Name: Inayat Yousuf
Task :APACHE SERVER.

Web Server: Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP).
Basically, web servers are computers used to store HTTP files which makes a website and when a client requests a certain website, it delivers the requested website to the client. For example, you want to open Facebook on your laptop and enter the URL in the search bar of google. Now, the laptop will send an HTTP request to view the facebook webpage to another computer known as the webserver. This computer (webserver) contains all the files (usually in HTTP format) which make up the website like text, images, gif files, etc. After processing the request, the webserver will send the requested website-related files to your computer and then you can reach the website.
Different websites can be stored on the same or different web servers but that doesn't affect the actual website that you are seeing in your computer. The web server can be any software or hardware but is usually a software running on a computer. One web server can handle multiple users at any given time which is a necessity otherwise there had to be a web server for each user and considering the current world population, is nearly close to impossible. A web server is never disconnected from the internet because if it was, then it won't be able to receive any requests, and therefore cannot process them.
There are many web servers available in the market both free and paid.

- Apache HTTP server
- Microsoft Internet Information Services (IIS)
- Lighttpd
- Jigsaw Server

Apache is just one component that is needed in a web application stack to deliver web content. One of the most common web application stacks involves LAMP, or Linux, Apache, MySQL, and PHP.

Linux is the operating system that handles the operations of the application. Apache is the web server that processes requests and serves web assets and content via HTTP. MySQL is the database that stores all your information in an easily queried format. PHP is the programming language that works with apache to help create dynamic web content.

While actual statistics may vary, it's fair to say a large portion of web applications run on some form of the LAMP stack because it is easy to build and also free to use. For the most part, web applications tend to generally have similar architecture and structure even though they serve many different functions and purposes. Most web applications also benefit from Firewalls, Load Balancers, Web Servers, Content Delivery Networks, and Database Servers.

Firewalls help protect the web application from both external threats and internal vulnerabilities depending on where the firewalls are configured. Load Balancers help distribute traffic across the web servers which handle the HTTP(S) requests (this is where Apache comes in) and application servers (servers that handle the functionality and workload of the web app.) We also have Database Servers, which handle asset storage and backups. Depending on your infrastructure, your database and application can both live on the same server although it's recommended to keep those separate.

Apache functions as a way to communicate over networks from client to server using the TCP/IP protocol. Apache can be used for a wide variety of protocols, but the most common is HTTP/S. HTTP/S or Hyper Text Transfer Protocol (S stands for Secure) is one of the main protocols on the web, and the one protocol Apache is most known for.

HTTP/S is used to define how messages are formatted and transmitted across the web, with instructions for browsers and servers on how to respond to various requests and commands. Hypertext Transfer Protocol Secure is usually through port 443 with the unsecured protocol being through port 80.

The Apache server is configured via config files in which modules are used to control its behavior. By default, Apache listens to the IP addresses configured in its config files that are being requested. This is where one of Apaches many strengths come into play.

With the Listen directive, Apache can accept and route specific traffic to certain ports and domains based on specific address-port combination requests. By default, Listen runs on port 80 but Apache can be bound to different ports for different domains, allowing for many different websites and domains to be hosted and a single server. You can have domain1.com listening on port 80, domain2.com on port 8080 and domain3.com on port 443 using HTTPS all on Apache.

Once a message reaches its destination or recipient, it sends a notice, or ACK message, basically giving acknowledgment to the original sender that their data has successfully arrived. If there’s an error in receiving data, or some packets were lost in transit, the destination host or client sends a Not Acknowledged, or NAK message, to inform the sender that the data needs to be retransmitted.

Apache Installation:

first chech if already installed with command :
rpm -q httpd

Installation:

yum install httpd   yum means yellowdog updater modified
if you want more services to install then install these

yum install mod-perl
yum install php php-mysql
yum install openssl mod-ssl

Managing services

systemctl status httpd   check the status of server
systemctl start/stop/reload/restart/ httpd   used to start or stop or reload or restart apache server
netstat -tupan | grep -i http   to check on which port apache is listening.
httpd -v   used ot check the version of apache server

httpd -t used ot check syntax for configuration.

Main configuration file (/etc/httpd/conf/httpd.conf)

serverRoot -- root directory for web server configuration
listen -- listen on interface and port specified
include – includes files into the configuration
user/group – user and group for apache to run as
servertoken – default OS, change to productonly
serverSignature – set to off