What is an Apache Web server?

It is a web server, which helps to serve the web pages through internet requested by the user. It is used for static content delivery. It is free and open-source.To install apache web server in linux-

In Redhat based distros: sudo yum install httpd

In Debian Based distros: sudo apt install apache2

What is the root directory of the Apache webserver?

/var/www/html

What is configuration file of the Apache webserver?

apache2(Redhat Based): /etc/apache2 >>apache2.conf

httpd(Debian based): /etc/httpd/conf >> httpd.conf

------------------------------------------------------------------------------------------------------------------------------------------

Saas(Software As A Service):

It is a Software distribution model in which a third-party service provider hosts the application and makes them available to the customer over the internet. The user only interacts with the application. Examples: Email, Google Drive, Youtube.

Paas(Platform As A Service):

It is a cloud computing model in which cloud vendors provide developers with a platform for building applications. In Paas vendor manages containers, Operating systems, virtualization, servers, infrastructure. It allows you to focus on the deployment and management of the application Examples: Heroku, Google App Engine.

Iaas(Infrastructure As A Service):

It is a cloud computing model that provides virtualized computing resources over the internet. It provides access to networking features, computers, and data storage space. It provides the highest level of flexibility for using cloud resources. Also known as Hardware As A  Service.

Example: AWS, Microsoft Azure, Google App Engine.

Other Services:

Saas(Storage As A Service)

Caas(Communication As A Service)

Maas(Monitoring As A Service)

Naas(Network As A Service)

---------------------------------------------------------------------------------------------------------------------

Monolithic Architecture:

In monolithic architecture whole system resides in one single unit. Usually, it comprises a client-side user interface, a web server, and a database. There is no modularity. All the services are tightly coupled in one single server. If you want to make changes to a particular service then the whole server needs to go down. You cannot scale it easily. If a server goes down all the services inside the server will go down which means single-point failure. Development in monolithic is like you have to update all the services or none that means slow continuous development. Allows only a single technology stack. It's like putting all eggs in a single basket. It is less complex and easy to deploy.

Microservice Architecture:

In this application development is broken down into small services(deployable modules) and these services are loosely coupled. All these services communicate to each other via API. It follows a modular approach. API gateway helps to interact users with loosely coupled services. Services have their own database. Development in services can be done independently. If one service fails, it doesn't affect other services. Also, it allows multiple technology stacks. It has some disadvantages like complex to implement,

multiple databases need to be handled carefully and testing is complex due to distributed architecture.

---------------------------------------------------------------------------------------------------------------------

Multi-value Answer Policies let you configure route53 to return multiple values such as IP addresses for your web servers, in response to DNS queries.

Multiple values can be specified for almost any record. Route53 automatically performs health checks on resources and only returns values of ones deemed healthy.

It is used in load balancer scenarios. It ensures high availability. It gives only a healthy server IP address

-----------------------------------------------------------------------------------------------------------------

CDN (Content Delivery Network) is a service that gives business and web application developers an easy and cost-effective way to distribute content with low latency and high data transfer speed. Example: Caching of website content. Cloudfront is an example of an AWS CDN service.

Two services that cannot survive without CDN are on-demand and live streaming services and social media.

Static Website: Content of webpages cannot be changed at runtime. No interaction with the database. HTML, CSS, Javascript is used to build. The same content is delivered every time page is loaded.

Dynamic Website: The content of web pages can be changed. Interaction with database possible. Server side language such as Node.js, PHP is used.