Networking: Background

In this section we will discuss the fundamentals of networking, how AWS global network is organized and the components it is comprised of. A concise list of the topics is given below:

- Brief History of The Internet
- OSI Model
- IP Addresses and DNS, what they are and how they work
- AWS Global infrastructure, Availability Zones and Local Zones
- Caching and CDN, what they are and how they work

If the reader believes themselves to already be familiar with the listed technologies, then they can feel free to skip this section and move on to the rest of the chapter.

Brief History of The Internet

We would not be talking about cloud computing if it were not for the internet, this monumental invention of humans that allows a diverse range of computers quite geographically dispersed to communicate with each other. It is after all, the chief way in which most businesses relying on AWS access their rented resources and the chief way in which your customers will access AWS resources if you're running a public-facing operation. As such, I find it necessary to provide a brief history (albeit a very diluted one) on how the internet was created.

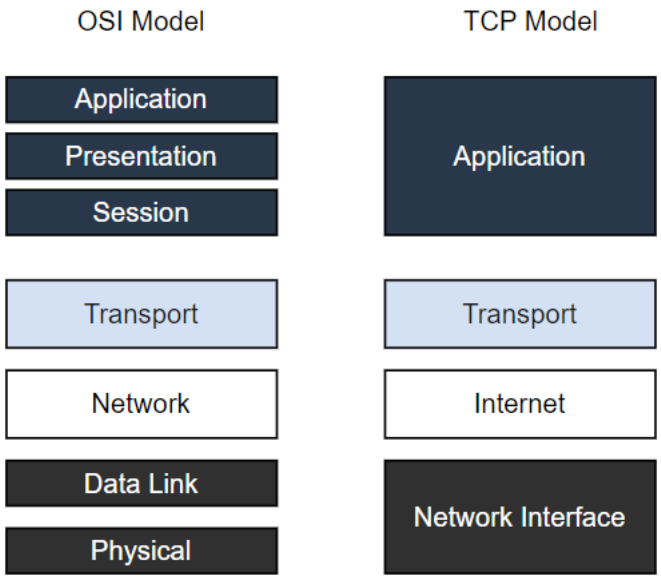
Before the advent of the internet, the largest networks of connected computers were all private networks, they were networks owned either by government agencies, university campuses and large corporate houses which were all using separate pieces of technology to organize and operate their networks. And since these networks did not utilize standardized ways of operating, communicating across networks was very difficult. A computer in a network owned by a university

campus therefore, could not talk and share information with a computer owned by a large corporate house, or even another university.

The United States government has an agency called the Defense Advanced Research Projects Agency (DARPA), which is responsible for performing R&D operations for the American military. In 1969, DARPA started a project that would later become known as ARPANET, a project to unify various private networks and allow their constituent computers to talk with one another. The success of ARPANET led to others trying the same and in 1973, a proposal to connect these disparately connected networks into a single, large network through an **internetworking protocol** was put forward: The internet was born.

OSI and TCP models of the internet

The internet today has evolved works through not just one but through many internetworking protocols layered on top of one another. There are two major models of these interlayered protocols taught in universities, **The OSI Model** and **The TCP Model**.



Now, every layer in that model has its own purpose in the sending and receiving of data across the internet, and every layer has its own set of protocols and rules for doing so. Now, of the two models, The OSI model is mostly an academic model, used to teach students about networking, while the TCP model is the one actually used in the real world, the one that reflects what businesses are using as I write this very sentence.

As such, the TCP model, short for the Transmission Control Protocol model, is the one that this section will elaborate on and explain, but the reader may feel free to look into the OSI model if they find the topic to be interesting. The four layers of the TCP model are as follows:

Layer 1, Network Interface: The network interface layer, also known as the data link layer, handles the physical infrastructure that lets computers communicate with one another over the internet. This covers ethernet cables, wireless networks, network interface cards, and so on. The network interface layer also includes the technical infrastructure, such as the code that converts digital data into transmittable signals, that makes network connection possible. Some major layer 1 protocols are: **Ethernet**, **Wifi** and **ARP**.

Layer 2, Internet: The internet layer, also known as the network layer, controls the flow and routing of traffic to ensure data is sent speedily and accurately. This layer is also responsible for reassembling the data packet at its destination. If there's lots of internet traffic, the internet layer may take a little longer to send a file, but there will be a smaller chance of an error corrupting that file. **Internet Protocol (IP)** is for example, a layer 2 protocol.

Layer 3, Transport Layer: The transport layer provides a reliable data connection between two communicating devices. It's like sending an insured package: the transport layer divides the data in packets, acknowledges the packets it has received from the sender, and ensures that the recipient acknowledges the packets it receives. **TCP** and **UDP** are two common layer 3 protocols.

Layer 4, Application Layer: The application layer is the group of applications that let the user access the network. For most of us, that means email, messaging apps, and cloud storage programs. This is what the end-user sees and interacts with when sending and receiving data. **HTTPS** and **FTP** are two very common layer 4 protocols.

IP Addresses and DNS

Now, IP addresses are like the ID assigned to every computer that is part of the internet, it has two types, IPv4 and IPv6 and looks something like this: `192.168.1.1` (IPv4 addresses at least). IP addresses also help us locate a device geographically when we are either sending or receiving packets to them, and every website, API or service on the internet that the reader has ever used is in part powered by them. These numerical addresses however, are quite hard to remember and asking people to remember them every time they wish to visit a website might prove difficult, enter DNS.

A **Domain Name** is a unique name that identifies a website on the internet, imagine nytimes.com or google.com. We map these domain names to IP Addresses like the ones above, allowing us to navigate to the devices possessing the address without having to remember their complex numerical designation.

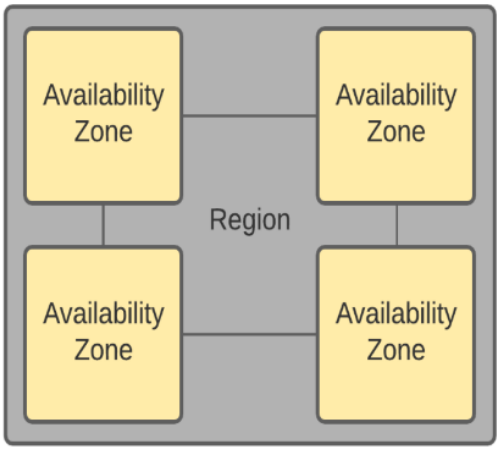
DNS (Domain Name System) then is like the phonebook of the internet, its a giant registry of domain names and the IP addresses that they point to. Whenever we go to a webpage, say Amazon.com, our browser consults the DNS to find out which IP Address it must navigate and send requests to.

AWS Global Infrastructure

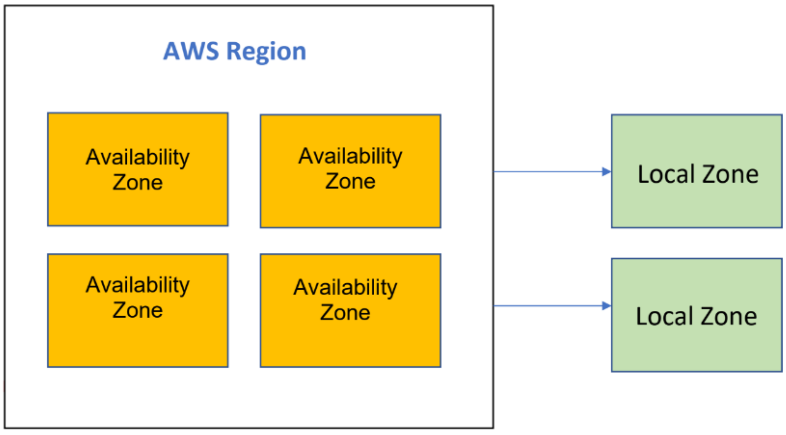
We've mentioned before how AWS operates infrastructure in over 245 countries, with presence in every one of the seven continents. This infrastructure is organized into **Regions**, **Availability Zones**, **Local Zones** and **Edge Locations**.

Regions in AWS represent data centers clustered around a large geographic area i.e. a continent or a part of a continent. Example, us-east-1 represents a group of

data centers clustered around the eastern coast of the United States. An Availability Zone (AZ) on the other hand, is simply one of the data centers within it. This is succinctly illustrated in the diagram below:

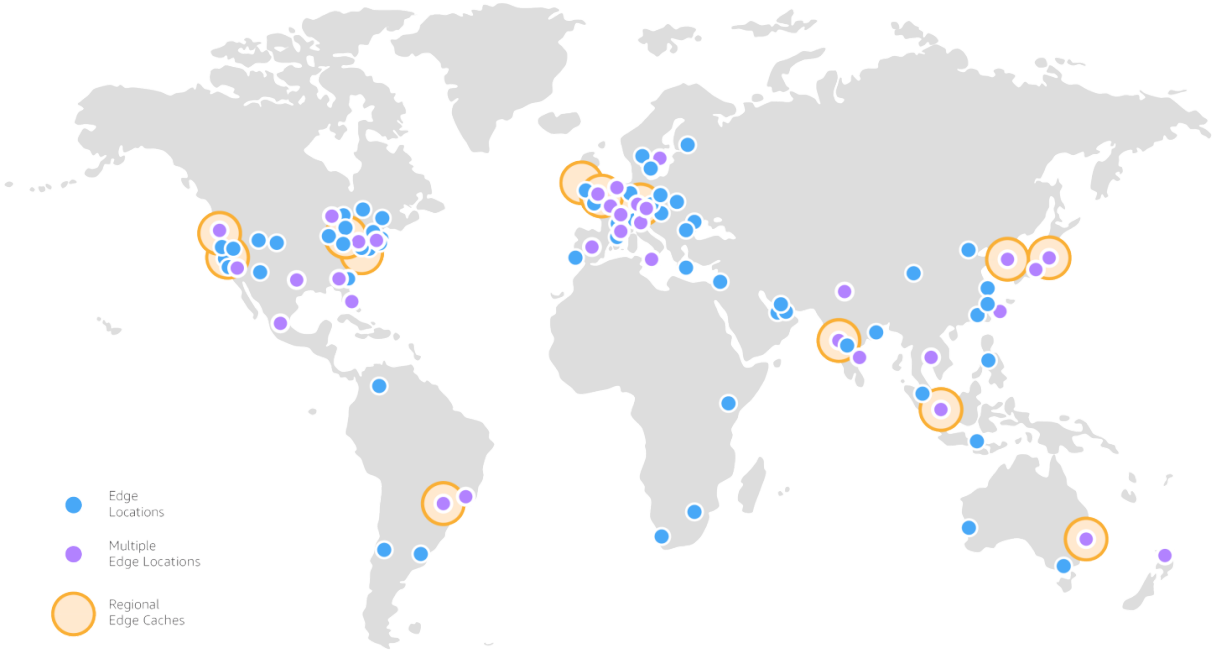


There may however, be times where a data center is needed outside the boundaries of these regions, due to a large cluster of users being concentrated in areas at a significant distance from the region, which is usually the case for large cities. In order to address such situations, AWS created the Local Zone. The local zone is effectively, a data center closer to the user. Organizations place systems that require low latency in these local zones. A diagram of a local zone is given below:



Finally, Edge Locations are the small pieces of infrastructure used to power CloudFront, Amazon's Content Delivery Network. They mostly serve the purpose of improving internet performance and website security. CDNs and CloudFront are both discussed liberally throughout the chapter.

At the time of writing, AWS infrastructure spans across 34 Geographical regions, 108 Availability Zones, 34 Local Zones and 600+ Edge locations. A map of the regions and edge locations is given below:



Caching and CDN

Caching is nothing but the process of storing frequently accessed data in a location that can be retrieved quickly, minimizing the time and resources it takes to serve that data to a user. Its like putting the car keys in a drawer right by the door.

We use the keys frequently, and it saves time to have them in a convenient, easy-to-reach spot rather than searching the entire house for them each time we need to go out. In web terms, this means data that users often request (like images,

website code, or video content) is temporarily stored closer to them, resulting in less loading times.

Now, if we were to create a full-blown distributed network of these caches, and connect them both to AZs and to each other, what we would get is a **Content Delivery Network (CDN)**, a system designed to speed up the delivery of critical and/or frequently requested data to the user.

Application Programming Interfaces (APIs)

An unplanned addition to this section, APIs (Application Programming Interfaces) are sets of rules that allow different software systems to communicate. They are genuinely the cornerstone of many internet applications, and dedicating a paragraph to the concept in this section was thus believed to be necessary.

APIs, in simple words, enable one system to request or send data to another without needing to know how the other system works internally. They handle the creation of a standard channel for communication between two systems on our behalf, allowing two systems to talk with each other in extremely convenient ways.

AWS has its own set of APIs, with the python Boto3 API in particular being a popular tool in many cloud engineer’s arsenals. Also many of the individual services like S3 and EC2 also have their own APIs, allowing developers to more easily automate tasks and build integrations.