AIDS – Cloud Computing Notes

What is cloud computing?

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

Who is using cloud computing?

Organizations of every type, size, and industry are using the cloud for a wide variety of use cases, such as data backup, disaster recovery, email, virtual desktops, software development and testing, big data analytics, and customer-facing web applications. For example, healthcare companies are using the cloud to develop more personalized treatments for patients. Financial services companies are using the cloud to power real-time fraud detection and prevention. And video game makers are using the cloud to deliver online games to millions of players around the world.

Benefits of cloud computing

Agility

The cloud gives you easy access to a broad range of technologies so that you can innovate faster and build nearly anything that you can imagine. You can quickly spin up resources as you need them–from infrastructure services, such as compute, storage, and databases, to Internet of Things, machine learning, data lakes and analytics, and much more.

You can deploy technology services in a matter of minutes, and get from idea to implementation several orders of magnitude faster than before. This gives you the freedom to experiment, test new ideas to differentiate customer experiences, and transform your business.

Elasticity

With cloud computing, you don't have to over-provision resources up front to handle peak levels of business activity in the future. Instead, you provision the amount of resources that you actually need. You can scale these resources up or down to instantly grow and shrink capacity as your business needs change.

Cost savings

The cloud allows you to trade fixed expenses (such as data centers and physical servers) for variable expenses, and only pay for IT as you consume it. Plus, the variable expenses are much lower than what you would pay to do it yourself because of the economies of scale.

Deploy globally in minutes

With the cloud, you can expand to new geographic regions and deploy globally in minutes. For example, AWS has infrastructure all over the world, so you can deploy your application in multiple physical locations with just a few clicks. Putting applications in closer proximity to end users reduces latency and improves their experience.

Cloud Delivery Models

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) contains the basic building blocks for cloud IT and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space. Infrastructure as a Service vendors can help you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.

Platform as a Service (PaaS)

Platforms as a service (PaaS) vendors remove the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and this integration allows you to focus on the deployment and management of your applications. This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

Software as a Service (SaaS)

Software as a Service (SaaS) vendors provide you with software applications that is run and managed by the vendor. In most cases, people referring to Software as a Service are referring to third-party end-user applications. With a SaaS offering you do not have to think about how the service is maintained or how the underlying infrastructure is managed; you only need to think about how you will use that particular piece of software. A common example of a SaaS application is web-based email where you can send and receive email without having to manage feature additions to the email product or maintaining the servers and operating systems that the email program is running on.

Bare Metal Servers vs. Hypervisors

Bare Metal Servers: A physical server

The term "bare metal" refers to a physical server with tangible components such as RAM, CPU, network cards, and more. In this configuration, the operating system communicates directly with these hardware elements, optimizing performance and resource utilization.

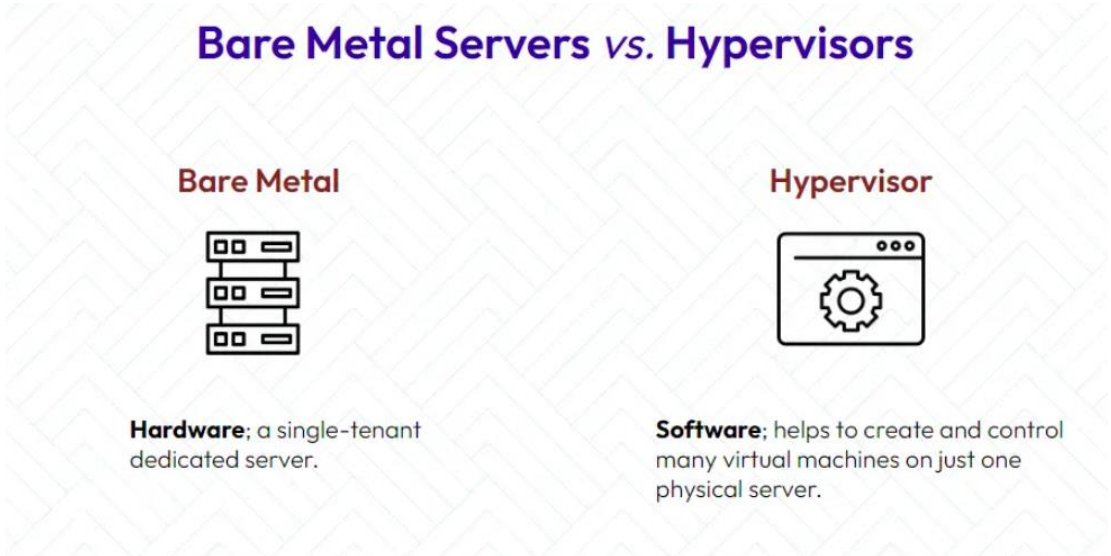
Take a look at our bare metal servers to understand that these are physical servers with tangible components such as RAM, hard drives, and Intel processors/cores.

Performance

Bare metal servers, stripped of a virtualization layer, excel in raw performance. With direct access to components like RAM and CPU, they are ideal for high-performance applications such as databases and real-time analytics.

Security

In security-sensitive environments, bare metal servers are preferred due to the absence of an intermediary layer between the hardware components and the operating system, thereby reducing the attack surface.



Hypervisors: A software layer

On the other hand, hypervisors aren't physical things; they are software layers that create a separation between hardware components and the operating system. This separation makes it possible to create and manage multiple virtual machines, each running its own operating system.

Flexibility and Isolation

Hypervisors provide flexibility by allowing the simultaneous operation of multiple virtual machines with different operating systems. This is valuable for testing environments and scenarios requiring isolation between workloads.

Resource Sharing

Unlike bare metal servers, hypervisors enable resource sharing among multiple virtual machines, offering a cost-effective solution where resource efficiency is prioritized over raw performance.

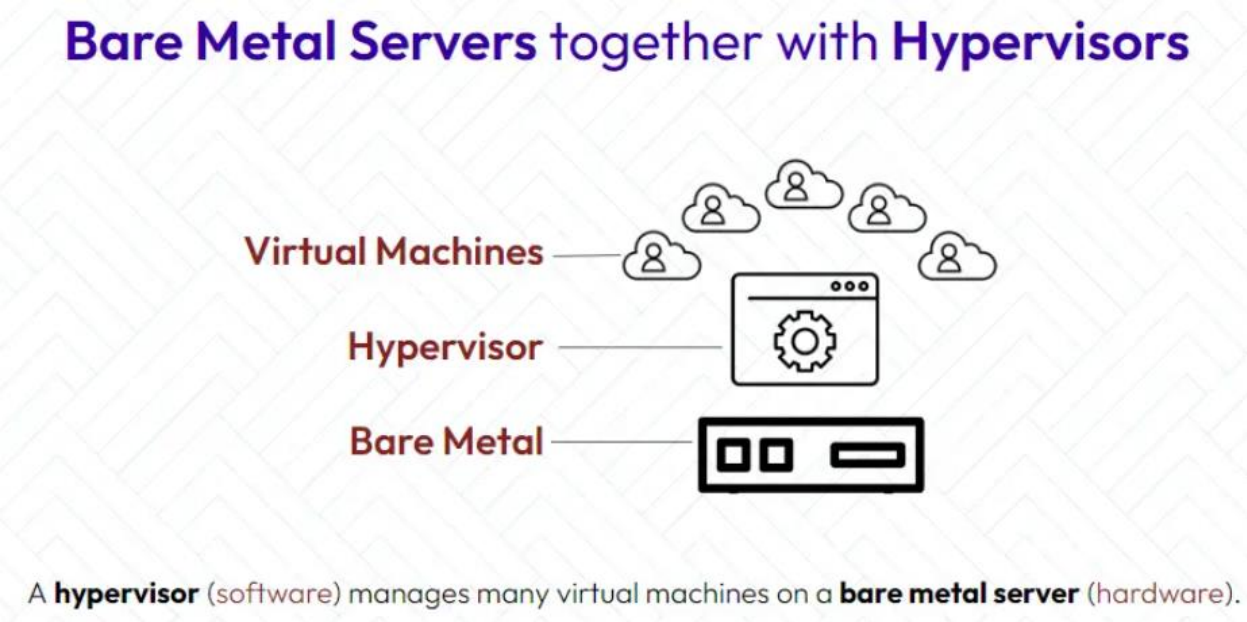
Ease of Management

Hypervisors offer a user-friendly interface for managing virtual machines, making them suitable for development, testing, and environments requiring rapid deployment and scalability.

To learn more about hypervisors, check out our what is a hypervisor? article.

How do Bare Metal Servers and Hypervisors work together?

A bare metal server, essentially a powerful physical machine, becomes more flexible when a hypervisor is added. The hypervisor acts like a manager, allowing the bare metal server to host multiple virtual machines. These virtual machines operate independently, each running its own operating system and applications. The beauty of this collaboration is that it combines the raw performance of the bare metal server with the efficiency and adaptability of virtualization.



On Premises VS On Cloud.

On Premises: In on-premises, from use to the running of the course of action, everything is done inside; whereby backup, privacy, and updates moreover should be managed in-house. At the point when the item is gotten, it is then installed on your servers; requiring additional power laborers, database programming software and operating systems to be purchased. With no prior commitment, you anticipate complete ownership.

On Cloud : Cloud refers to the delivery of on-demand computing services over the internet on "Pay As U Use "services, in simple words rather than managing files and Services on the local storage device you can do the same over the Internet in a cost-efficient manner. With a Cloud-based enrolment model, there is no convincing motivation to purchase any additional establishment or licenses.

Difference between On-Premises and On Cloud :

Scalability –

When it comes to scalability we pay more for on-premises set up and get lesser option too and once you scale up it is difficult to scale down and turn into heavy loss like infrastructure and maintenance cost while on the other hand Cloud allows you to pay only how much you use with much easier and faster for scaling upper and down.

Server Storage –

On-premises need a lot of space, power, and maintenance to store while on the other hand cloud solution are offered by the provider and maintain the server which saves your money and space.

Data Security –

On promises offers less security and for security, we need physical and traditional IT security measures whereas the cloud offers much better security, and I avoiding all other physical and other security options.

Data Loss or Recovery –

If data loss occurs recovery in on-premises is very least while cloud offers you the backup for easier and faster data recovery.

Maintenance –

On promises require an extra team for maintenance which increases the cost while the cloud is maintained by the provider.

What is DNS?

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

How does DNS work?

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it's important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs "behind the scenes" and requires no interaction from the user's computer apart from the initial request.

There are 4 DNS servers involved in loading a webpage:

DNS recursor - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the

recursor is then responsible for making additional requests in order to satisfy the client's DNS query.

Root nameserver - The root server is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.

TLD nameserver - The top level domain server (TLD) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is "com").

Authoritative nameserver - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

What's the difference between an authoritative DNS server and a recursive DNS resolver?

Both concepts refer to servers (groups of servers) that are integral to the DNS infrastructure, but each performs a different role and lives in different locations inside the pipeline of a DNS query. One way to think about the difference is the recursive resolver is at the beginning of the DNS query and the authoritative nameserver is at the end.

Recursive DNS resolver

The recursive resolver is the computer that responds to a recursive request from a client and takes the time to track down the DNS record. It does this by making a series of requests until it reaches the authoritative DNS nameserver for the requested record (or times out or returns an error if no record is found). Luckily, recursive DNS resolvers do not always need to make multiple requests in order to track down the records needed to respond to a client; caching is a data persistence process that helps short-circuit the necessary requests by serving the requested resource record earlier in the DNS lookup.

Authoritative DNS server

Put simply, an authoritative DNS server is a server that actually holds, and is responsible for, DNS resource records. This is the server at the bottom of the DNS lookup chain that will respond with the queried resource record, ultimately allowing the web browser making the request to reach the IP address needed to access a website or other web resources. An authoritative nameserver can satisfy queries from its own data without needing to query another source, as it is the final source of truth for certain DNS records.

What is an AWS Region?

An AWS Region is a cluster of data centers in a specific geographic area, such as the Northeastern United States or Western Europe. It is best practice to choose a region that is geographically close to users; this reduces latency because data reaches the users more quickly.

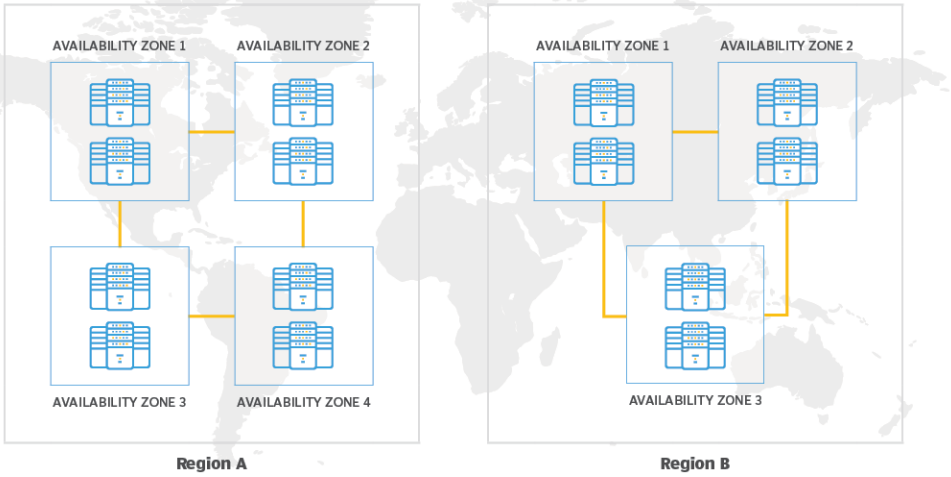
Each AWS Region includes multiple AZs. However, each AZ is restricted to a specific AWS Region. You can use multiple AZs within one Region, but you can't use the same AZ across multiple Regions.

What is an AWS Availability Zone?

An AZ is a standalone data center or set of data centers within a Region. Each AZ operates independently, so a failure in one won't affect others. In disaster recovery plans, enterprises use multiple AZs to increase redundancy and reliability.

AZs shouldn't be confused with AWS Local Zones, which are extensions of a Region. Local Zones let you choose more specific geographic locations, such as Boston or Los Angeles. They are not designed to increase workload redundancy. They are valuable if your users are concentrated in a relatively small area, as they help reduce latency and meet strict compliance requirements.

Availability zones vs. regions



SOURCE: PAUL KIRVAN; DATA CENTERS: NEXUSBYVADOBE STOCK

What is Amazon EC2 (Elastic Compute Cloud)?

Amazon Web service offers EC2 which is a short form of Elastic Compute Cloud (ECC) it is a cloud computing service offered by the Cloud Service Provider AWS. You can deploy your applications in EC2 servers without any worrying about the underlying infrastructure. You configure the EC2-Instance in a very secure manner by using the VPC, Subnets, and Security groups. You can scale the configuration of the EC2 instance you have configured based on the demand of the application by attaching the autoscaling group to the EC2 instance. You can scale up and scale down the instance based on the incoming traffic of the application.

Use Cases of Amazon EC2 (Elastic Compute Cloud)

The following are the use cases of Amazon EC2:

Deploying Application: In the AWS EC2 instance, you can deploy your application without maintaining the underlying infrastructure.

Scaling Application: Once you deployed your web application in the EC2 instance know you can scale your application based upon the demand you are having by scaling the AWS EC2-Instance.

Hybrid Cloud Environment: You can deploy your web application in EC2-Instance and you can connect to the database which is deployed in the on-premises servers.

Cost-Effective: Amazon EC2-instance is cost-effective so you can deploy your gaming application in the Amazon EC2-Instances

AWS EC2 Instance Types

Different Amazon EC2 instance types are designed for certain activities. Consider the unique requirements of your workloads and applications when choosing an instance type. This might include needs for computing, memory, or storage.

The AWS EC2 Instance types are as follows:

1. General Purpose Instances

2. Compute Optimized Instances
3. Memory-Optimized Instances
4. Storage Optimized Instances
5. **Accelerated Computing Instances**

What Is Amazon Route 53?

Route 53 is a web service that is a highly available and scalable Domain Name System (DNS.)

Let's understand what Amazon Route 53 in technical terms is. AWS Route 53 lets developers and organizations route end users to their web applications in a very reliable and cost-effective manner. It is a Domain Name System (DNS) that translates domain names into IP addresses to direct traffic to your website. In simple terms, it converts World Wide Web addresses like www.example.com to IP addresses like 10.20.30.40.

Basically, domain queries are automatically routed to the nearest DNS server to provide the quickest response possible. If you use a web hosting company like GoDaddy, it takes 30 minutes to 24 hours to remap a domain to a different IP, but by using Route 53 in AWS it takes only a few minutes.

How does Amazon Route 53 work?

AWS Route 53 connects requests to the infrastructure running in AWS also used to route users to infrastructure outside of AWS.

AWS Route 53 can be easily used to configure DNS health checks, continuously monitor your applications' ability to recover from failures, and control application recovery with Route 53 Application Recovery Controller. Further, AWS Route 53 traffic flow helps to manage traffic globally via a wide variety of routing types including latency-based routing, geo DNS, weighted round-robin, and geo proximity. All these routing types can be easily combined with DNS Failover in order to enable a variety of low-latency, fault-tolerant architectures.

Let us understand, step by step, how does AWS Route 53 work:

- A user accesses www.example.com, an address managed by Route 53, which leads to a machine on AWS.
- The request for www.example.com is routed to the user's DNS resolver, typically managed by the ISP or local network, and is forwarded to a DNS root server.

- The DNS resolver forwards the request to the TLD name servers for ".com" domains.
- The resolver obtains the authoritative name server for the domain—these will be four Amazon Route 53 name servers that host the domain's DNS zone.
- The DNS resolver chooses one of the four Route 53 servers and requests details for the hostname www.example.com.
- The Route 53 name server looks in the DNS zone for www.example.com, gets the IP address and other relevant information, and returns it to the DNS resolver.
- The DNS resolver returns the IP address to the user's web browser. The DNS resolver also caches the IP address locally as specified by the Time to Live (TTL) parameter.
- The browser contacts the webserver or other Amazon-hosted services by using the IP address provided by the resolver.
- The website is displayed on the user's web browser.

Now, look at the benefits provided by Route 53.

Amazon Route 53 Benefits

Route 53 provides the user with several benefits.

They are:

- Highly Available and Reliable
- Flexible
- Simple
- Fast
- Cost-effective
- Designed to Integrate with Other AWS Services
- Secure
- Scalable

AWS Routing Policies

There are several types of routing policies. The below list provides the routing policies which are used by AWS Route 53.

- Simple Routing
- Failover Routing Policy
- Latency-based Routing
- Geolocation Routing
- Geoproximity Routing Policy
- Multivalue Routing Policy
- Weighted Routing Policy

Amazon VPC

Amazon VPC or Amazon Virtual Private Cloud is a service that allows its users to launch their virtual machines in a protected as well as isolated virtual environment defined by them. You have complete control over your VPC, from creation to customization and even deletion. It's applicable to organizations where the data is scattered and needs to be managed well. In other words, VPC enables us to select the virtual address of our private cloud, and we can also define all the sub-constituents of the VPC like subnet, subnet mask, availability zone, etc on our own.

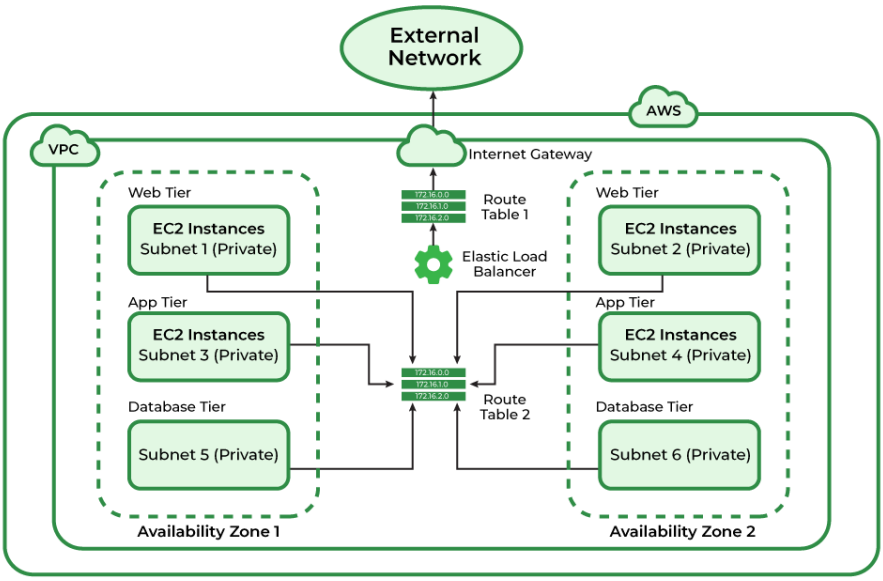
We can place the necessary resources and manage access to those resources in the VPC, a private area of Amazon that we control.

A default "VPC" will be generated when we register an AWS account, allowing us to manage the virtual networking environment, the IP address, the construction of subnets, route tables, and gateways.

Amazon VPC (Virtual Private Cloud) Architecture

The basic architecture of a properly functioning VPC consists of many distinct services such as Gateway, Route table, Subnets, etc. Altogether, these resources are clubbed under a VPC to create an isolated virtual environment. Along with these services, there are also security checks on multiple levels.

It is initially divided into subnets, connected with each other via route tables.



Amazon VPC (Virtual Private Cloud) Components

VPC

You can launch AWS resources into a defined virtual network using Amazon Virtual Private Cloud (Amazon VPC). With the advantages of utilizing the scalable infrastructure of AWS, this virtual network closely mimics a conventional network that you would operate in your own data center. /16 user-defined address space maximum (65,536 addresses)

Subnets

To reduce traffic, the subnet will divide the big network into smaller, connected networks. Up to /16, 200 user-defined subnets.

Route Tables

Route Tables are mainly used to Define the protocol for traffic routing between the subnets.

Network Access Control Lists

Network Access Control Lists (NACL) for VPC serve as a firewall by managing both inbound and outbound rules. There will be a default NACL for each VPC that cannot be deleted.

Internet Gateway(IGW)

The Internet Gateway (IGW) will make it possible to link the resources in the VPC to the Internet.

Network Address Translation (NAT)

Network Address Translation (NAT) will enable the connection between the private subnet and the internet.