Contents

Acronyms & Abbreviations

AMI	Amazon Machine Image
EBS	Elastic Block Store (Amazon EBS)
EC2 Instances	Amazon Elastic Compute Cloud (EC2 Instances)
ECS Amazon ECS	Amazon Elastic Container Service
EFS	Amazon Elastic File System
EKS Amazon	EKS Amazon Elastic Container Service for Kubernetes
ELB	Elastic Load Balancer
IGW	Internet Gateway
RDS	(Amazon) Relational Database Service (RDS)
S3	Amazon Simple Storage Services
VGW	Virtual private Gateway
VPC	Virtual Private Cloud

Cloud Computing

- Cloud computing:
    - compute power
    - database storage
    - analytics
    - networking
    - mobile, developer tools, management tools, Internet of Things (IoT), security, and enterprise applications
    - and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing

Regions

- Region is a collection of availability *zones*. To decide in which region:
    - Latency (where are your customers located?) -> in order to increase **speed** (when speed is more, latency is less)
    - Cost
    - Compliance (legal issues)
    - Availability (service availability)

A region is made upon multiple **availability zones.** Applications in AWS run at least to 2 availability zones so that if there are connectivity issues to one, all the data can be maintained and secured to the other (effective and scalable).

Therefore: Each AWS Region has multiple, isolated locations that are known as Availability Zones.

Amazon Relational Database Service (Amazon RDS) provides the ability to place resources (such as instances) and data in multiple locations. Resources aren't replicated across AWS Regions unless this is done specifically.

Compute Services on AWS

How different services work together to build an application.

Architecture

Traffic through internet → (hit) → Elastic Load Balancer (ELB) → (distribute traffic across) → Amazon Elastic Compute Cloud (EC2 Instances) and Amazon Simple Storage Services (S3) (e.g. storing images of application)

- EC2 uses databases:
    - DynamoDB, or
    - Amazon Relational Database Service (RDS)

Compute Capacity

1) Traditional way:

You would first need to estimate how much compute capacity you're going to need, buy the necessary hardware to support that capacity, and stand up the computers or servers to run your application on. Once you deploy your application to those servers, you must maintain that server from the physical maintenance perspective, as well as from a software perspective.

2) Cloud-native applications:

Using a compute as a service model, which allows provisioning and consuming raw compute or server capacity over the internet with pay-as-you-go pricing.

This would *take away the burden of standing up and maintaining physical servers*, while still allowing to have the *control* over what type of hardware you need to run and the software that runs on top of it. In addition to building out and maintaining computer infrastructure in a traditional on-premise environment, it solves problems with reference to the difficulty for accurate initial *estimate for resources capacity*.

If you underprovision resources, your users will feel the effects of a slow application or service. Latency leads to user dissatisfaction, which could impact your business. To remedy this, you need to go out, purchase more servers, and follow the same process of installing, setting up, and maintaining those physical servers. On the flip side, if you overprovisioned, you're going to be paying for those idle resources, and driving up costs unnecessarily.

AWS compute services can easily eliminate the pain of underprovisioning or overprovisioning resources by offering flexible, scalable, and configurable compute resources in the cloud to meet specific needs. AWS also offers managed compute options, like Amazon Lightsail, that allow you to use compute capacity without worrying about provisioning or managing the underlying hardware. In addition, there are options that go beyond raw server capacity. AWS offers container services that allow the user to use Docker through Elastic Container Service, or ECS, or Kubernetes through EKS. Serverless solutions can also be offered, like AWS Lambda, and any application can be virtually run in the cloud.

Useful Links:

- full range of AWS compute services
- AWS Lamda: AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume- there is no charge when your code isn't running.
- Amazon Container Services
    - Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers. It allows you to run and scale containerized applications on AWS.

- Amazon Elastic Container Service for Kubernetes (Amazon EKS) makes it straightforward to deploy, manage, and scale containerized applications that use Kubernetes on AWS.
- AWS **Fargate** is a compute engine for Amazon ECS and Amazon EKS that allows you to run containers without having to manage servers or clusters.

➢ Eligibility for the AWS Free Tier

Amazon Elastic Compute Cloud (EC2 – EC2 Instances)

- Each **virtual server** you provision is called EC2 instance.
- EC2 instances can be configured based on customized needs.
- You choose an **Amazon Machine Image (AMI)**
  - It contains information about how you want your instance to be configured (e.g. **operating system** (linux/windows), desired **applications** to be installed on that instance)
  - A single AMI can have one or more EC2 Instances, which all would share the same **configuration**
  - You can also configure the **instance type & size,** i.e. amount of compute (*compute optimized*), memory (*memory optimized*), storage (*storage optimized*), networking capabilities available per instance
    - There are **instance types** which lead to different types of **hardware** based on the situation
    - In the short term, you can experiment on different servers until you find the optimal configurations for your applications for the long term period

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure and resizable compute capacity in the cloud. It's designed to make web-scale cloud computing easier for developers.

Amazon EC2 presents a true **virtual computing environment**, and it allows you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network's access permissions, and run your image by using as many or few systems as you want.

Details on the features and cost of Amazon EC2: https://aws.amazon.com/ec2/

Amazon EC2 instance types

Amazon EC2 provides a wide selection of instance types that are optimized to fit different **use cases**. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity. They give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, which allows you to scale your resources to the requirements of your target workload. Details: https://aws.amazon.com/ec2/instance-types/

Amazon Lightsail

Amazon Lightsail is the easiest way to get started with AWS for developers, small businesses, students, and other users who need a simple virtual private server (VPS) solution. Lightsail provides developers compute, storage, and networking capacity, and it also provides capabilities to deploy and manage websites and web applications in the cloud. Lightsail includes everything you need to launch your project quickly--a virtual machine, solid state drive (SSD)-based storage, data transfer, Domain Name System (DNS) management, and a static IP, for a low, predictable, and monthly price.

A more detailed introduction from AWS re:Invent 2017 is available here: https://www.youtube.com/watch?v=29_LqYnomdg. Note that pricing has changed (decreased) since this video was created. Specific details are on the Lightsail web page.

- Lightsail: https://aws.amazon.com/lightsail/, Lightsail pricing: https://aws.amazon.com/lightsail/pricing

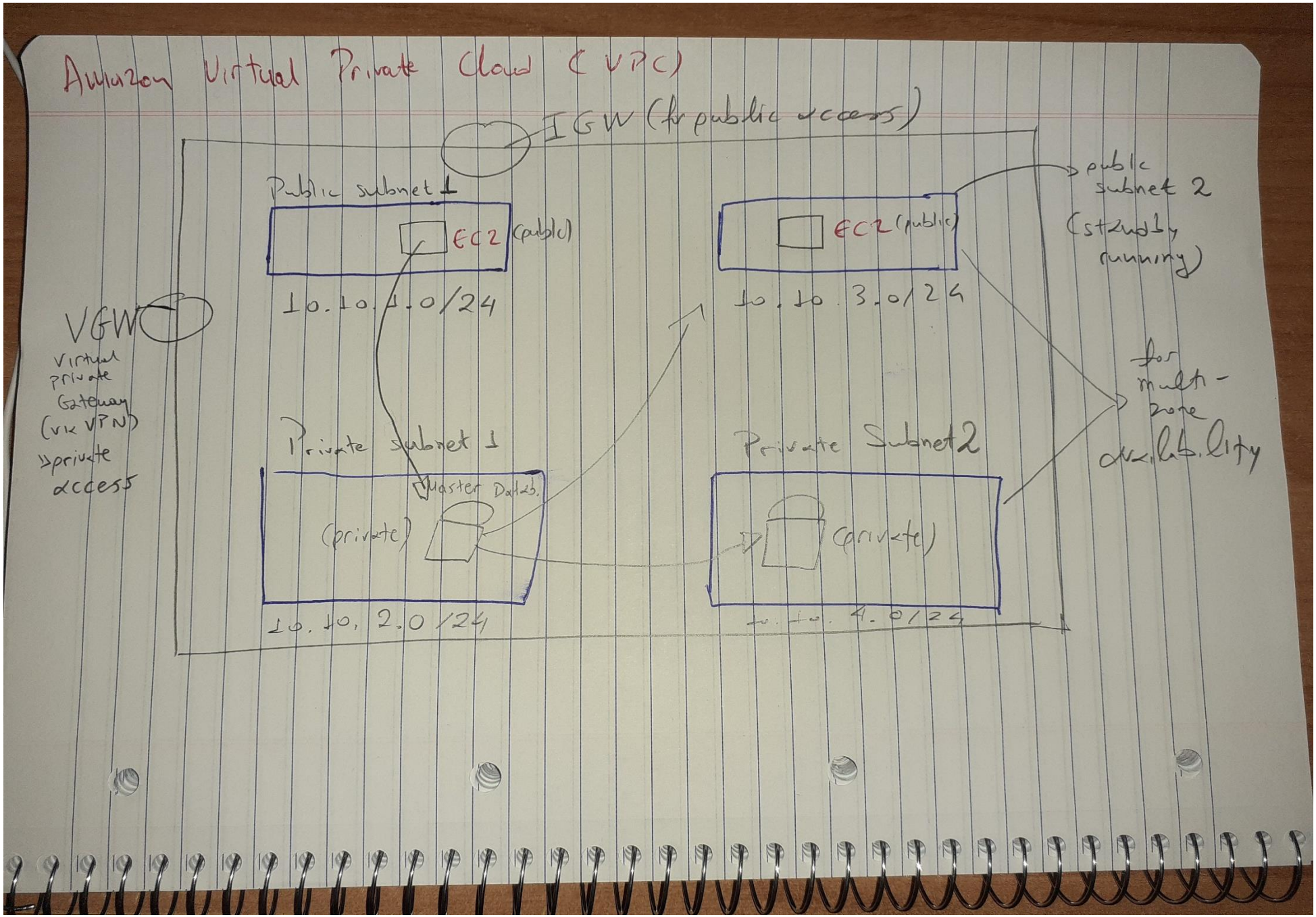
Networking & Storage

Amazon Virtual Private Cloud (VPC)

Providing a frame in which all applications on website are inside the VPC.

We have to select:

- Region
- IP range


Amazon Virtual Private Cloud (VPC)
IGW (for public access)
Public subnet 1
EC2 (public)
10.10.1.0/24
public subnet 2
(stay subtly running)
EC2 (public)
10.10.3.0/24
VGW
virtual private Gateway (via VPN)
»private access
for multi-zone availability
Private subnet 1
Master Datab.
(private)
10.10.2.0/24
Private Subnet 2
(private)
10.10.4.0/24

Subnets

Subnet is a subset of the IP ranges for the VPC which hosts the EC2 instances (public subnet)

- **Private subnet:** special category of subnet, which hosts the database-data plane (will be the **master database** if it is our 1st private subnet), which we do not want to be directly interacted with users coming in the website, i.e. IGW does not have access in the private subnet.

We can create multiple public subnets -for expanding our availability zone- which will have the same VPC

- More public subnets will again have to be associated with their private subnet, as before
- Again, the public subnet will host an EC2 Instance, a new one (that is going to have a standby running in this other availability zone)
- Again, the private subnet will host the database, but this time will be a Multi-AZ RDS (let's suppose we had RDS in the 1st subnet master database) to support multi-zone availability

Internet Gateway : IGW

- So that others (customers) can interact within that framework = VPC and our subnet. Otherwise, without a IGW, out VPC and subnet would be fully isolated (not able to allow public traffic)

Virtual private Gateway VGW

- For creating private access, contrary to the IGW which gives public access

Elastic Load Balancer (ELB)

When you have more than 1 public subnet (and therefore more than one range of Ips for customers), you need an elastic load balancer (ELB) associate with the EC2 Instances (hosted in the 2 public subnets) so that it doesn't matter which one gets the customer traffic: they will both get an even amount of traffic distributed automatically by the ELB.

Amazon Networking Notes

CIDR Notation

An important concept that's used in networking on AWS is CIDR, or *Classless Inter-Domain Routing*. CIDR network addresses are allocated in a virtual private cloud (VPC) and in a subnet by using CIDR notation. A /16 block provides 65,536 IPv4 addresses. A /24 block provides 256 addresses. CIDR: https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing

Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon 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. You have complete control over your virtual networking environment, including the selection of your own IP address range, the creation of subnets, and the configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications. You could create up to five non-default VPCs per AWS account per Region. (See below for information about default VPCs.)

Details on Amazon VPC can be found here: https://aws.amazon.com/vpc

Subnets

A VPC spans all the Availability Zones in the Region. After creating a VPC, you can add one or more subnets in each Availability Zone. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block. Each subnet must reside entirely within one Availability Zone, and it can't span Availability Zones.

More info: https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html

Security in a VPC is provided by using Security Groups and Network Access Control Groups.

Default VPC

In each Region, AWS will provision a default VPC. The route table that is associated with the default VPC will have a public route, which in turn is associated with a provisioned internet gateway.

You can modify or delete the default VPC if you want to do so.

The most current details on the default VPC can be found here: https://docs.aws.amazon.com/vpc/latest/userguide/default-vpc.html

Storage

S**tatic** data-files, like images, are stored in S3 **(object-level storage)**, but **dynamic** data in RDS **(block-level storage)**. This happens because if you want to replace an image you have to delete the previous one, while if you want to update a contact form, e.g. changing the location info while maintaining the other info as it was, then you do not need to overwrite that form/platform.

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated inside an Availability Zone to protect you from component failure, which offers high availability and durability. Amazon EBS volumes offer consistency and low-latency performance for workloads.

Amazon EBS options:

1. SSD-backed storage for transactional workloads, such as databases and boot volumes (performance depends primarily on IOPS)
2. Hard disk drive (HDD)-backed storage for throughput-intensive workloads, such as MapReduce and log processing (performance depends primarily on MB/s).

The Elastic Volume feature of Amazon EBS allows you to dynamically increase capacity, tune performance, and change the type of live volumes with no downtime or performance impact. This allows you to easily right-size your deployment and adapt to performance changes.

- Pricing for Amazon EBS is based on the amount (volume) and type of Amazon EBS volume that you provision.
    - Info: https://aws.amazon.com/ebs/pricing/. Confirm that you are looking at cost in the correct Region.
- Details on Amazon EBS: https://aws.amazon.com/ebs

Amazon S3

- **Objects are stored in the so-called buckets**
    - Buckets= repositories
    - http://mybucketname.s3.amazon.com/*object_name*

Amazon Simple Storage Service (Amazon S3) stores data as **objects** within resources that are called ***buckets***. You can store as many objects as you want within a bucket, and you can write, read, and delete objects in your bucket. Objects can be up to 5 TB in size.

You can control access to both the bucket and the objects (who can create, delete, and retrieve objects in the bucket for example), and view access logs for the bucket and its objects. You can also choose the AWS Region where a bucket is stored to optimize for latency, minimize costs, or address regulatory requirements.

- Price: AWS Simple Monthly Calculator. Also based on the location of your Amazon S3 bucket.
    - https://aws.amazon.com/s3.

Elastic File System (EFS)

Amazon Elastic File System (Amazon **EFS**) provides simple, scalable, elastic file storage for use with AWS Cloud services and on-premises resources. It is straightforward to use, and it offers a simple interface that allows you to create and configure file systems quickly and easily.

It provides massively parallel shared **access to thousands of Amazon EC2 instances**. This enables your applications to achieve high levels of aggregate throughput and IOPS (Input/output operations per second) that scale as a file system grows, with consistent low latencies.

When an Amazon EFS file system is mounted on Amazon EC2 instances, it provides a **standard** file **system interface** and **file system access semantics**, which allows you to seamlessly integrate Amazon EFS with your existing applications and tools. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, thus allowing Amazon EFS to provide a common data source for workloads and applications that run on more than one Amazon EC2 instance.

Details: https://aws.amazon.com/efs/