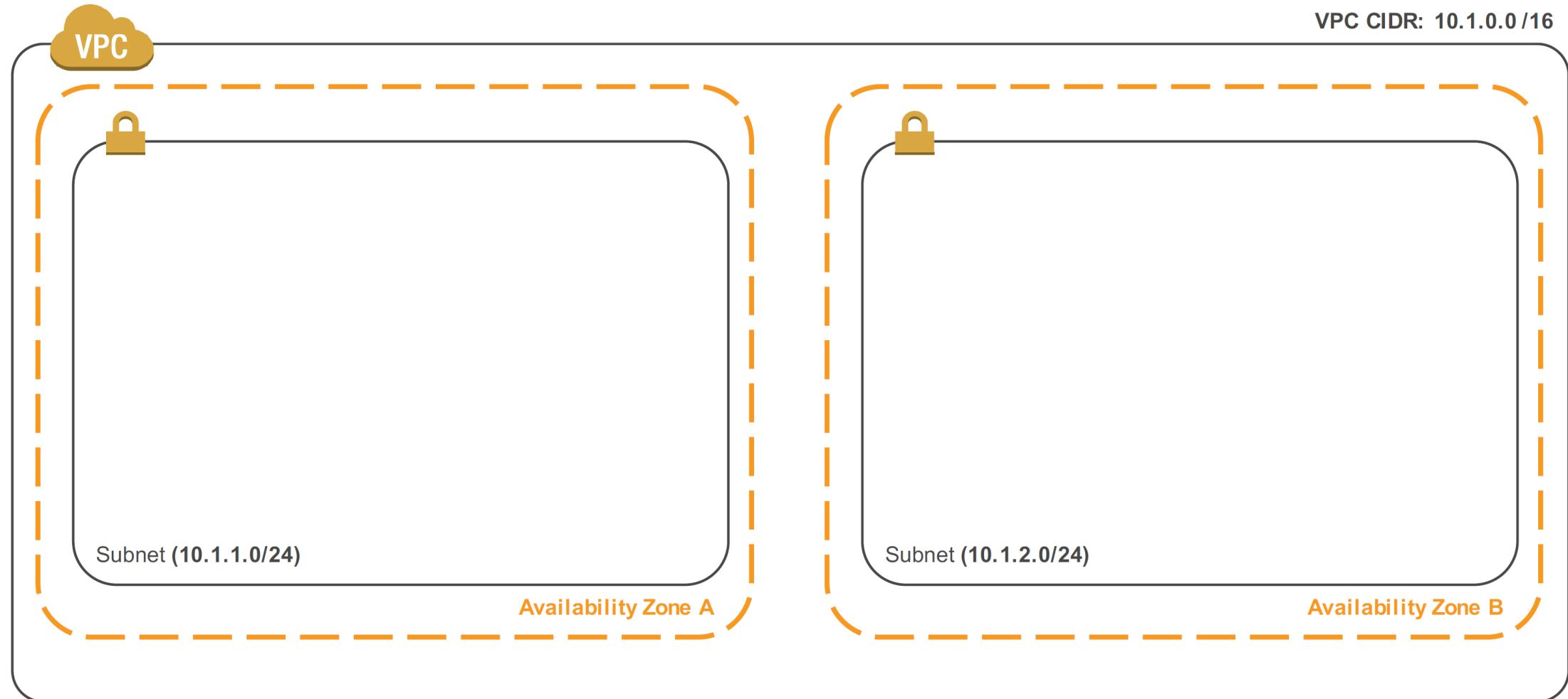


Customer Gateway - The router at the customer end of a VPN connection

VPCs span an entire region



Subnets sit in a single VPC in a single AZ

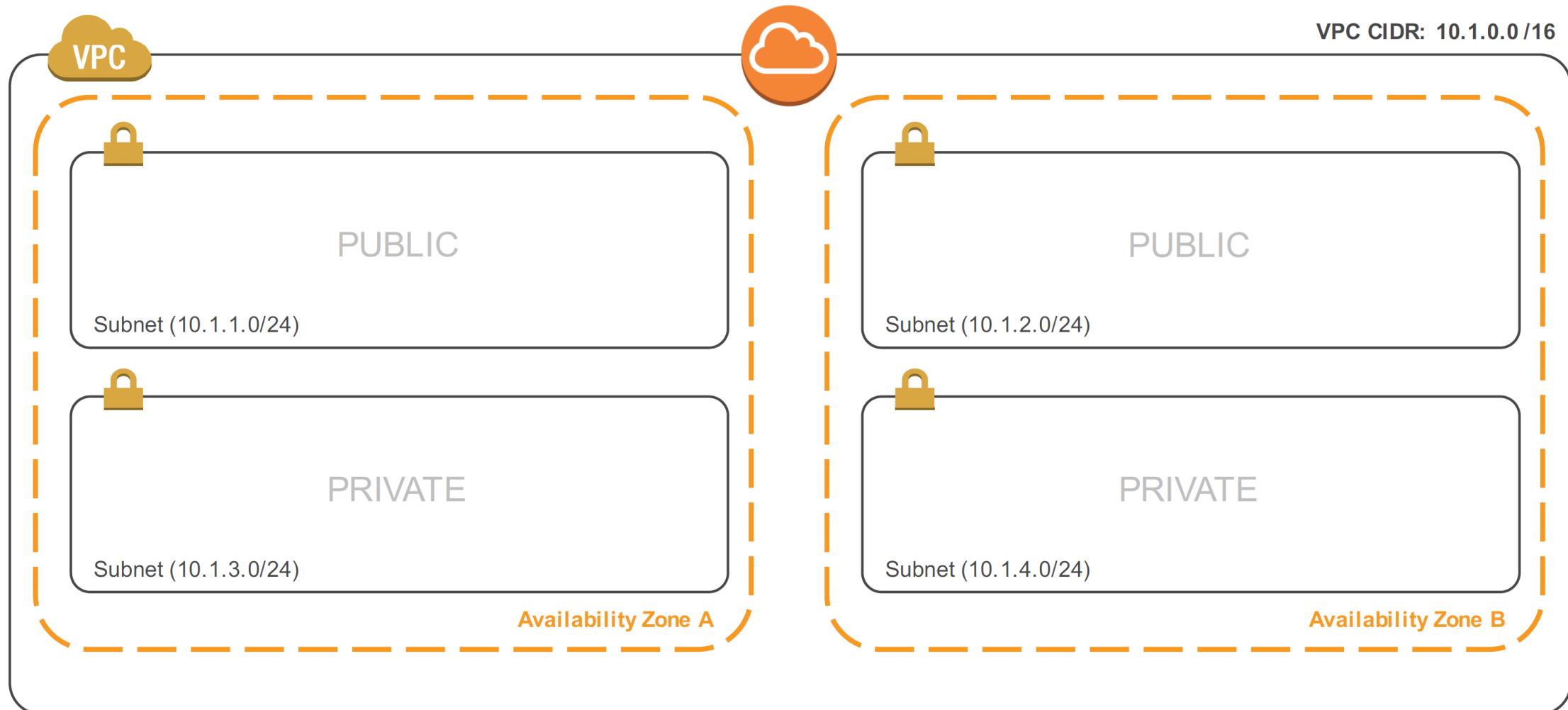


Plan your VPC IP space before creating it

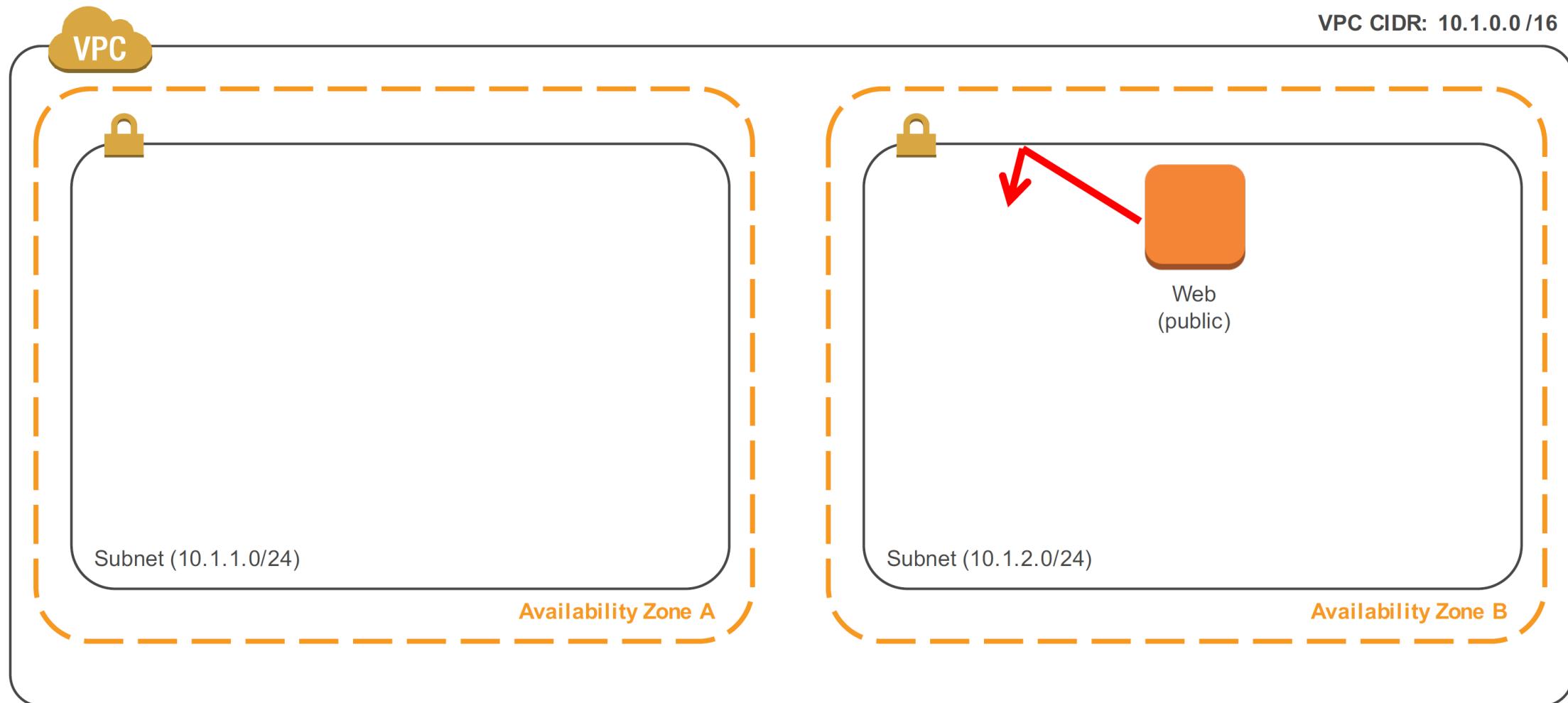


- Consider future AWS region expansion
- Consider future connectivity to your internal networks
- Consider subnet design
- VPC can be /16 down to /28
- CIDR cannot be modified after creation

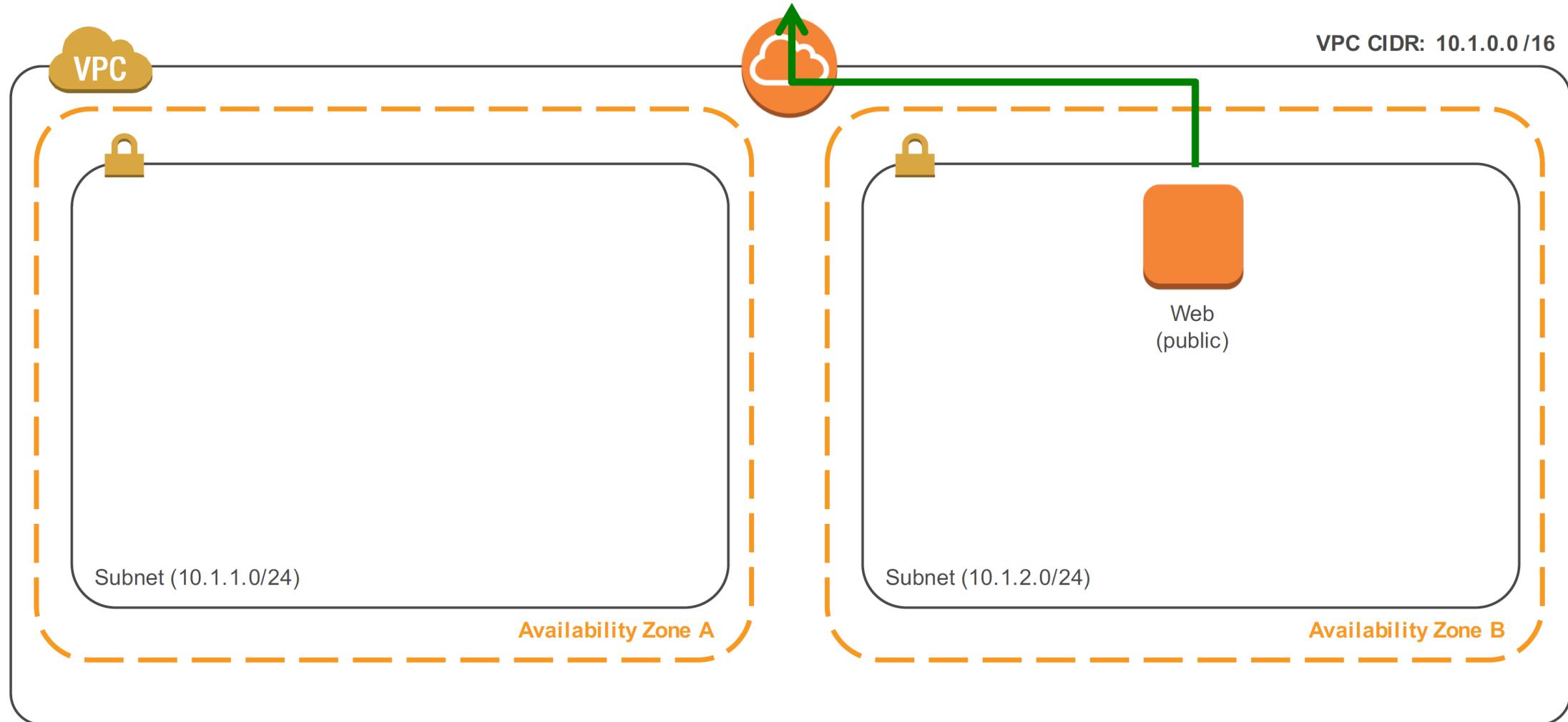
Public / Private Subnets



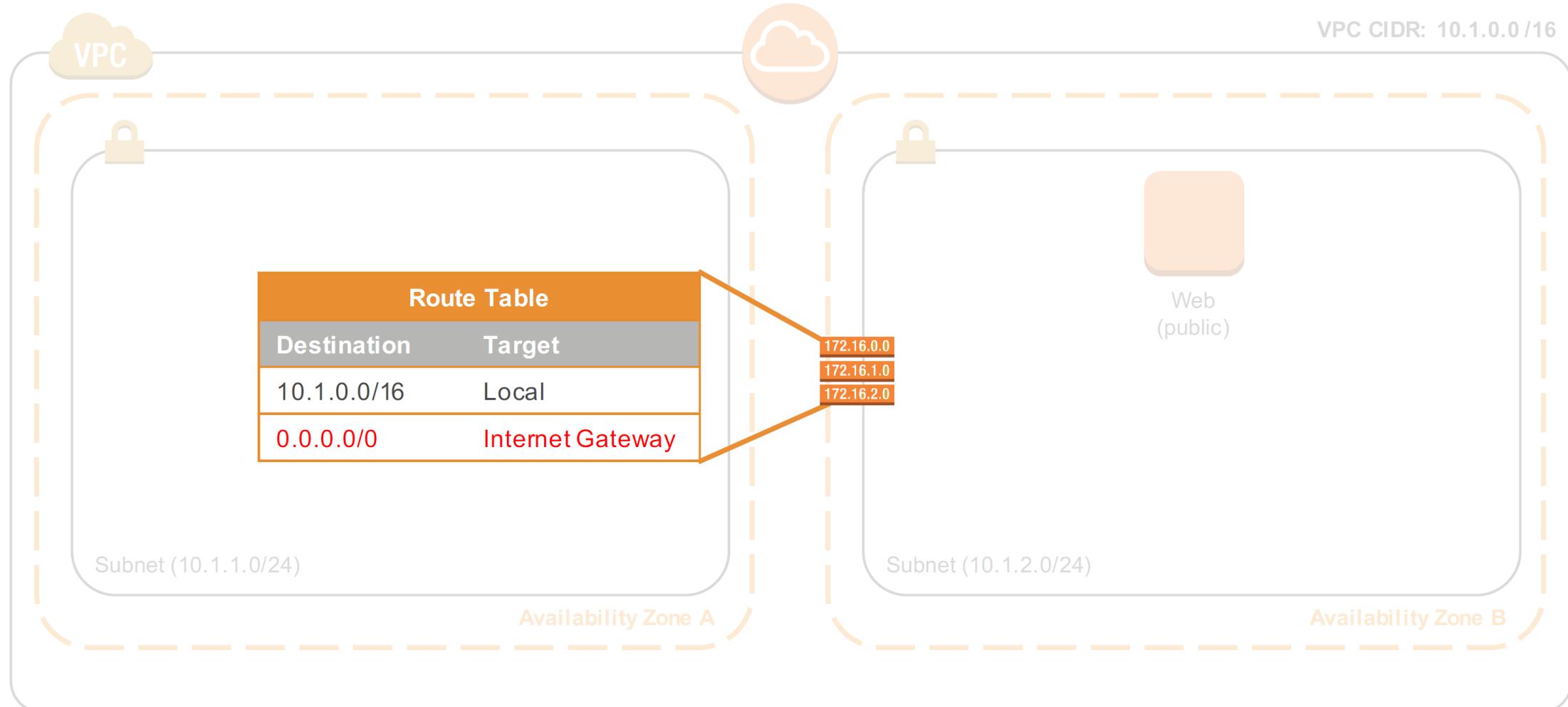
Public Subnet Routing



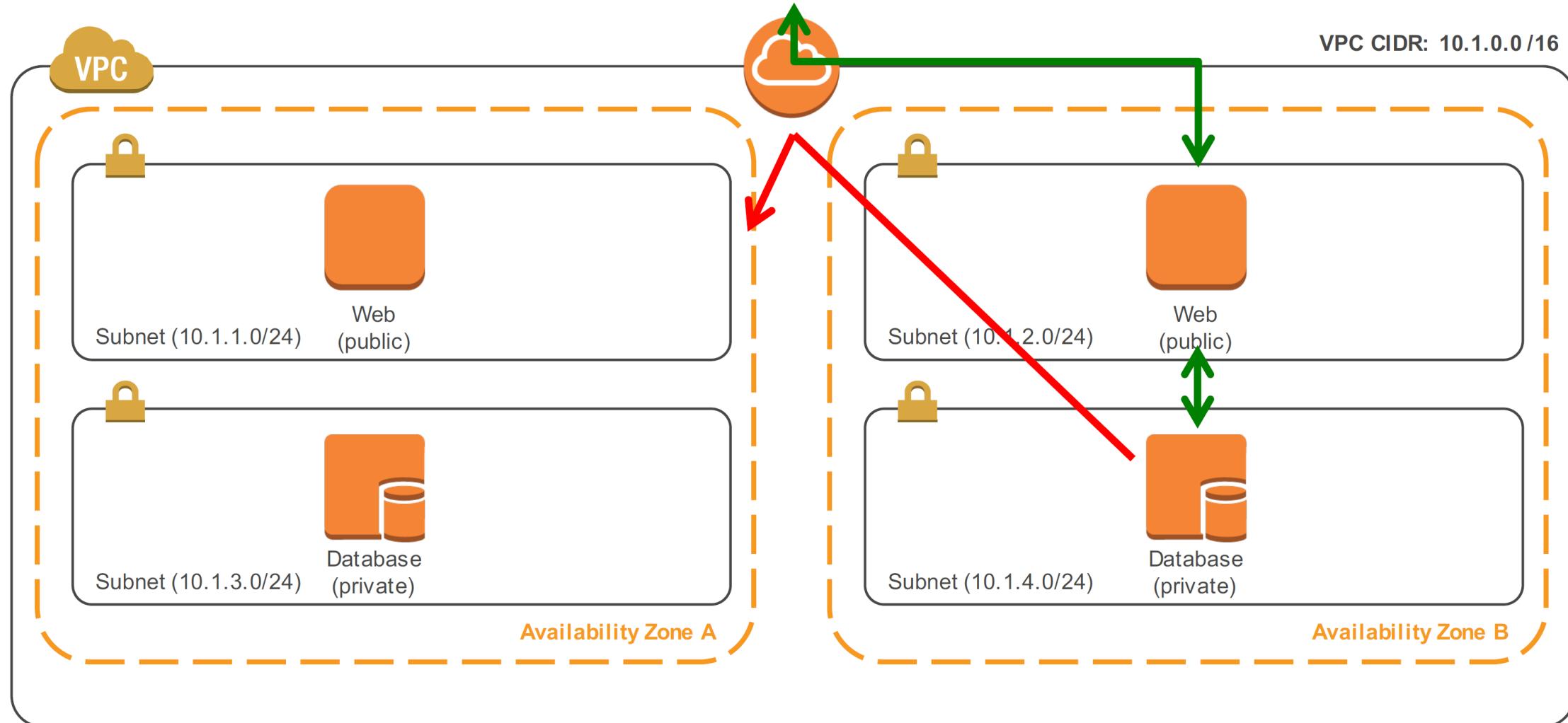
Public Subnet Routing – Internet Gateway



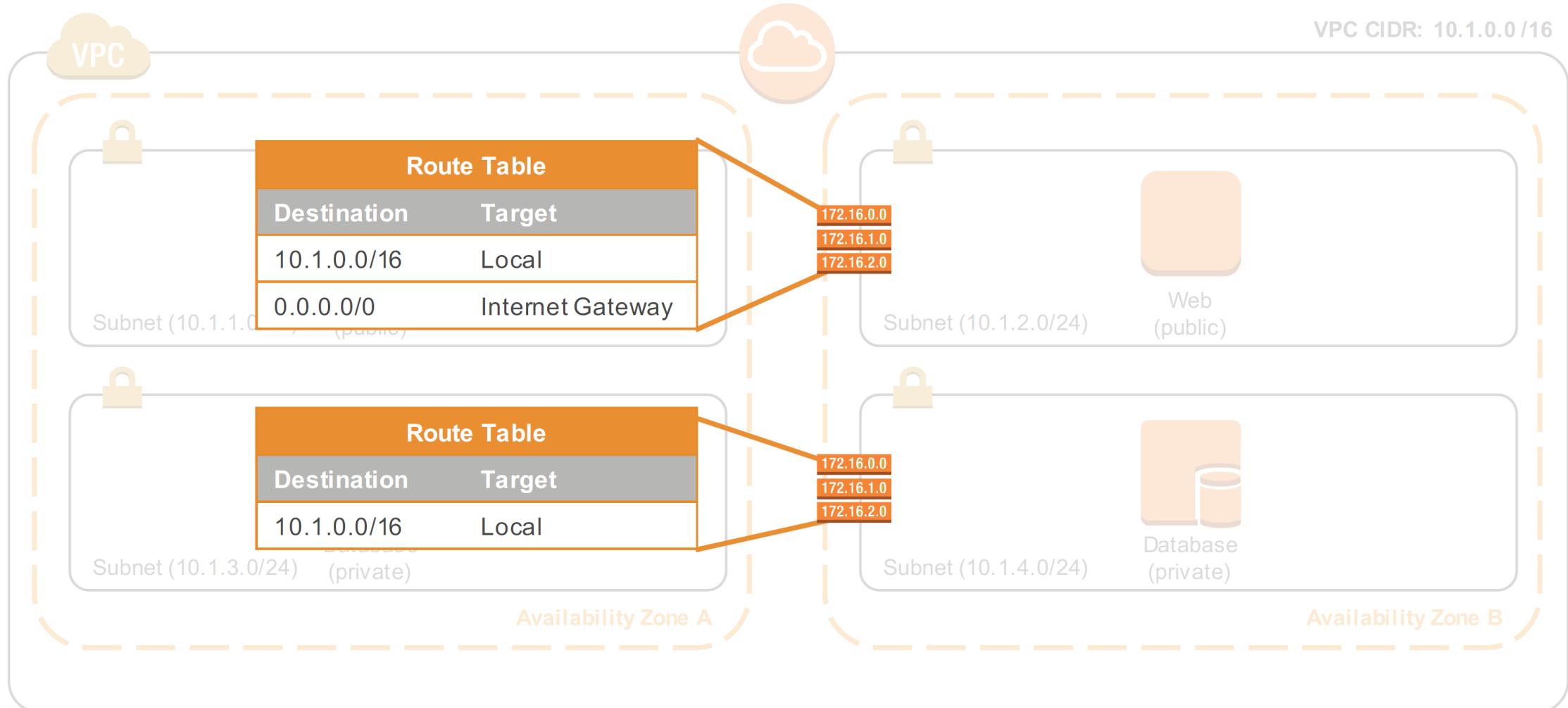
Public Subnet Routing – Internet Gateway



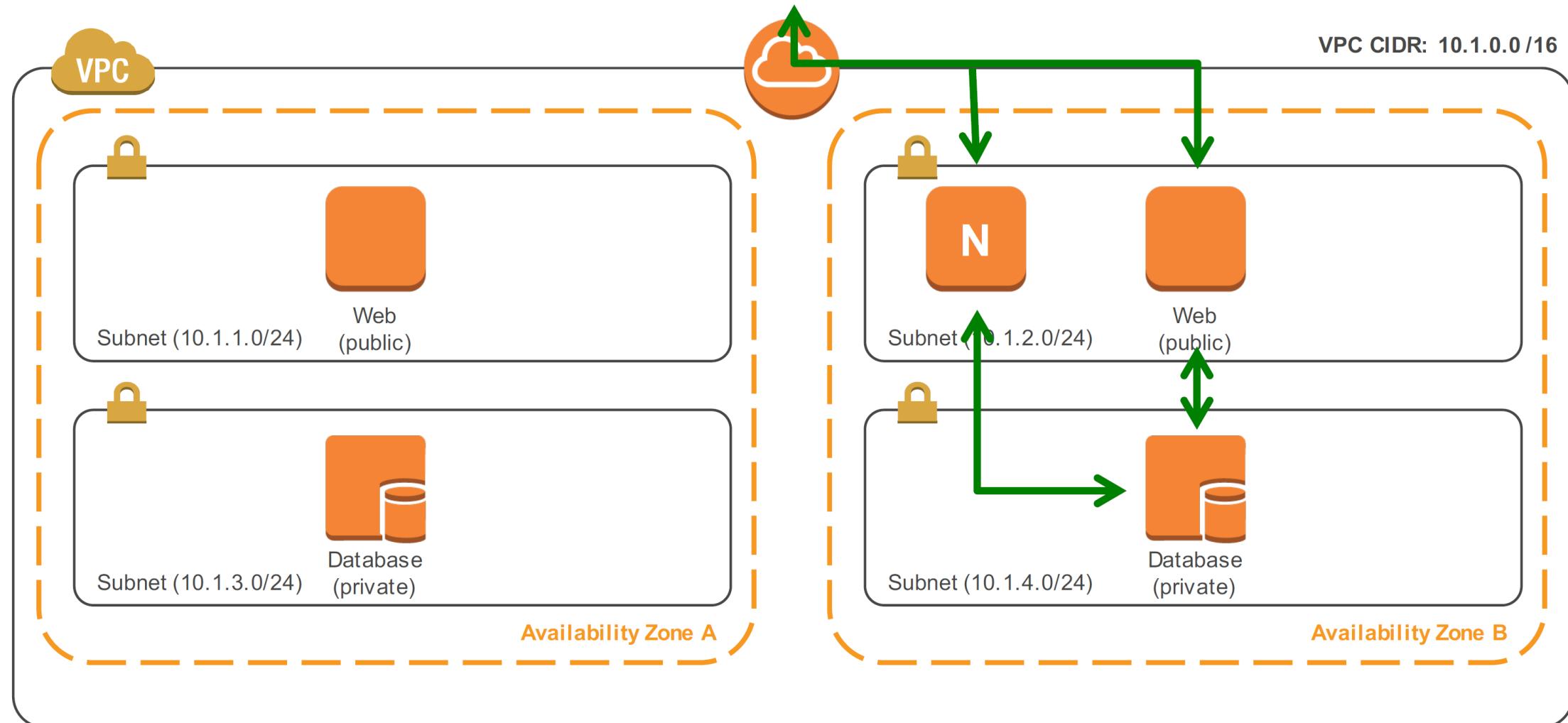
Private Subnet Routing



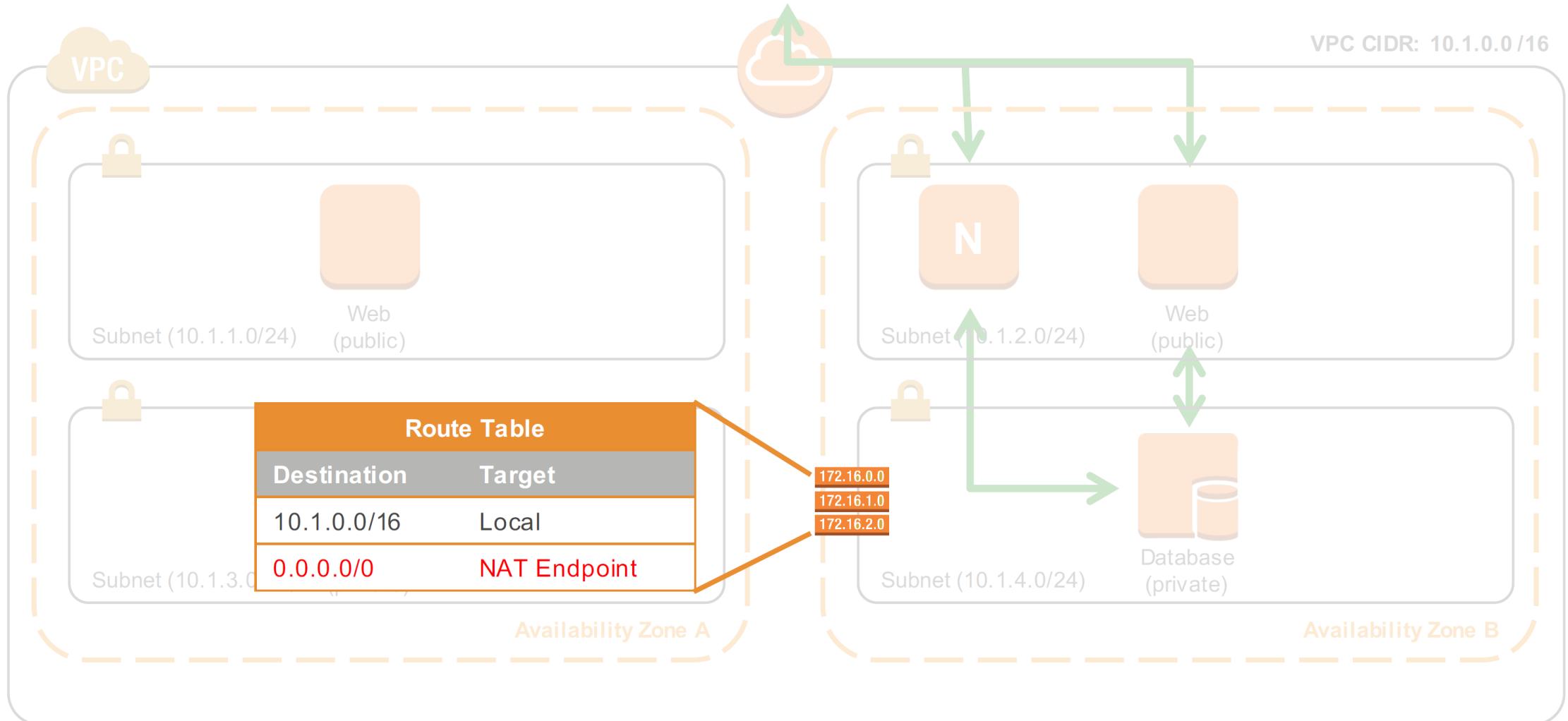
Private Subnet Routing



Private Subnet Routing – NAT Gateway



Private Subnet Routing - NATGateway



Network ACLs = Stateless Firewall Rules

Can be applied on a subnet basis

The screenshot shows the AWS Network ACLs interface. At the top, there's a search bar with placeholder text "Search Network ACLs and the X". Below it is a table with columns: Name, Network ACL ID, Associated VPC, Subnets, and Status. One row is selected, showing "acl-5cc5b539", "vpc-327d1857 (172.31.0)", "3 Subnets", and "Yes". A yellow arrow points from the "Can be applied on a subnet basis" text to this row. To the right of the table, another yellow box contains the text "English translation: Allow all traffic in". Below the table, under the heading "acl-5cc5b539", is a detailed view of the rules. It has a table with columns: Rule #, Type, Protocol, Port Range, Source, and Allow / Deny. Two rows are listed: Rule #100 with Type "ALL Traffic", Protocol "ALL", Port Range "ALL", Source "0.0.0.0/0", and Allow/Deny "ALLOW"; and a wildcard rule "*" with the same parameters, Allow/Deny "DENY". This table is also highlighted with a black border.

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

Security Groups = Stateful Firewall Rules

Create Security Group Delete Security Group

Filter VPC security groups X « < 1 to 3

<input type="checkbox"/>	Name tag	Group ID	Group Name	VPC	Description
<input checked="" type="checkbox"/>	MyWebServers	sg-82ba7ee6	MyWebServers	vpc-327d1857	Allows all traffic from the Internet
<input type="checkbox"/>	MyBackends	sg-8fba7eeb	MyBackends	vpc-327d1857	Allows only traffic from MyWebServers
<input type="checkbox"/>		sg-07996163	default		

In English: Hosts in this group are reachable from the Internet on port 80 (HTTP)

Type	Protocol	Port Range	Source
HTTP (80)	TCP (6)	80	0.0.0.0/0

sg-82ba7ee6 | MyWebServers

Summary Edit

Type Protocol

HTTP (80) TCP (6) 80 0.0.0.0/0

Security Group Mutual Trust

Create Security Group Delete Security Group

Filter VPC security groups « < 1 to 3 of »

<input type="checkbox"/>	Name tag	Group ID	Group Name	VPC	Description
<input type="checkbox"/>	MyWebServers	sg-82ba7ee6	MyWebServers	vpc-327d1857	Allows all traffic from the Internet
<input checked="" type="checkbox"/>	MyBackends	sg-8fba7eeb	MyBackends		
<input type="checkbox"/>		sg-07996163	default		

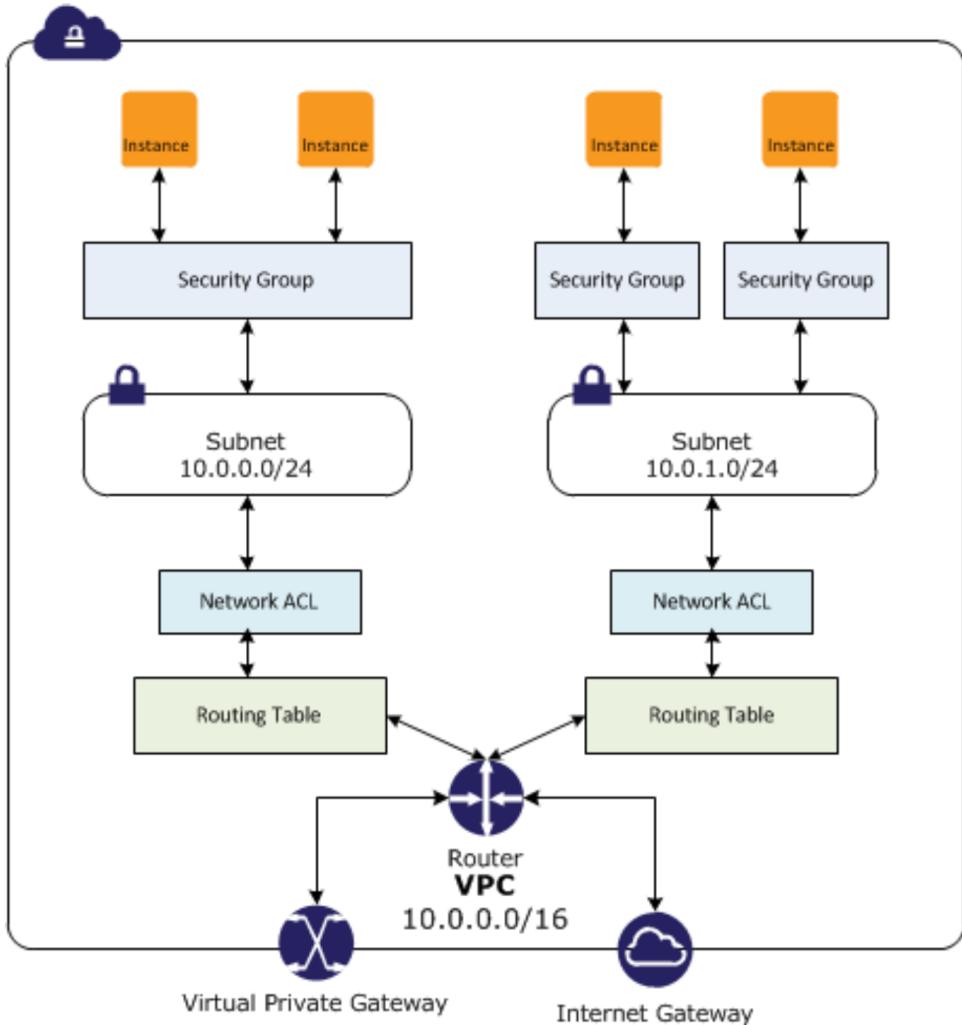
In English: Only instances in the MyWebServers Security Group can reach instances in this Security Group

sg-8fba7eeb | MyBackends

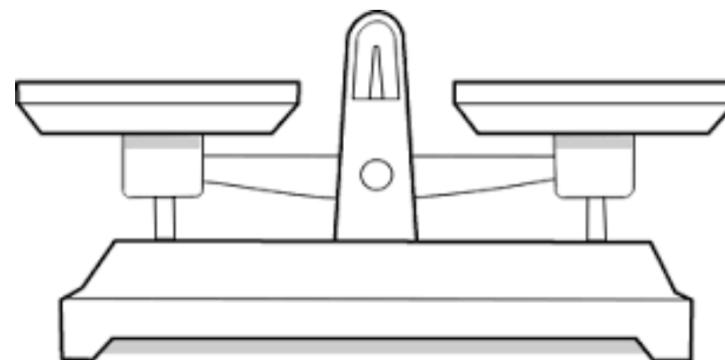
Summary	Type	Protocol	Port Range	Source
Edit	Custom TCP Rule	TCP (6)	2345	sg-82ba7ee6

Custom TCP Rule TCP (6) 2345 sg-82ba7ee6

Security Balancing Act

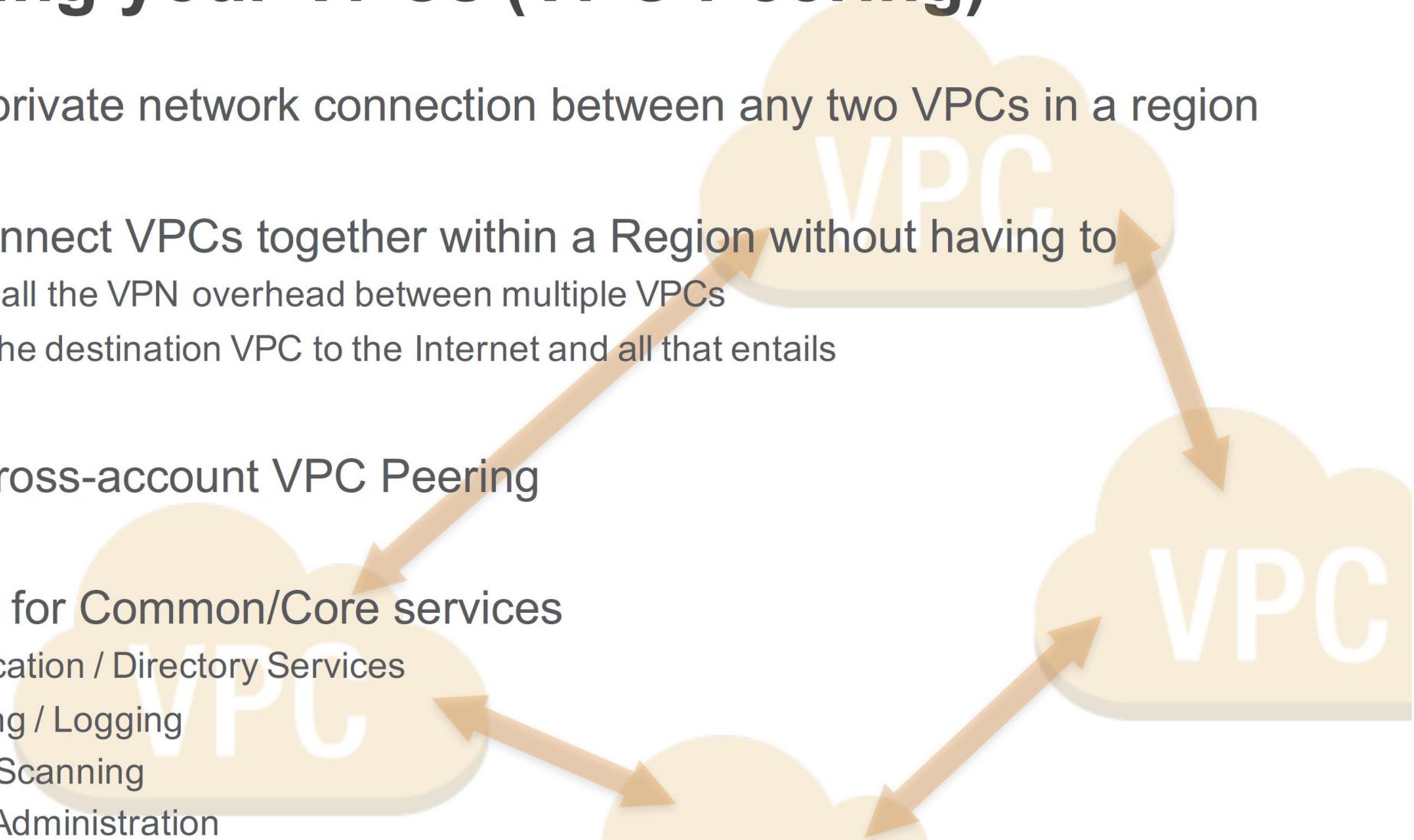


Comparison between Security Groups and Network ACLs		
Area of Security	Security Group	Network ACL
Operational Level	Instance level	Subnet level
Supports ALLOW rules...	...only	...and DENY rules
State Type	Stateful	Stateless
Evaluation method	All rules evaluated	Stop on first match
Applicability to Instances	Only if SG explicitly added to instance	Automatically to all instances in subnet
Source / Destination	IP CIDR and other Security Groups	IP CIDR only



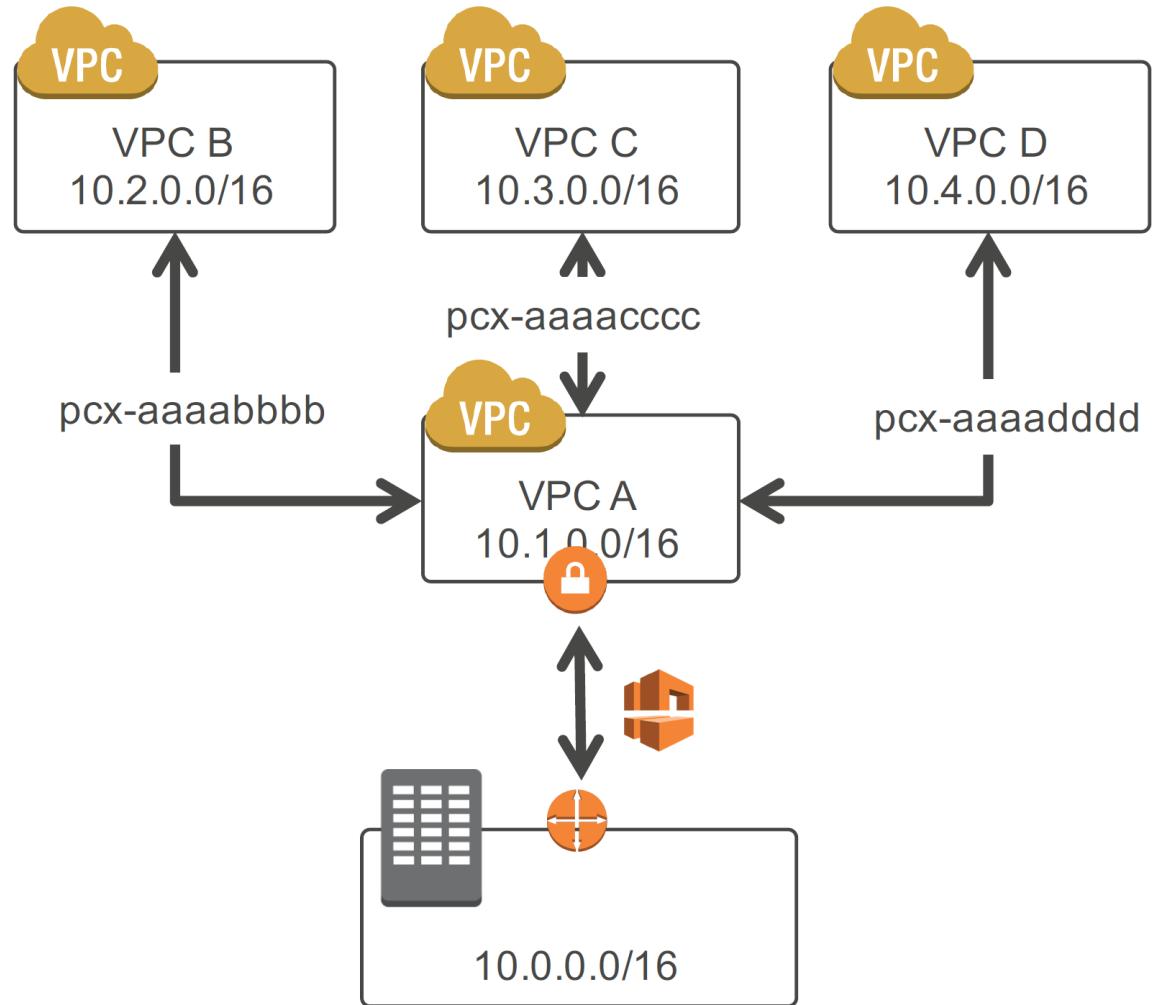
Connecting your VPCs (VPC Peering)

- Creates a private network connection between any two VPCs in a region
- You can connect VPCs together within a Region without having to
 - Maintain all the VPN overhead between multiple VPCs
 - Expose the destination VPC to the Internet and all that entails
- Including cross-account VPC Peering
- Often used for Common/Core services
 - Authentication / Directory Services
 - Monitoring / Logging
 - Security Scanning
 - Remote Administration



Common Design – Shared Services VPC

- Move shared services such as Active Directory, Logging, Monitoring and Service Buses to a shared services VPC (A)
- None of the other VPCs can send traffic directly to each other – they must go through VPC A (= app isolation)
- Only VPC A has direct network access to your data center via Direct Connect
- Routing Tables define which subnets are allowed to route over a peer connection
- Security Groups and NACLs still apply, and Security Groups in VPC A can be defined to mutually trust the Security Groups in the other VPCs



VPC and an Enterprise Data Center

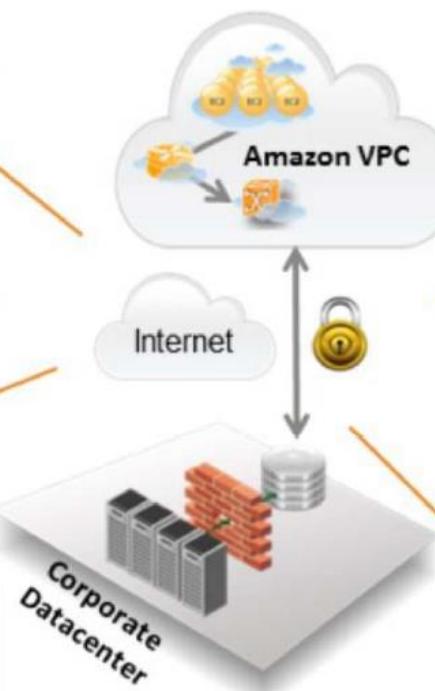
Amazon VPC



Create an Amazon VPC and specify its **private IP address range** from any range you choose.

Divide your VPC's private IP address range into multiple **subnets**.

Bridge your Amazon VPC to your own IT infrastructure via an encrypted **VPN connection**.

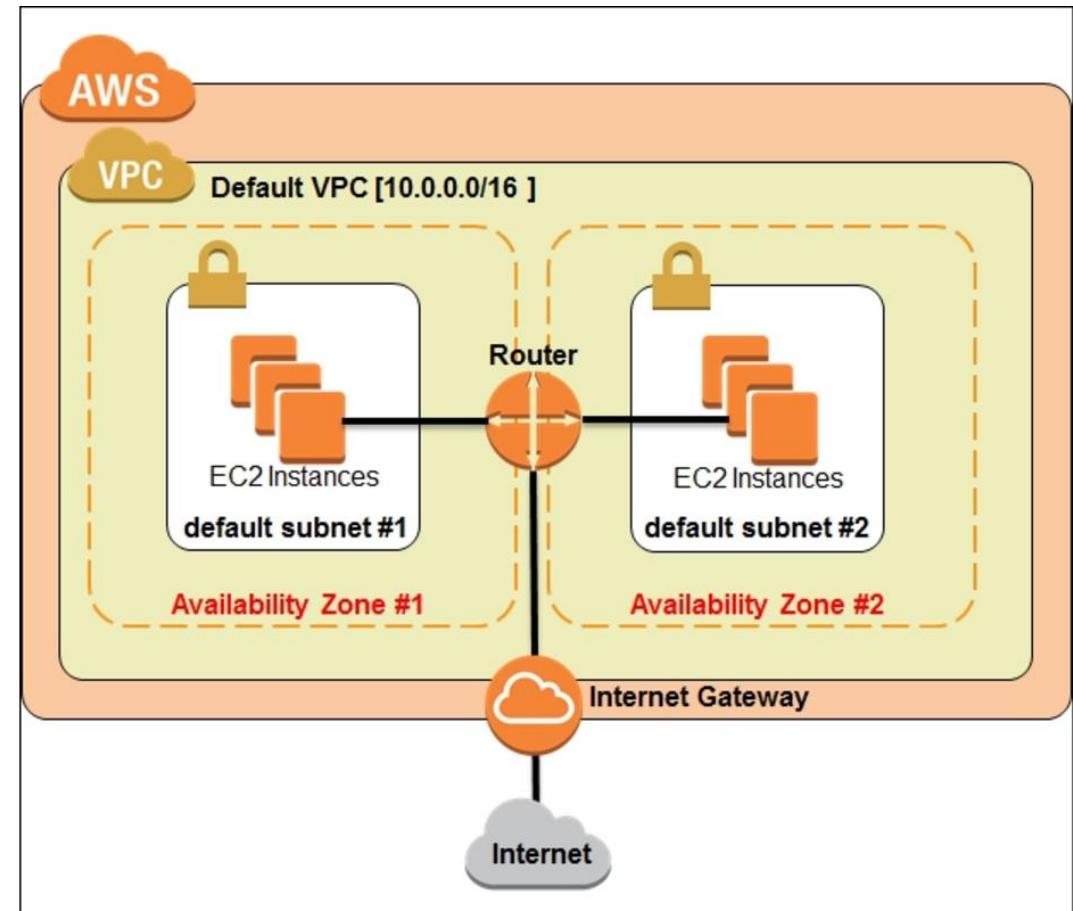


Control inbound and outbound access to subnets by using **Network Access Control Lists**.

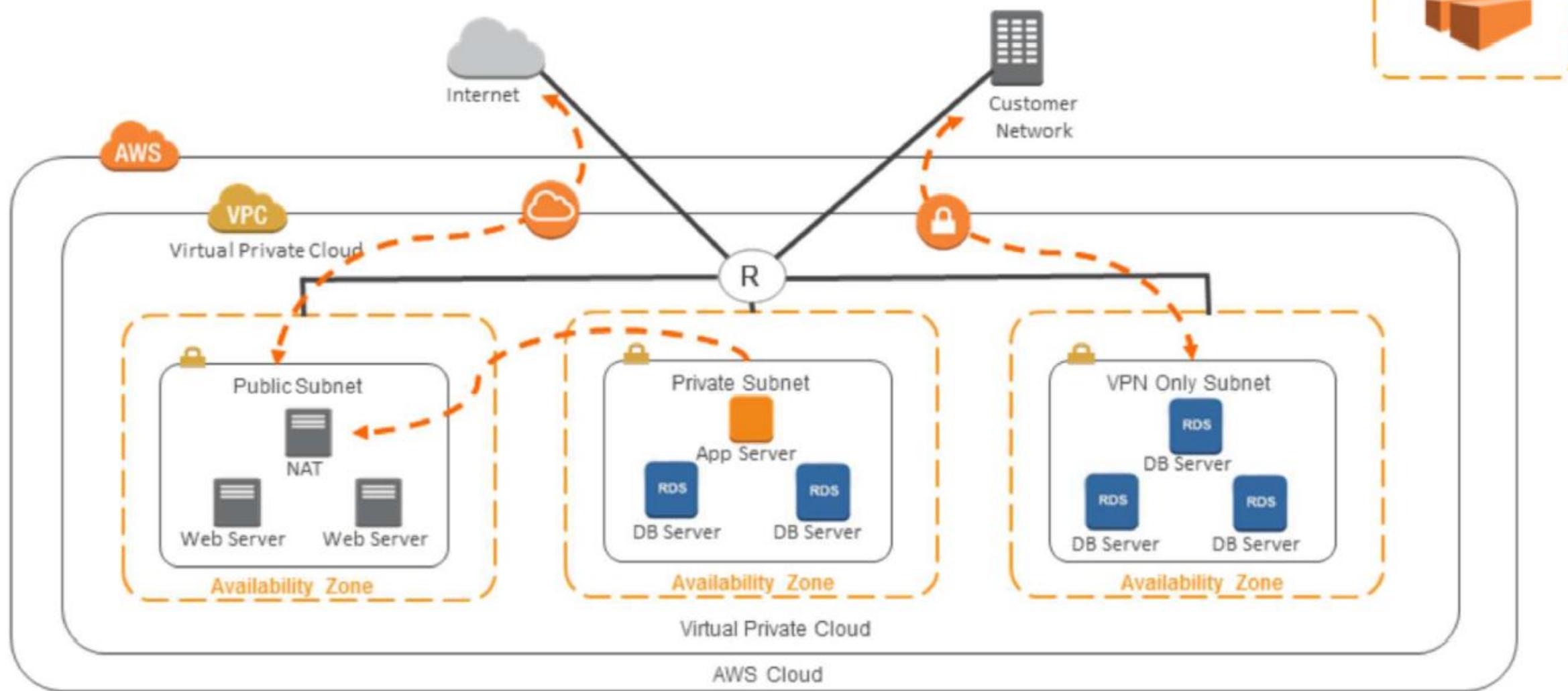
Attach an Amazon **Elastic IP address**—a type of static, public IP address—to any instance in your VPC to ensure continuity of access to the instance from the Internet.

Default VPC Key Concepts

- Subnets ~ a range of valid IP addresses that you specify
 - Private
 - Public
- The default VPC is always created with a CIDR block of /16, which means it supports 65,536 IP addresses in it.
- A default subnet is created in each AZ of your selected region. Instances launched in these default subnets have both a public and a private IP address by default as well.
- An Internet Gateway is provided to the default VPC for instances to have Internet connectivity.
- A few necessary route tables, security groups, and ACLs are also created by default that enable the instance traffic to pass through to the Internet.



Amazon VPC Example



To be seen

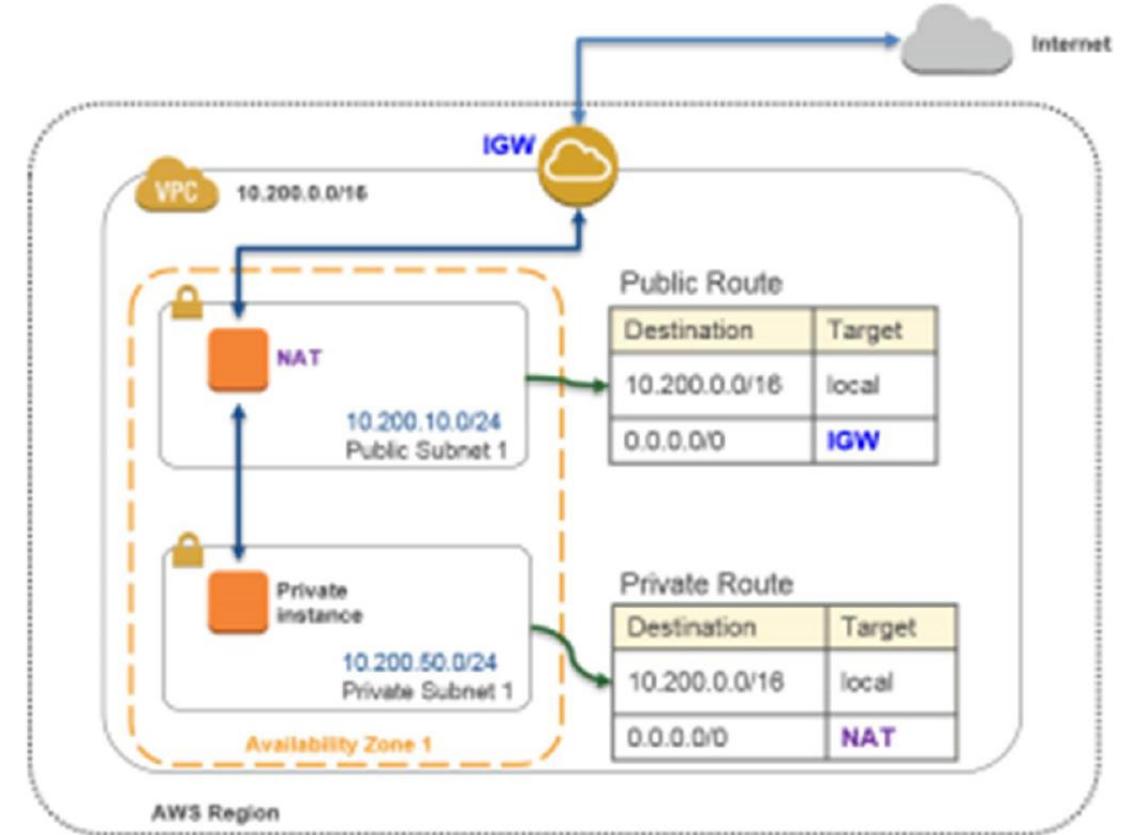
- VPC
 - <https://www.youtube.com/watch?v=jcyZmj6Ywh4> (6:32)
- IAM
 - <https://www.youtube.com/watch?v=Z4U5ymvEvKc>

Homework VPC

- <http://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/ExerciseOverview.html>

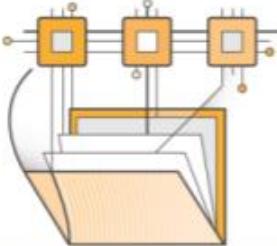
Assignment

- Create a new VPC (LabVPC), with 2 subnets, an internet gateway to allow traffic from the internet.
- Manually create a routing tables and associate them to your subnets
- Manually create a NAT EC2 instance inside your public subnet, and an Linux instance inside your private subject
- Make sure your you can connect to your private instance via the NAT Instance



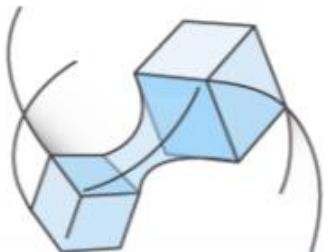
Storage

AWS storage maturity

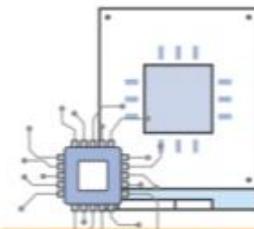


Amazon EFS

File



Amazon EBS

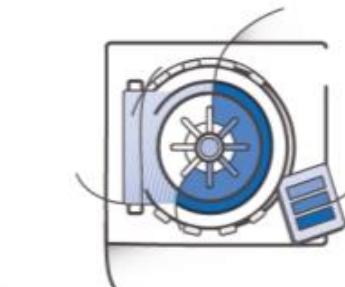


Amazon EC2
instance store

Block



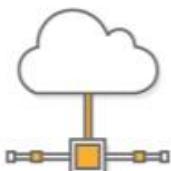
Amazon S3



Amazon Glacier

Object

Data transfer



AWS Direct
Connect



Snowball



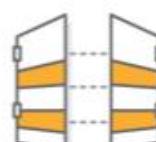
ISV connectors



Amazon
Kinesis
Firehose



Transfer
Acceleration



AWS Storage
Gateway

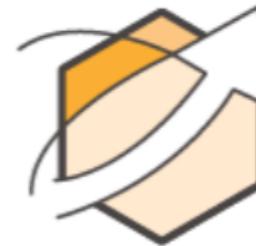
Value Proposition



Durable
11 9s



Available
Designed for 99.99%



Scalable
Gigabytes -> Exabytes

If You Need:

Consider Using:

Persistent local storage for Amazon EC2, for relational and NoSQL databases, data warehousing, enterprise applications, Big Data processing, or backup and recovery

[Amazon Elastic Block Storage \(Amazon EBS\)](#)

A file system interface and file system access semantics to make data available to one or more EC2 instances, for content serving, enterprise applications, media processing workflows, Big Data storage or backup and recovery

[Amazon Elastic File System \(Amazon EFS\)](#)

A scalable, durable platform to make data accessible from any Internet location, for user-generated content, active archive, serverless computing, Big Data storage or backup and recovery

[Amazon Simple Storage Service \(Amazon S3\)](#)

Highly affordable long-term storage that can replace tape for archive and regulatory compliance

[Amazon Glacier](#)

A hybrid storage cloud augmenting your on-premises environment with Amazon cloud storage, for bursting, tiering or migration

[AWS Storage Gateway](#)

A portfolio of services to help simplify and accelerate moving data of all types and sizes into and out of the AWS cloud

[Cloud Data Migration Services](#)

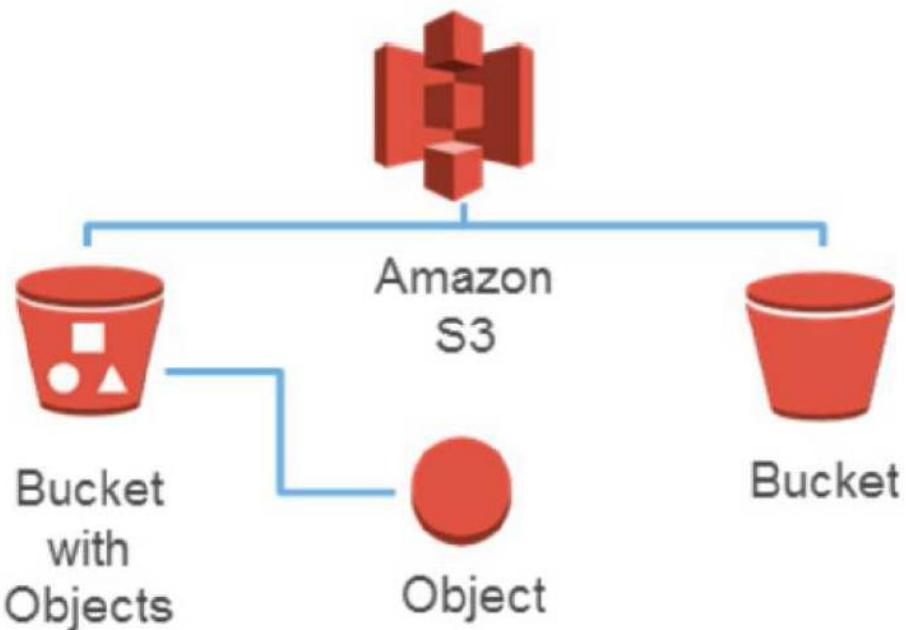
S3

- Amazon Simple Storage Service (Amazon S3) is object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the web. It is designed to deliver 99.99999999% durability, and scale past trillions of objects worldwide.
- Customers use S3 as primary storage for cloud-native applications; as a bulk repository, or "[data lake](#)," for analytics; as a target for [backup & recovery](#) and disaster recovery; and with [serverless computing](#).
- It's simple to move large volumes of data into or out of Amazon S3 with Amazon's [cloud data migration](#) options. Once data is stored in S3, it can be automatically tiered into lower cost, longer-term [cloud storage](#) classes like S3 Standard - Infrequent Access and [Amazon Glacier](#) for archiving.



Introduction to Amazon S3

S3 Key Concepts



- Amazon S3 stores data as objects within buckets
- An object is comprised of a file and optionally any metadata that describes that file
- You can have up to 100 buckets in each account
- You can control access to the bucket and its objects

S3 Buckets

- A bucket is a container in Amazon S3 where the files are uploaded. For using Amazon S3 to store a file, you need to create at least one bucket. Files (objects) are stored in buckets.
- The following are a few features of buckets:
 - The bucket name should be unique because it is shared by all users.
 - Buckets can contain logical nested folders and subfolders. But it cannot contain nested buckets.
 - You can create a maximum of 100 buckets in a single account.
 - The bucket name can contain letters, numbers, periods, dash, and the underscore.
 - The bucket name should start with a letter or number, and it should be between 3 to 25 characters long.
- Buckets can be managed via the following:
 - REST-style HTTP interface
 - SOAP interface
- A bucket doesn't have any size restrictions for the user. It can store objects of any size.
- Buckets can be accessed via HTTP URLs as follows:
 - `http://< BUCKET_NAME>.s3.amazonaws.com/< OBJECT_NAME >`
 - `http://s3.amazonaws.com/< BUCKET_NAME >/< OBJECT_NAME >`
- In the preceding URLs, BUCKET_NAME will be the name of the bucket that you provided while creating it. And OBJECT_NAME will be the name of the object that you provided while creating the object.

S3 Objects

- An object is a stored file in Amazon S3. Each object consists of a unique identifier, the user who uploaded the object, and permissions for other users to perform CRUD operations on it. Every object is stored in a bucket.
- Objects can be managed via the following:
 - REST-style HTTP interface
 - SOAP interface
- Objects can be downloaded via the following:
 - HTTP GET interface
 - BitTorrent protocol
- The bucket can consist of any type of object, be it a PDF, text, video, audio, or any other kind of files
- While creating an object, a key will be assigned to the object. This key will be used for retrieving the object. The key should have the following features:
 - Be unique in the bucket
 - Contain alphabets, numbers, and special characters such as-, !, _, ., *, ', (, and)

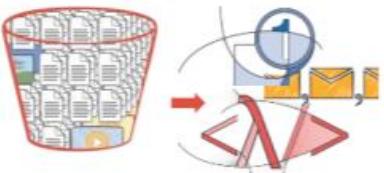
S3 Key Features

- **Allows website hosting:** Amazon S3 allows users to host a website and map it to their domain. This is very cost effective, because the user pays only for what he/she uses. Moreover, the user doesn't require highly configured servers to serve the website.
- **Scalable:** Amazon S3 doesn't restrict the user to any size limit for storing data. As it is a pay-as-you-go service, it stores the data, and the bill is generated accordingly. So the subscriber never faces a lack of space.
- **Reliable:** Amazon S3 guarantees a server uptime of 99.9 percent. Therefore, the subscriber does not need to worry about data reliability.
- **Security:** Amazon S3 provides a strong authentication mechanism where the stored data can be manipulated.
- **Standard interfaces:** Amazon S3 provides the **Representational State Transfer (REST)** and **Simple Object Access Protocol (SOAP)** web services that can be consumed by any web framework.
- **Reduced Redundancy Storage:** Amazon S3 provides the subscribers with an optional feature for storing data with the **Reduced Redundancy Storage (RRS)** storage class. It is basically used for storing non-critical and reproducible data at lower levels of redundancy. The cost of storing on an RRS storage class is quite less as compared to the standard storage class.
- **Torrent tracking and seeding:** Amazon S3 can act as a torrent tracker, and seed the files from your machine.
- **Share the data with a temporary URL:** Amazon S3 provides the subscriber the ability to share a URL, which auto-expires after a period of time. This helps the subscriber in sharing the data for a minimal period of time. Other users cannot use that data after the URL expires.
- **Logging:** It provides the logging of all activities that are performed on bucket. This makes it easy for the subscriber to audit the activities on the bucket if he so wishes. Generally, when a subscriber hosts a website on Amazon S3, he enables the logging feature to track the activities.
- **Versioning:** Amazon S3 allows storing of multiple versions of an object. It is basically used for recovering old data that is lost unintentionally.
- **Security:** Amazon S3 provides security on buckets and objects. While creating the buckets, you can provide access control lists for other users of the bucket who can create, update, delete, or list objects. You can even set the geographical location of your data.
- **Integration:** Amazon S3 can be integrated with several other services such as Amazon EC2, Amazon EBS, Amazon Glacier, and many other applications. Generally, developers use Amazon S3 for storing images, videos, or documents, and for accessing them via HTTP Get.

S3 Cases Studies

- The Amazon S3 can be utilized for different purposes:
 - **File hosting:** Companies often deploy their images, videos, audios, PDFs, DOCs, and other files in Amazon S3. This helps in loading the files directly from Amazon S3 without managing the on-premise infrastructure.
 - **Storing data on mobile-based applications:** Many users/companies go for Amazon S3 to store mobile app data. This becomes easy for user/companies to manage mobile user data over Amazon S3.
 - **Static website hosting:** Users can host their static website over Amazon S3 along with Amazon Route53.
 - **Video Hosting:** Companies upload their videos over Amazon S3, which can then be accessed on their website. Amazon S3 can also be configured to provide video streaming.
 - **Backup:** Users can keep a backup of their data, which will be securely and reliably stored in Amazon S3. Amazon S3 can also be configured to move the old data over to Amazon Glacier for archiving, as the Glacier costs less as compared to S3.

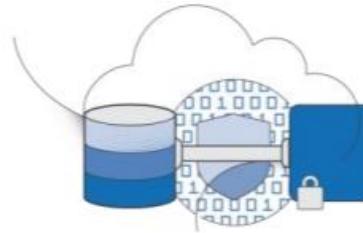
S3 Innovations



Event notifications



Cross-region
replication



VPC endpoint
for Amazon S3



- Amazon CloudWatch metrics for Amazon S3
- AWS CloudTrail support

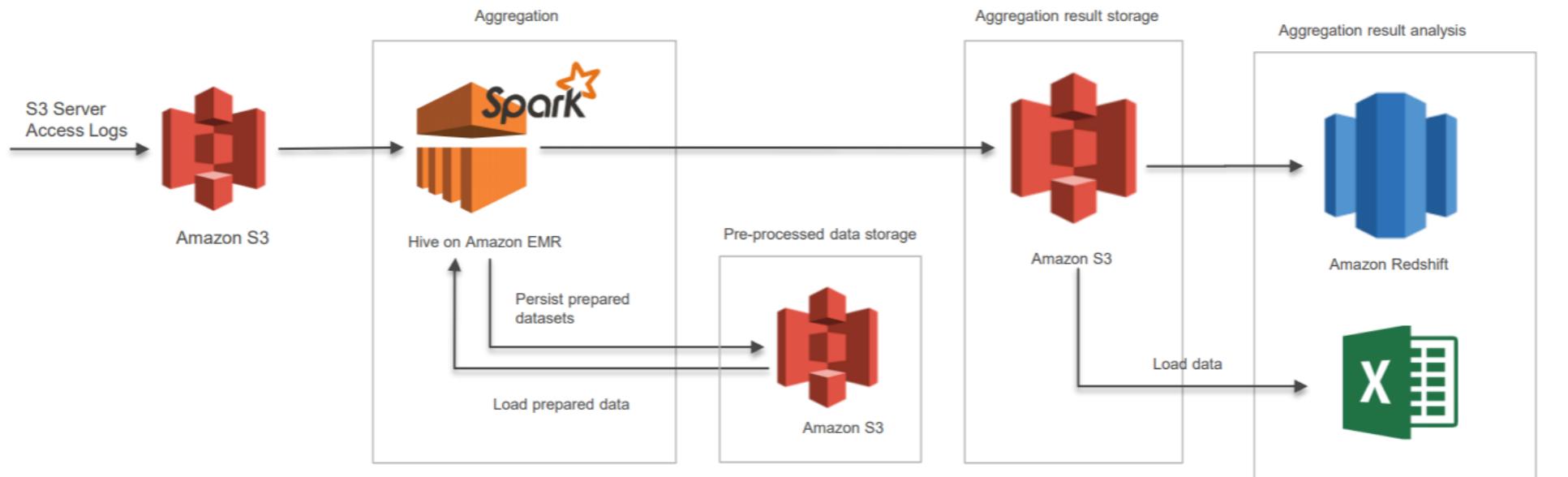


Amazon S3 bucket
limit increase



Read-after-write
consistency in all regions

S3 and Data Analytics



1. Enable Access Logs

2. Create EMR Cluster
3. Spark code to aggregate logs
4. Submit code to EMR

5. Persist interim results on S3

6. Persist final results on S3

7. Visualize Data

Using the AWS Management Console

- The Amazon S3 [Getting Started Guide](#) shows you how to start using the AWS Management Console with Amazon S3. To get started you:
 - 1. [Sign up for Amazon S3](#)
 - 2. [Create a Bucket](#)
 - 3. [Add an Object to a Bucket](#)
 - 4. [View an Object](#)
 - 5. [Move an Object](#)
 - 6. [Delete an Object and Bucket](#)
 - 7. [Where Do I Go From Here](#)



Homework

- Use S3 to create a very simple web site that can be viewed by any internet user
 - <http://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html>
 - <http://docs.aws.amazon.com/AmazonS3/latest/dev/hosting-websites-on-s3-examples.html>

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
    <h1>Welcome to my website</h1>
    <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

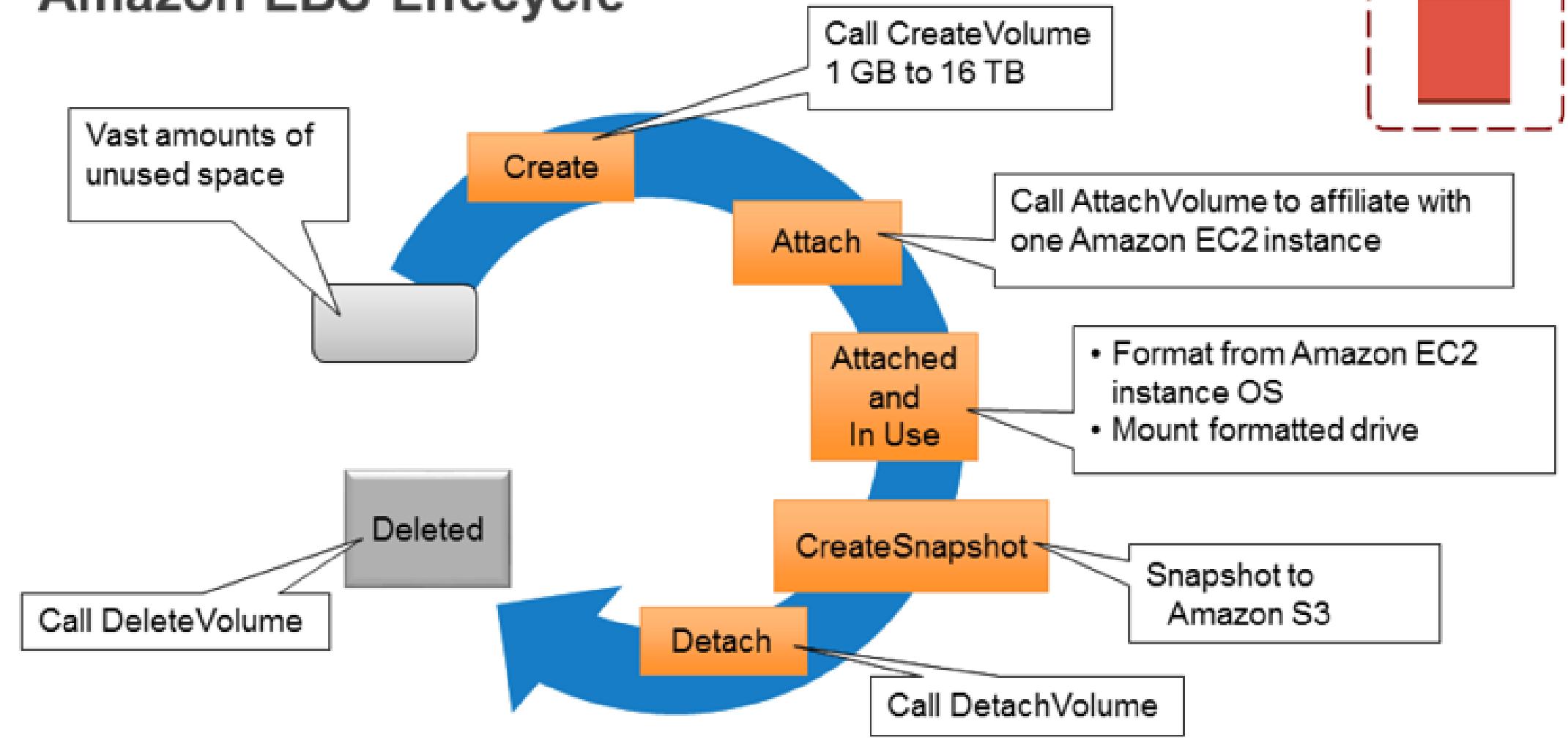
EBS



Amazon
EBS

- **Persistent block level storage** volumes offering consistent and low-latency performance
- Automatically replicated within its Availability Zone
- Snapshots stored durably in Amazon S3

Amazon EBS Lifecycle



EBS vs S3

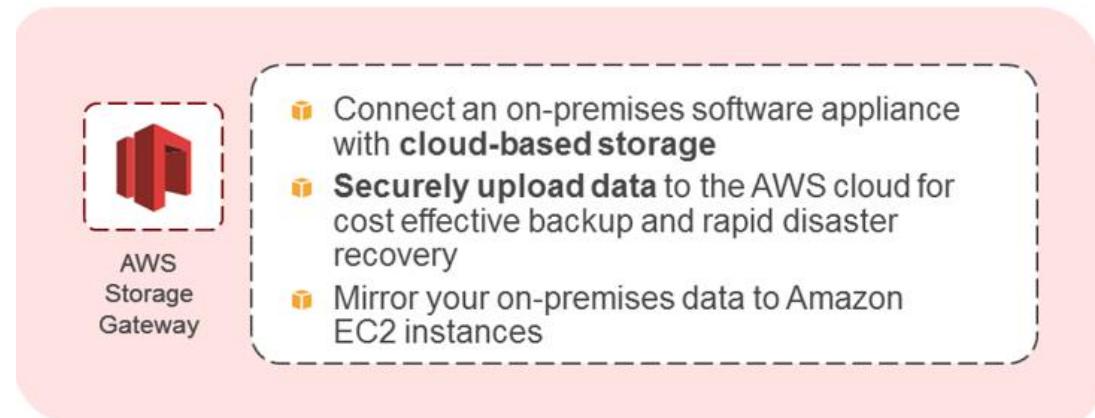
	Amazon EBS 	Amazon S3 
Paradigm	File system	Object store
Performance	Very fast	Fast
Redundancy	Across multiple servers in an Availability Zone	Across multiple facilities and on multiple devices within each facility
Security	EBS Encryption – Data volumes and Snapshots	Server Side Encryption
Access from the Internet?	No (1)	Yes (2)
Typical use case	It is a disk drive	Write once, read many

(1) Accessible from the Internet if mounted to server and set up as FTP, etc.

(2) Only with proper credentials, unless ACLs are world-readable

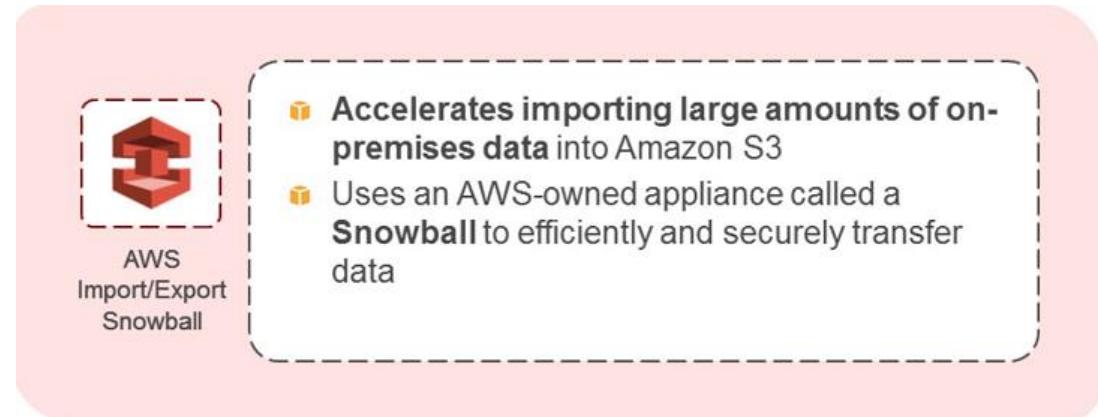
Storage Gateway

- AWS Storage Gateway provides low-latency performance by maintaining frequently accessed data on-premises while securely storing all of your data encrypted in Amazon S3. It is a service connecting an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and the AWS' storage infrastructure.
- Using the AWS Storage Gateway, you can back up point-in-time snapshots of your on-premises application data to Amazon S3 for future recovery. If you need to replace capacity for disaster recovery purposes, or leverage Amazon EC2's on-demand compute capacity for additional capacity during peak periods, new projects, or as a more cost-effective way to run normal workloads, you can use the AWS Storage Gateway to mirror your on-premises data to Amazon EC2 instances.



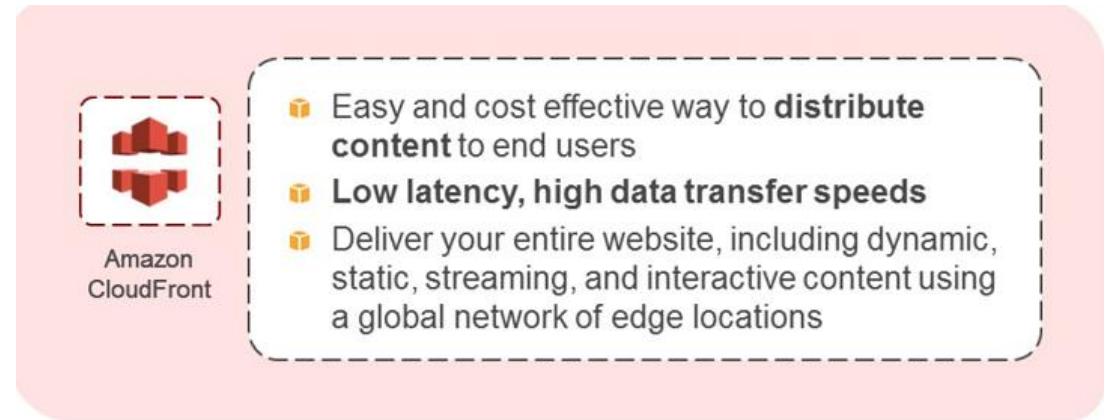
Snowball

- The Snowball is a physical, self-contained, tamper- and weather-resistant appliance used to efficiently and securely transfer data from your data center to AWS where the data is imported into Amazon S3 buckets.

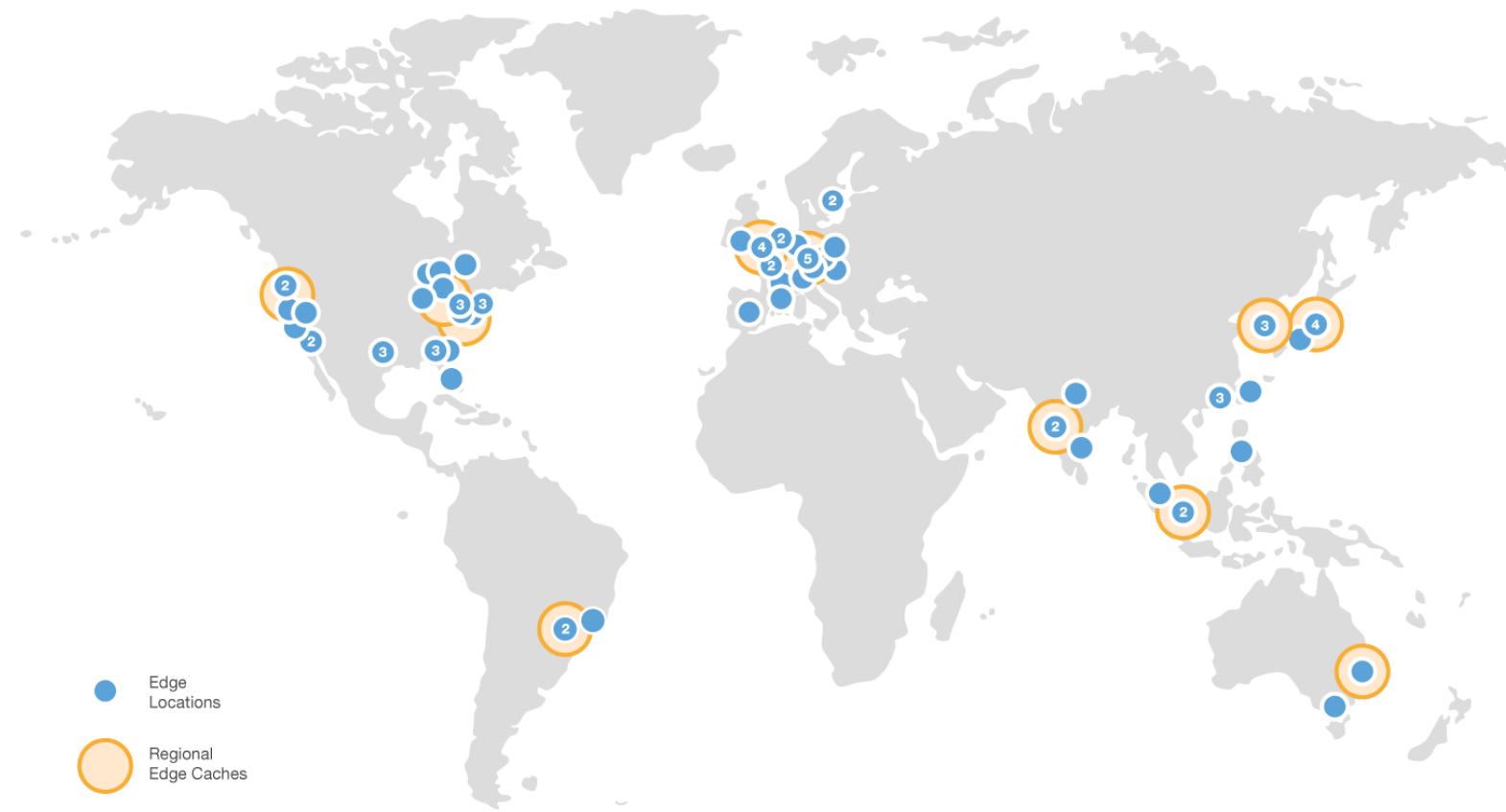


Cloud Front

- Amazon CloudFront integrates with other Amazon Web Services to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no minimum commitments. You can use Amazon CloudFront to deliver your entire website
- Amazon CloudFront uses a global network of 79 edge locations and 11 regional edge caches across 22 countries and 49 cities for content delivery



Cloud Front Edge Locations



Cloud Front Homework

- <https://aws.amazon.com/getting-started/tutorials/deliver-content-faster/>

Compute

EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
- Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate them from common failure scenarios.



EC2 Benefits



Elastic Web-Scale Computing

Amazon EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds, or even thousands of server instances simultaneously. Because this is all controlled with web service APIs, your application can automatically scale itself up and down depending on its needs.



Completely Controlled

You have complete control of your instances including root access and the ability to interact with them as you would any machine. You can stop any instance while retaining the data on the boot partition, and then subsequently restart the same instance using web service APIs. Instances can be rebooted remotely using web service APIs, and you also have access to their console output.



Flexible Cloud Hosting Services

You have the choice of multiple instance types, operating systems, and software packages. Amazon EC2 allows you to select a configuration of memory, CPU, instance storage, and the boot partition size that is optimal for your choice of operating system and application. For example, choice of operating systems includes numerous Linux distributions and [Microsoft Windows Server](#).



Integrated

Amazon EC2 is integrated with most AWS services such as Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), and Amazon Virtual Private Cloud (Amazon VPC) to provide a complete, secure solution for computing, query processing, and cloud storage across a wide range of applications.



Reliable

Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and predictably commissioned. The service runs within Amazon's proven network infrastructure and data centers. The Amazon EC2 Service Level Agreement commitment is 99.95% availability for each Amazon EC2 Region.



Secure

Cloud security at AWS is the highest priority. As an AWS customer, you will benefit from a data center and network architecture built to meet the requirements of the most security-sensitive organizations. Amazon EC2 works in conjunction with Amazon VPC to provide security and robust networking functionality for your compute resources.



Inexpensive

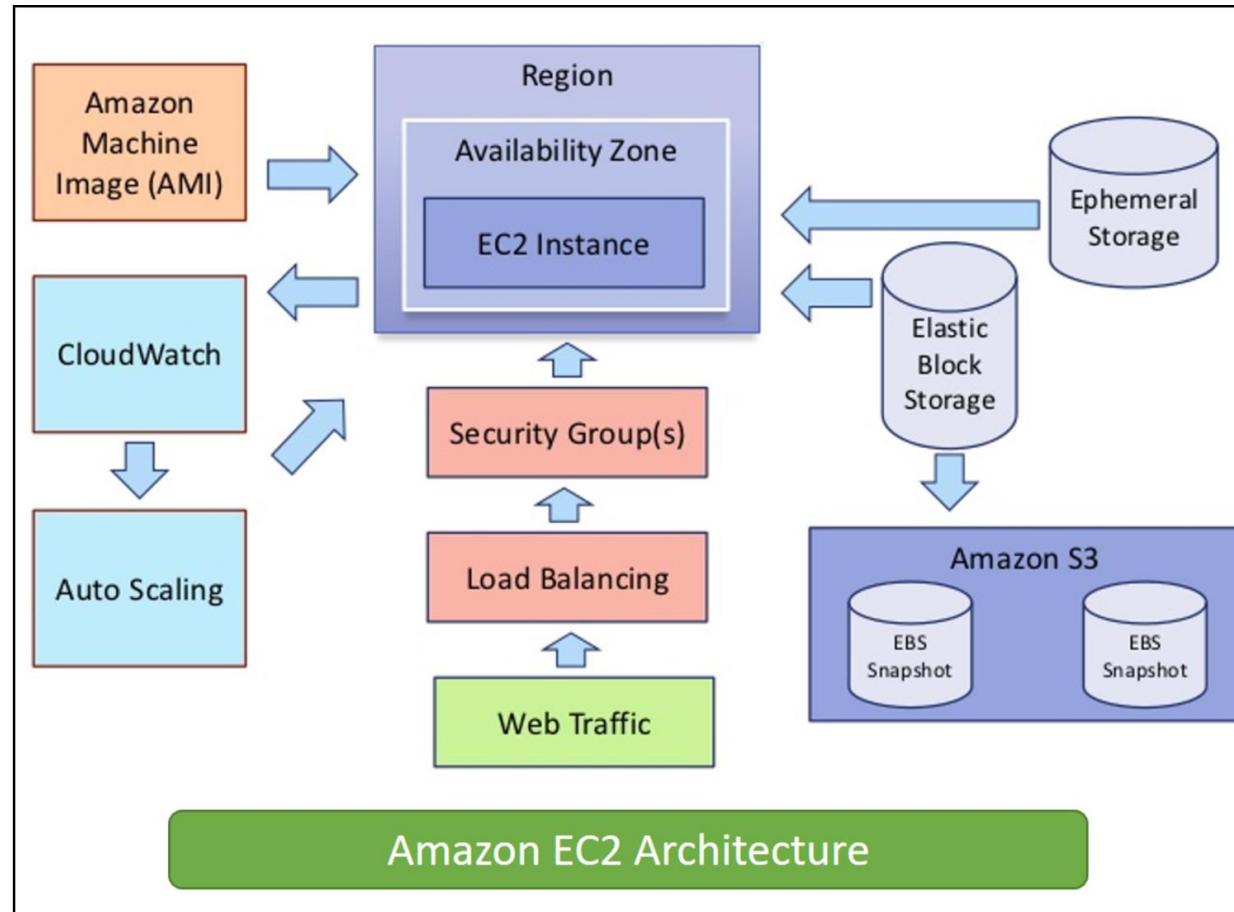
Amazon EC2 passes on to you the financial benefits of Amazon's scale. You pay a very low rate for the compute capacity you actually consume. See [Amazon EC2 Instance Purchasing Options](#) for more details.



Easy to Start

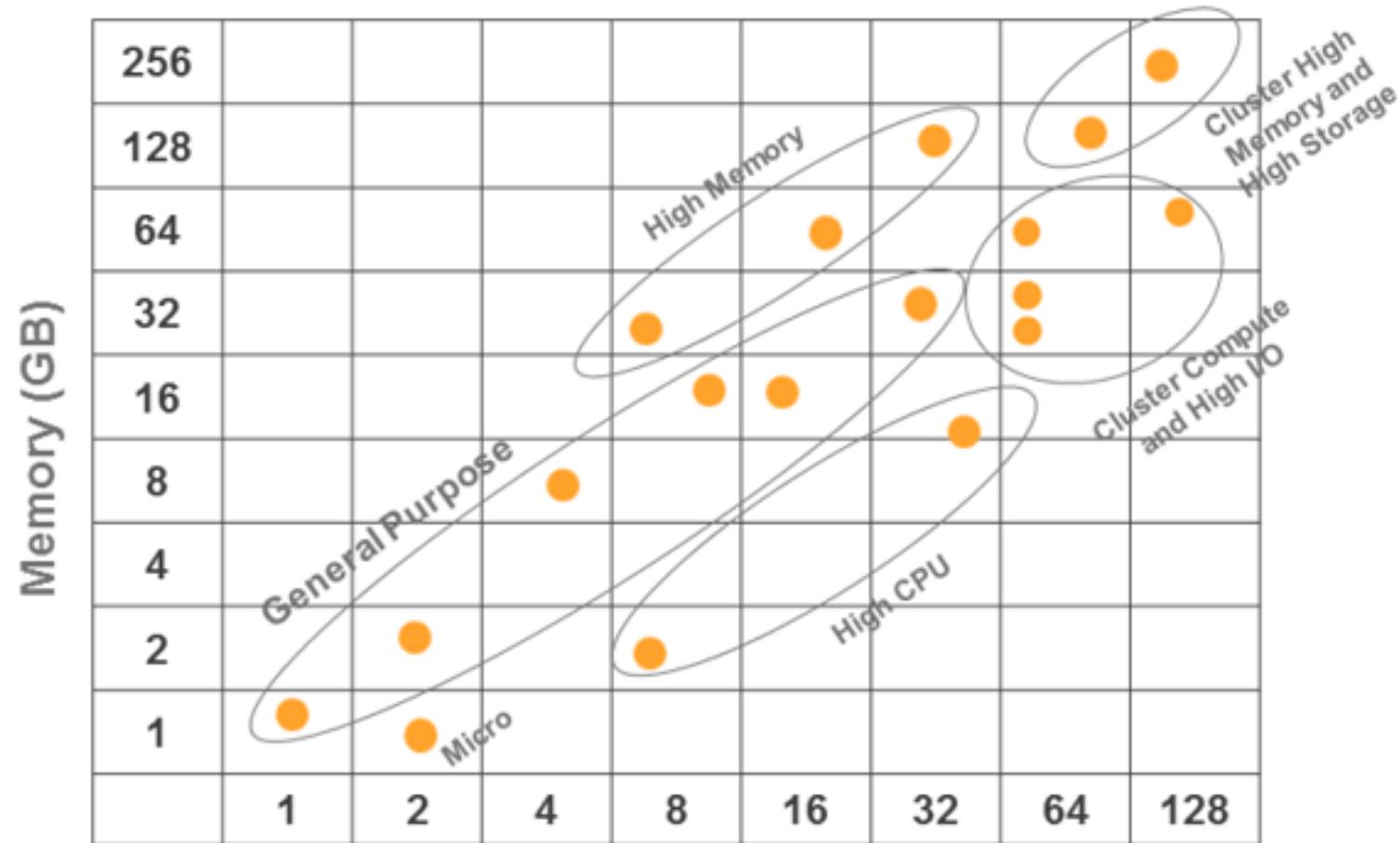
There are several ways to get started with Amazon EC2. You can use the [AWS Management Console](#), the AWS Command Line Tools (CLI), or [AWS SDKs](#). AWS is [free](#) to get started. To learn more, please visit our [tutorials](#).

EC2 Architecture



Source: Raja Malleswara Rao Pattamsetti, 2017

EC2 Instances



Choosing the right instance

- General Purpose Instances
 - T2 instances are a low-cost, burstable performance instance type that provide a baseline level of CPU performance with the ability to burst above the baseline. They offer a balance of compute, memory, and network resources for workloads that occasionally need to burst, such as web servers, build servers, and development environments.
 - M3 and M4 instances provide a balance of compute, memory, and network resources. These instances are ideal for applications that require high CPU and memory performance, such as encoding applications, high traffic content management systems, and memcached applications.
- Compute-Optimized Instances
 - C3 and C4 instances are optimized for compute-intensive workloads. These instances have proportionally more CPU than memory (RAM). They are well suited to applications such as high performance web servers, batch processing, and high-performance scientific and engineering applications.
- Memory-Optimized Instances
 - R3 instances are optimized for memory-intensive workloads. These instances offer large memory sizes for high throughput applications such as high performance databases, distributed memory caches, in-memory analytics, and large enterprise deployments of software such as SAP.
- GPU Instances
 - G2 instances are optimized for graphics and graphic processing unit (GPU) compute applications, such as machine learning, video encoding, and interactive streaming applications.
- Storage-Optimized Instances
 - I2 instances are optimized for storage and high random I/O performance, such as NoSQL databases, scale-out transactional databases, data warehousing, Hadoop, and cluster file systems.

Instance Types

Type	Feature Highlight	Use Case
General purpose	T2 <ul style="list-style-type: none">• Lowest-cost general-purpose instance type• Balance of compute, memory, and network resources	<ul style="list-style-type: none">• Development environments• Build servers• Code repositories• Low-traffic web applications• Early product experiments• Small databases
	M3 <ul style="list-style-type: none">• SSD-based instance storage for fast I/O performance• Balance of compute, memory, and network resources	<ul style="list-style-type: none">• Small and mid-size databases• Data processing tasks that require additional memory• Caching fleets and for running backend servers for SAP• Microsoft SharePoint and other enterprise applications

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

Type	Feature Highlight	Use Case
Compute-Optimized	C3 <ul style="list-style-type: none">• Support for enhanced networking• Support for clustering• SSD-backed instance storage	<ul style="list-style-type: none">• High-performance front-end fleets• Web servers• On-demand batch processing• High-performance science and engineering applications <ul style="list-style-type: none">• Distributed analytics• Ad serving• Batch processing• MMO gaming• Video encoding• Distributed analytics

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

Type	Feature Highlight	Use Case
Compute-Optimized	C4 <ul style="list-style-type: none">• Uses Intel Xeon E5-2666v3 (code name Haswell) processor• Support for enhanced networking• EBS-optimized• Support for clustering• SSD-backed instance storage	<ul style="list-style-type: none">• High-performance front-end fleets• Top-end web servers• Risk analysis and rendering• High-performance science and engineering applications <ul style="list-style-type: none">• Distributed analytics• Ad serving• Batch processing• Online gaming• Simulation• Distributed analytics

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

Instance Size	vCPU Count	RAM	Network Performance	Dedicated EBS Throughput
c4.large	2	3.75 GiB	Moderate	500 Mbps
c4.xlarge	4	7.5 GiB	Moderate	750 Mbps
c4.2xlarge	8	15 GiB	High	1,000 Mbps
c4.4xlarge	16	30 GiB	High	2,000 Mbps
c4.8xlarge	36	60 GiB	10 Gbps	4,000 Mbps

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

Type	Feature Highlight	Use Case
Memory-Optimized R3	<ul style="list-style-type: none">• Lowest price point per GB of RAM• SSD Storage• Support for enhanced networking	<ul style="list-style-type: none">• High-performance databases• Distributed memory caches• In-memory analytics• Genome assembly and analysis• Larger deployments of SAP, Microsoft SharePoint, and other enterprise applications

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

Type	Feature Highlight	Use Case
GPU G2	<ul style="list-style-type: none">High-performance NVIDIA GPU with 1,536 CUDA cores and 4 GB of video memoryOn-board hardware video encoder designed to support up to eight real-time HD video streams	<ul style="list-style-type: none">Game streamingVideo encoding3D application streamingOther server-side graphics workloads

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

Type	Feature Highlight	Use Case
Storage-Optimized	I2	<ul style="list-style-type: none">SSD StorageSupport for TRIMSupport for enhanced networkingHigh random I/O performance <ul style="list-style-type: none">NoSQL databases (e.g., Cassandra and MongoDB)Scale out transactional databasesData warehousingHadoopCluster file systems
	D2	<ul style="list-style-type: none">Up to 48 TB of HDD-based storageHigh disk throughputLowest price per disk throughput performance on Amazon EC2Amazon EC2 Enhanced Networking support <ul style="list-style-type: none">Massively Parallel Processing (MPP) data warehousingMapReduce and Hadoop distributed computingDistributed file systemsNetwork file systemsLog or data-processing applications

More information: <https://aws.amazon.com/ec2/instance-types/>

Instance Types

	Local Storage	EBS-Backed Storage
Storage Volume	Storage volume is defined by the instance type that you chose	User defined
Over the network	No; Local direct attach storage	Yes
Persistence	<p>Non-persisted storage</p> <ul style="list-style-type: none">• Temporal in nature• When the Amazon EC2 instance is stopped or terminated, the resources allocated for the Amazon EC2 instance are released back to the fleet	<p>Persistent storage</p> <ul style="list-style-type: none">• User can create snapshots• If “<i>DeleteOnTermination</i>” attribute is set to “<i>false</i>”, the EBS volume will persist after termination of the EC2 instance to which it was attached

More information: <https://aws.amazon.com/ec2/instance-types/>

Intel Technologies

	Burstable	Balanced	Compute	Memory	GPU	I/O	Storage
AWS Instance Type	T2	M4	C4	R3	G2	I2	D2
Intel® processor	Intel® Xeon® family	Intel® Xeon® E5-2676 v3	Intel® Xeon® E5-2666 v3	Intel® Xeon® E5-2670 v2	Intel® Xeon® E5-2670	Intel® Xeon® E5-2670 v2	Intel® Xeon® E5-2676 v3
Intel® process technology	●	22nm Haswell	22nm Haswell	22nm Ivy Bridge	32nm Sandy Bridge	22nm Ivy Bridge	22nm Haswell
Intel® AVX	●	●	●	●	●	●	●
Intel® AVX2		●	●				●
Intel® Turbo Boost	●	●	●	●	●	●	●
Storage	EBS only	EBS only	EBS only	SSD	SSD	SSD	HDD

EC2 Storage Types

	Local Storage	EBS-Backed Storage
Storage Volume	Storage volume is defined by the instance type that you chose	User defined
Over the network	No; Local direct attach storage	Yes
Persistence	<p>Non-persisted storage</p> <ul style="list-style-type: none">• Temporal in nature• When the Amazon EC2 instance is stopped or terminated, the resources allocated for the Amazon EC2 instance are released back to the fleet	<p>Persistent storage</p> <ul style="list-style-type: none">• User can create snapshots• If “<i>DeleteOnTermination</i>” attribute is set to “<i>false</i>”, the EBS volume will persist after termination of the EC2 instance to which it was attached

Purchase Options

On-Demand Instances	Reserved Instances	Spot Instances
<ul style="list-style-type: none">✓ Pay as you go for compute power✓ Pay only for what you use, no up-front commitments or long-term contracts 	<ul style="list-style-type: none">✓ 1- or 3-year terms✓ Reserve computing capacity at a significantly discounted hourly rate compared to On-Demand Instance pricing✓ Compute capacity is available when needed	<ul style="list-style-type: none">✓ Bid on unused Amazon EC2 capacity✓ Spot Price based on supply/demand, determined automatically✓ Spot Price below bid, instances start✓ Spot Price above bid, instances terminate

Amazon EC2 Quick Start

adapted from

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Ec2_GetStarted.html

Amazon Elastic Compute Cloud (EC2)

- Amazon Machine Images (**AMIs**) are the basic building blocks of Amazon EC2
- An AMI is a template that contains a software configuration (operating system, application server and applications) that can run on Amazon's computing environment
- AMIs can be used to launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud.

Getting Started with Amazon EC2

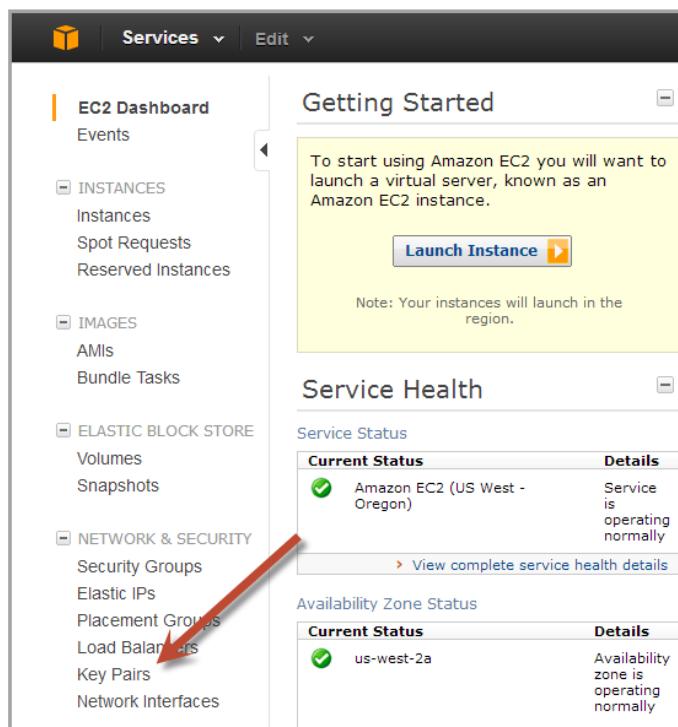
- Step 1: Sign up for Amazon EC2
- Step 2: Create a key pair
- Step 3: Launch an Amazon EC2 instance
- Step 4: Connect to the instance
- Step 5: Customize the instance
- Step 6: Terminate instance and delete the volume created

Creating a key pair

- AWS uses public-key cryptography to encrypt and decrypt login information.
- AWS only stores the public key, and the user stores the private key.
- There are two options for creating a key pair:
 - Have Amazon EC2 generate it for you
 - Generate it yourself using a third-party tool such as OpenSSH, then import the public key to Amazon EC2

Generating a key pair with Amazon EC2

1. Open the Amazon EC2 console at
<http://console.aws.amazon.com/ec2/>
2. On the navigation bar select region for the key pair
3. Click **Key Pairs** in the navigation pane to display the list of key pairs associated with the account

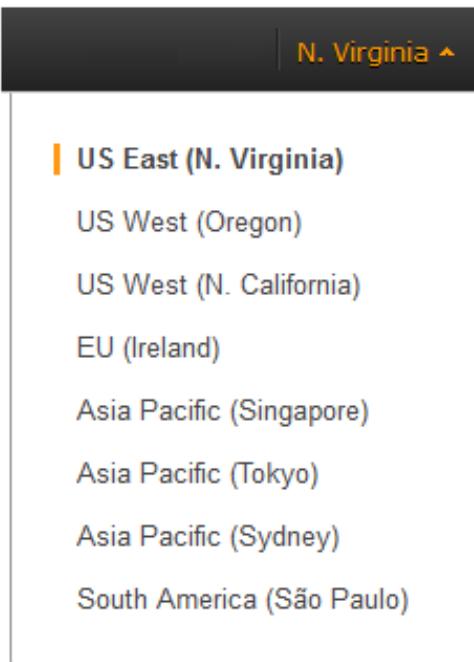


Generating a key pair with EC2 (cont.)

4. Click **Create Key Pair**
5. Enter a name for the key pair in the **Key Pair Name** field of the dialog box and click **Create**
6. The private key file, with .pem extension, will automatically be downloaded by the browser.

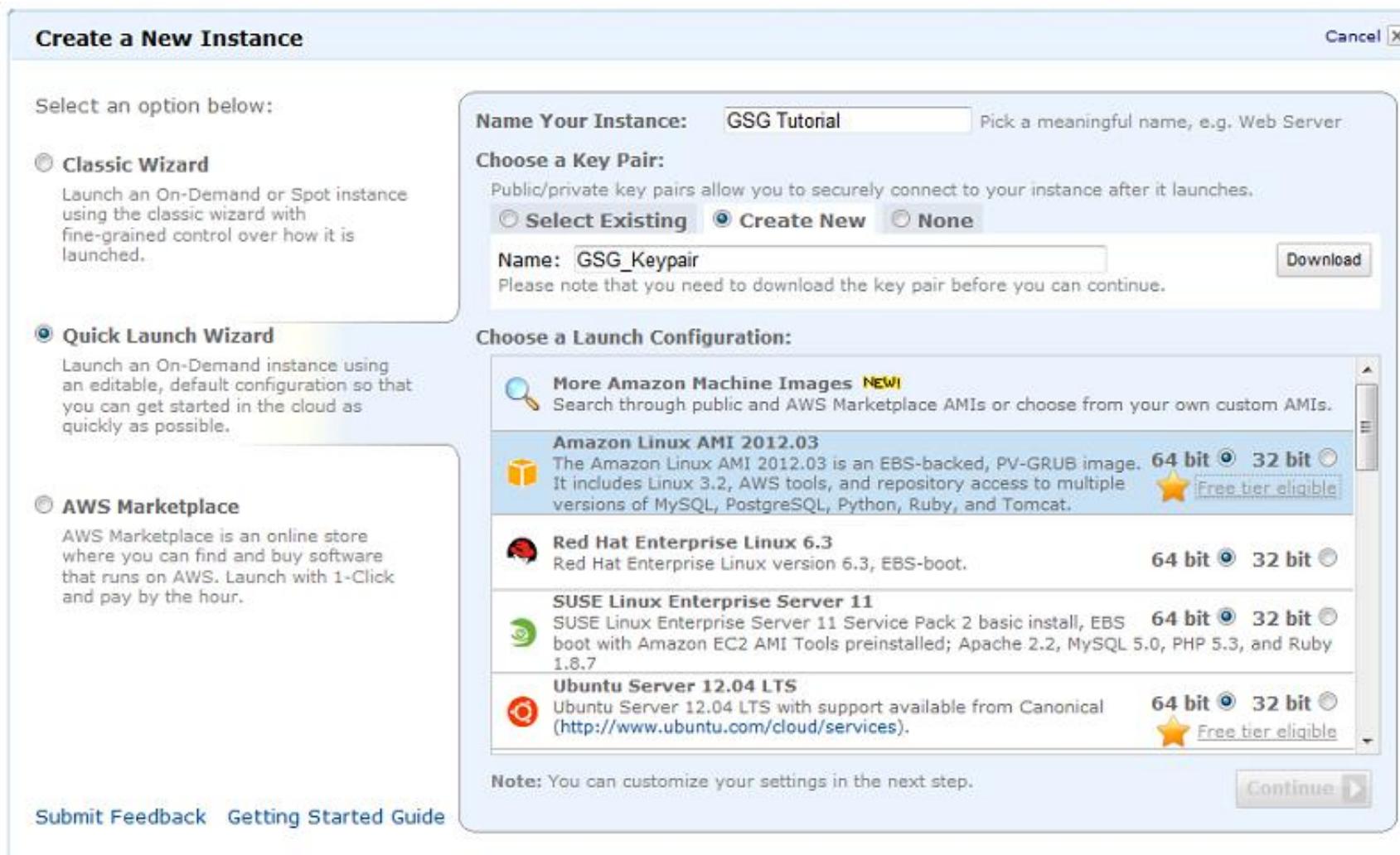
Launching an Amazon EC2 instance

1. Sign in to AWS Management Console and open the Amazon EC2 console at
<http://console.aws.amazon.com/ec2/>
2. From the navigation bar select the region for the instance



Launching an Amazon EC2 instance (cont.)

- From the Amazon EC2 console dashboard, click **Launch Instance**



Launching an Amazon EC2 instance (cont.)

4. On the **Create a New Instance** page, click **Quick Launch Wizard**
5. In **Name Your Instance**, enter a name for the instance
6. In **Choose a Key Pair**, choose an existing key pair, or create a new one
7. In Choose a Launch Configuration, a list of basic machine configurations are displayed, from which an instance can be launched
8. Click continue to view and customize the settings for the instance

Launching an Amazon EC2 instance (cont.)

9. Select a security group for the instance. A **Security Group** defines the firewall rules specifying the incoming network traffic delivered to the instance. Security groups can be defined on the Amazon EC2 console, in **Security Groups** under **Network and Security**

Security Group: quicklaunch-1

Inbound

Create a new rule: Custom TCP rule

Port range: (e.g., 80 or 49152-65535)

Source: 0.0.0.0/0 (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Add Rule

Apply Rule Changes

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

Launching an Amazon EC2 instance (cont.)

10. Review settings and click **Launch** to launch the instance
11. Close the confirmation page to return to EC2 console
12. Click **Instances** in the navigation pane to view the status of the instance. The status is **pending** while the instance is launching

After the instance is launched, its status changes to **running**

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	pending	

	Name	Instance	AMI ID	Root Device	Type	State	Public DNS
<input type="checkbox"/>	GSG Tutorial	i-e1ab569a	ami-aecd60c7	ebs	t1.micro	running	ec2-50-19-54-72.compute-1.amazonaws.com

Connecting to an Amazon EC2 instance

- There are several ways to connect to an EC2 instance once it's launched.
- **Remote Desktop Connection** is the standard way to connect to Windows instances.
- An **SSH client** (standalone or web-based) is used to connect to Linux instances.

Connecting to Linux/UNIX Instances from Linux/UNIX with SSH

Prerequisites:

- Most Linux/UNIX computers include an SSH client by default, if not it can be downloaded from openssh.org
 - Enable SSH traffic on the instance (using security groups)
 - Get the path the private key used when launching the instance
1. In a command line shell, change directory to the path of the private key file
 2. Use the **chmod** command to make sure the private key file isn't publicly viewable

```
chmod 400 My_Keypair.pem
```

Connecting to Linux/UNIX Instances(cont.)

3. Right click on the instance to connect to on the AWS console, and click **Connect**.
4. Click **Connect using a standalone SSH client**.
5. Enter the example command provided in the Amazon EC2 console at the command line shell



Transferring files to Linux/UNIX instances from Linux/UNIX with SCP

Prerequisites:

- Enable SSH traffic on the instance
- Install an SCP client (included by default mostly)
- Get the ID of the Amazon EC2 instance, public DNS of the instance, and the path to the private key

If the key file is `My_Keypair.pem`, the file to transfer is `samplefile.txt`, and the instance's DNS name is `ec2-184-72-204-112.compute-1.amazonaws.com`, the command below copies the file to the `ec2-user` home

```
scp -i My_Keypair.pem samplefile.txt ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~
```

Terminating Instances

- If the instance launched is not in the free usage tier, as soon as the instance starts to boot, the user is billed for each hour the instance keeps running.
- A terminated instance cannot be restarted.
- To terminate an instance:
 1. Open the Amazon EC2 console
 2. In the navigation pane, click **Instances**
 3. Right-click the instance, then click **Terminate**
 4. Click **Yes, Terminate** when prompted for confirmation

Homework

1. Simple EC2 with Python

1. Create an EC2 with Python and use Python to print out “Hello World”. An example about how to do it, you can check the following youtube:
 - <https://www.youtube.com/watch?v=M2Wc8JIS-p8>

2. Word counting

1. Use the EC2 instance created on (1), and develop a Python to count word frequencies.
Use: <http://programminghistorian.org/lessons/counting-frequencies>

3. Bootstrap

1. Create another EC2 instance
2. Create a bash or a shell script to bootstrap the EC2 instance to automatically install apache web server, PHP interpreter, and a sample PHP application. This would allow you to scale using dynamic configuration. Please refer to online document about how to run commands on your Unix instance:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>
3. Create an instance store-backed Unix AMI based on (2)
4. Instanced another EC2 using the AMI you created above