PG Department of Computer Applications

III BCA

Subject : COMPUTER NETWORKS

Sub.Code : 16SCCCA8

Staff in Charge : Asst.Prof. G.Shakunthala

Asst.Prof. S.Sathiyavani

UNIT - V

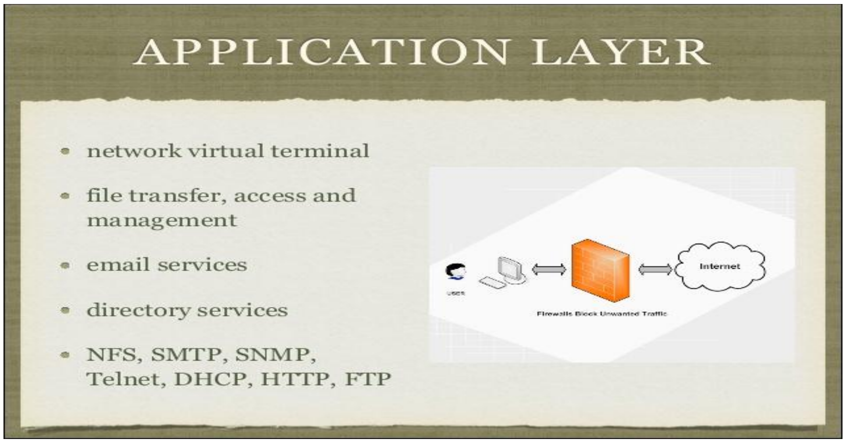
APPLICATION LAYER

Introduction:

- An **application layer** is an abstraction layer that specifies the shared communications protocols and interface methods used by hosts in a communications network.
- The application layer abstraction is used in both of the standard models of computer networking: the Internet Protocol Suite (TCP/IP) and the OSI model.
- TCP/IP, the application layer contains the communications protocols and interface methods used in process-to-process communications across an Internet Protocol (IP) computer network.
- Remote login to hosts: Telnet
- File transfer: File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP)
- Electronic mail transport: Simple Mail Transfer Protocol (SMTP)
- Networking support: Domain Name System (DNS)
- Host initialization: BOOTP
- Remote host management: Simple Network Management Protocol (SNMP), Common Management Information Protocolover TCP (CMOT)

Functions of the application layer:

- Ensures that the receiving device is identified, can be reached and is ready to accept data.
- Enables, if appropriate, authentication to occur between devices for an extra layer of security.
- Makes sure necessary communication interfaces exist. For example, is there an Ethernetor Wi-Fi interface in the sender's computer?
- Ensures agreement at both ends about error recovery procedures, data integrity and privacy.
- Determines protocol and data syntax rules at the application level.
- Presents the data on the receiving end to the user application.



APPLICATION LAYER

- network virtual terminal
- file transfer, access and management
- email services
- directory services
- NFS, SMTP, SNMP, Telnet, DHCP, HTTP, FTP

USER
Internet
Firewalls Block Unwanted Traffic

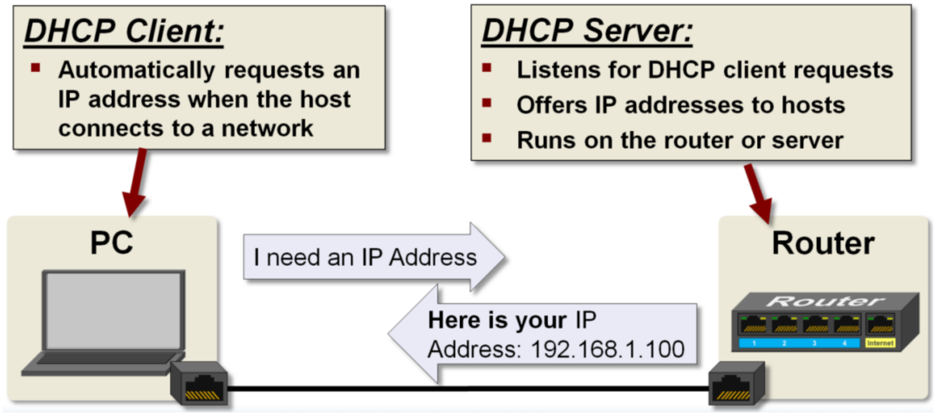
Client Server Model Defined

The **client-server** programming model is a distributed computing architecture that segregates information users (clients) from information providers (servers).

- A **client** is an application that needs something like a web page or IP address from a server. Clients may contact a server for this information at any time. Clients are information users.

- A **server** is an application that provides information or resources to clients. It needs to be always up and running, waiting for requests from clients.

Example: DHCP Client Server

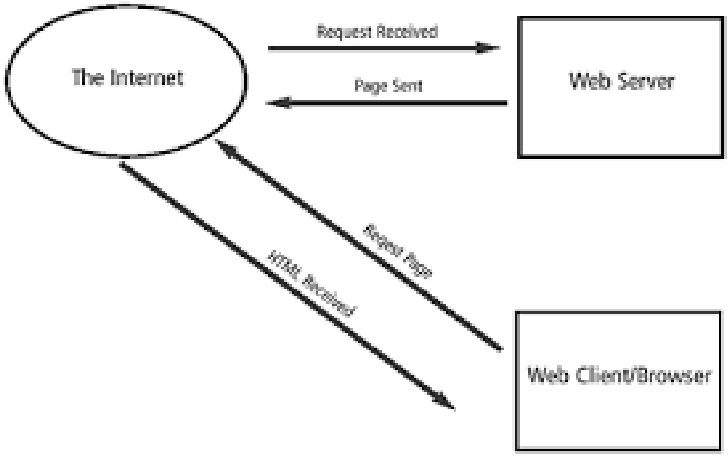


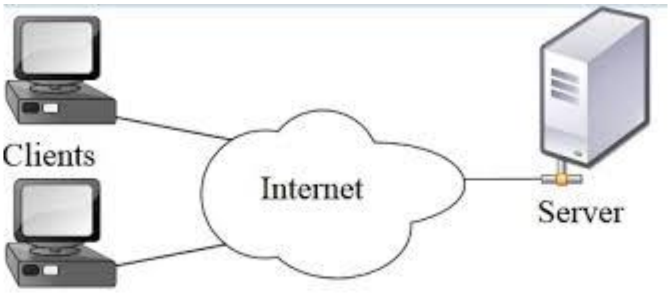
DHCP Client:

- Automatically requests an IP address when the host connects to a network

DHCP Server:

- Listens for DHCP client requests
- Offers IP addresses to hosts
- Runs on the router or server

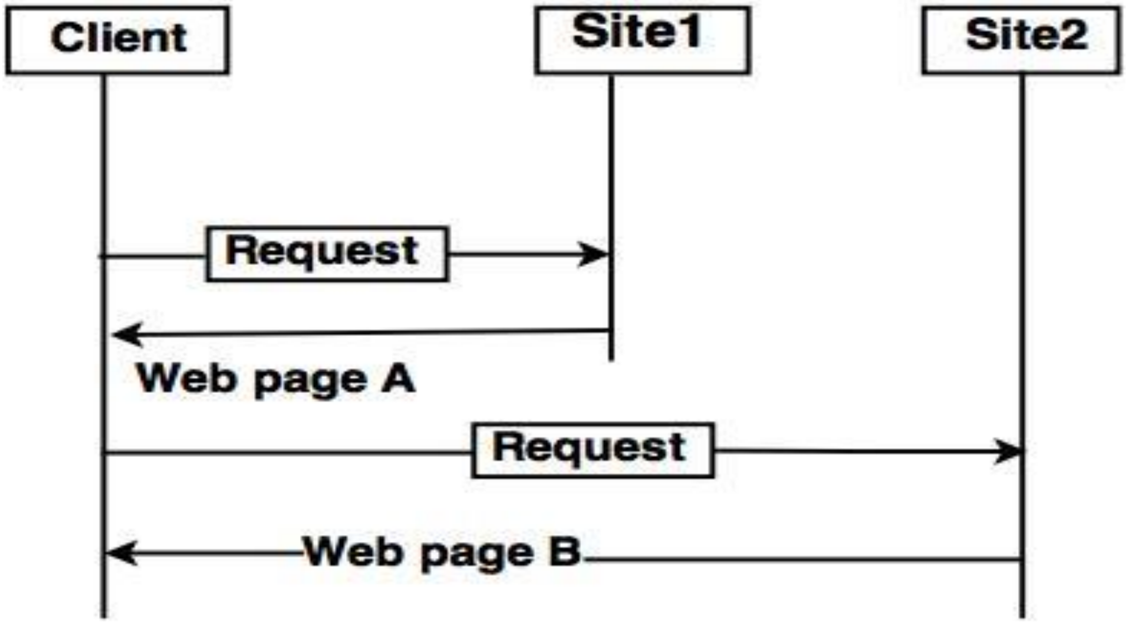




World Wide Web

Introduction to World Wide Web

- The World Wide Web (WWW) is a collection of documents and other web resources which are identified by URLs, interlinked by hypertext links, and can be accessed and searched by browsers via the Internet.
- World Wide Web is also called the Web and it was invented by Tim Berners-Lee in 1989.
- Website is a collection of web pages belonging to a particular organization.
- The pages can be retrieved and viewed by using browser.



Architecture of WWW

Client (Browser):

- Web browser is a program, which is used to communicate with web server on the Internet.
- Each browser consists of three parts: a controller, client protocol and interpreter.

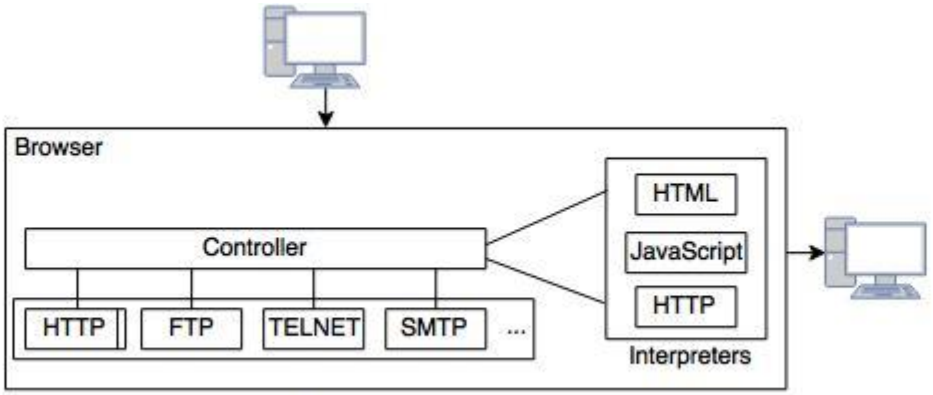


Fig: Cllient (Browser)

Server:

- A computer which is available for the network resources and provides service to the other computer on request is known as server.
- The web pages are stored at the server.
- Server accepts a TCP connection from a client browser.

Uniform Resource Locater (URL)

- The URL is a standard for specifying any kind of information on the Internet.
- The URL consists of four parts: protocol, host computer, port and path.
- The protocol is the client or server program which is used to retrieve the document or file. The protocol can be ftp or http.

FTP (File Transfer Protocol)

- **FTP** is File Transfer Protocol.
- It used to exchange files on the internet.
- To enable the data transfer FTP uses TCP/IP, FTP is most commonly used to upload and download files from the internet.
- FTP can be invoked from the command prompt or some graphical user interface.
- FTP also allows to update (delete, rename, move, and copy) files at a server.

DHCP

- **Dynamic Host Configuration Protocol** (DHCP) is used assigning IP addresses to computers in a network.
- The IP addresses are assigned dynamically.
- Certainly, using DHCP, the computer will have a different IP address every time it is connected to the network.

HTTP

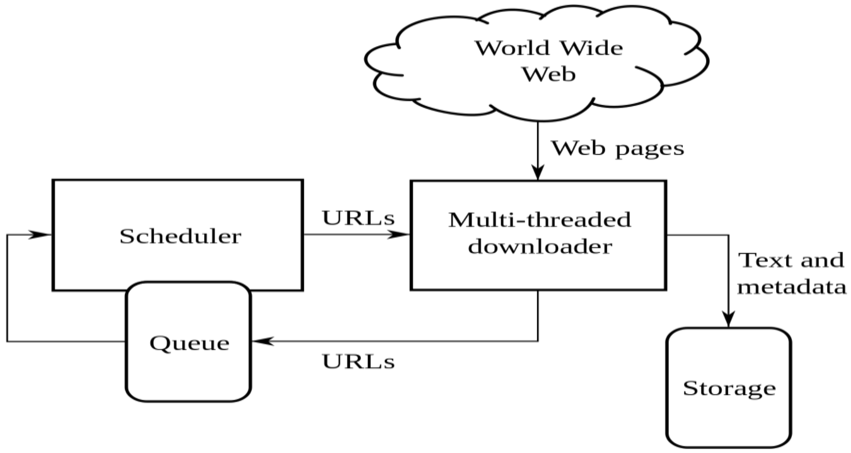
- The Hypertext Transfer Protocol (HTTP) is used to access the data on WWW(World Wide Web).
- The functions of HTTP are the **combination of FTP (File Transfer Protocol) and SMTP (Simple Mail Transfer Protocol).**
- HTTP is similar to FTP, because it uses only one TCP connection (data is transferred between Client and Sever).
- In SMTP, the messages are stored and then forwarded to the destination, but the HTTP messages are delivered immediately.
- HTTP uses the services of TCP on well-known port 80.

Telnet

- It is an application layer protocol, which can be used on the internet or LAN(Local Area Network).
- It provides a bi-directional interactive text oriented communication service by using **virtual terminal connection.**
- Telnet is basically a client server protocol, which is based on a reliable connection-oriented transport.
- It uses a port number 23, to establish the connection with TCP (Transmission Control Protocol).

SMTP (Simple Mail Transport Protocol)

- It is an internet standard for **e-mail Transmission.**
- SMTP connections are **secured with SSL (Secure Socket Layer).**
- In SMTP, the messages are stored and then forwarded to the destination.
- SMTP uses a port number 25 of TCP.



HTTP

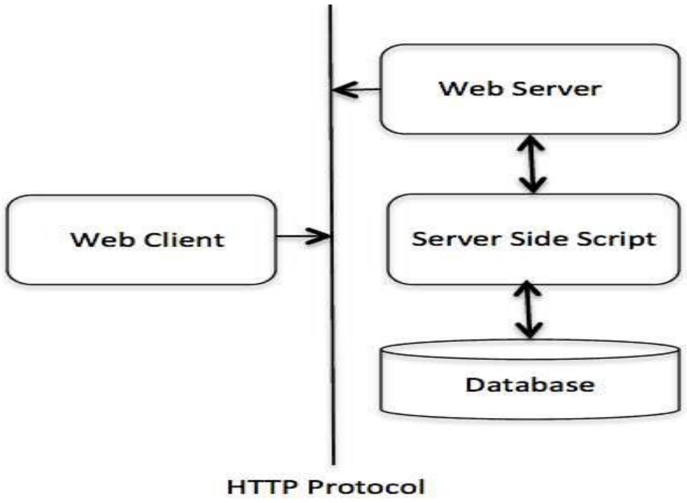
- HTTP full form *HyperText Transfer Protocol* used mainly to access data on the World Wide Web.
- HTTP is a Server and Client communication Protocol, which is primarily set of rules for formating and transferring webpage data (text, images, video and Multimedia files) over the world wide web.

- This is the Protocol used to create communication between Web Servers and Web Users. HTTP is an application layer Protocol that works on the top of the TCP/IP suite of Protocols.
  - **For Example**, A client which is use web browser and a server is a Web host that hosts the website. Whenever a client transmits a request to the Website server, HTTP protocol proceeds that request and creates a connection between client and server through TCP. After that HTTP sends a request to the server, which picks up the requested data and HTTP sends the response back to the client. Let's look into the depth how these requests work.

Features of HTTP Protocol

- Http is Connectionless**:**
- Http is media independent**:**
- Http is stateless

Basic architecture client/server



Http Client: The http client makes the request in the form of a request method. Which is followed by the message body over a TCP/IP connection.

Http Server: The request sent by the client, is responded by the server in the form of a status line followed by the other necessary information with the message body.

HTTP Request Methods

Http Protocol can use two case sensitive request-response Methods between client/server such as GET and POST that are used to handle form submissions.

GET Method

A GET Requests data from a specified resource using a given URI to retrieve data.

POST Method

A POST request Submits data to be processed to a specified resource to the server.

HTTP request

A Http request message consists of a request line, headers and sometimes a body.
An HTTP request is a way that web browsers ask for information to load website pages. HTTP request contains HTTP version type, a URL, HTTP request headers and HTTP body.

HTTP response

An HTTP response means when the web client gets the answer back from the web server. It contains the information that was asked for in the HTTP request. HTTP response contains an HTTP status code, HTTP response headers, and HTTP body.

HTTP is a stateless protocol:

HTTP is a stateless protocol and it was likewise necessary. Because it executes every command independently without knowing the other commands those came before it. Otherwise, if a client sends multiple requests to the web server, the web server will handle multiple requests from one connection for a long time. That's why HTTP is a stateless protocol.

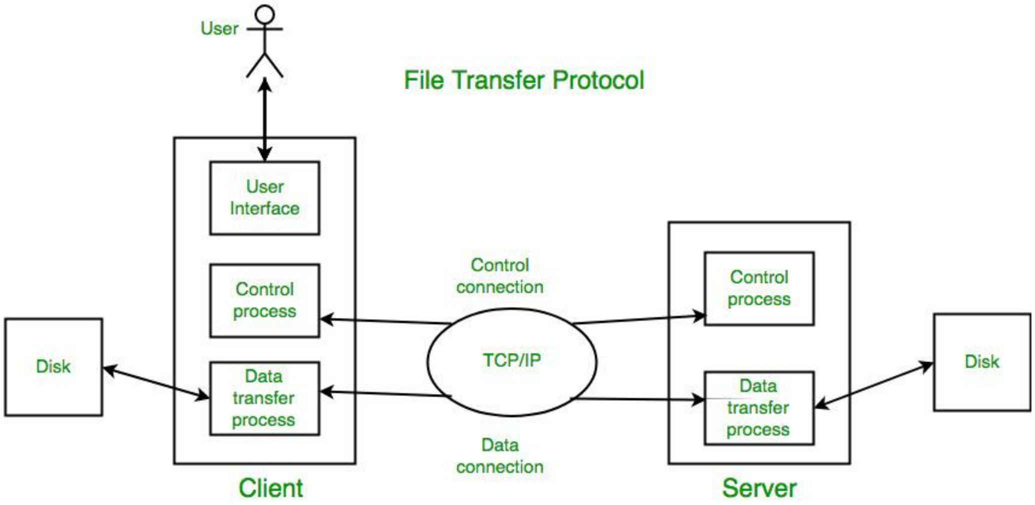
Working of the HTTP protocol

The working of the HTTP protocol is explained step by step as follows:
Step:1 **Establishing a TCP/IP connection by the client.**

Step 2: **Initiating an HTTP GET request to the server by the client.**

File Transfer Protocol

- The **File Transfer Protocol** (**FTP**) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.
- FTP is built on a client-server model architecture using separate control and data connections between the client and the server.



Type

1. FTP (File Transfer Protocol)

2. HTTP (Hypertext Transfer Protocol)

3. FTPS (FTP over SSL)

4. HTTPS (HTTP over SSL)

5. SFTP (SSH File Transfer Protocol)

6. SCP (Secure Copy)

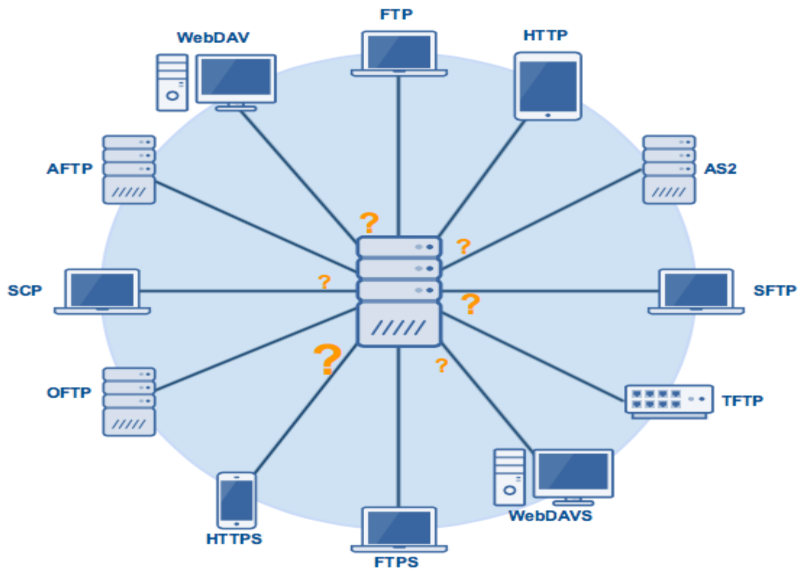
7. WebDAV (Web Distributed Authoring and Versioning)

8.TFTP (Trivial File Transfer Protocol)

9. AS2 (Applicability Statement 2)

10. OFTP (Odette File Transfer Protocol)

11. AFTP (Accelerated File Transfer Protocol)



E-mail Protocols

E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as **SMTP, POP,** and **IMAP.**

SMPTP

SMTP stands for **Simple Mail Transfer Protocol**. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

Key Points:

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.

IMAP

IMAP stands for **Internet Mail Access Protocol.** It was first proposed in 1986. There exist five versions of IMAP as follows:

1. Original IMAP
2. IMAP2
3. IMAP3
4. IMAP2bis
5. IMAP4

Key Points:

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail. It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.

POP

POP stands for Post Office Protocol. It is generally used to support a single client. There are several versions of POP but the POP 3 is the current standard.

Key Points

- POP is an application layer internet standard protocol.
- Since POP supports offline access to the messages, thus requires less internet usage time.
- POP does not allow search facility.
- In order to access the messaged, it is necessary to download them.
- It allows only one mailbox to be created on server.
- It is not suitable for accessing non mail data.
- POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.

E-mail System

E-mail system comprises of the following three components:

- Mailer
- Mail Server
- Mailbox

Mailer

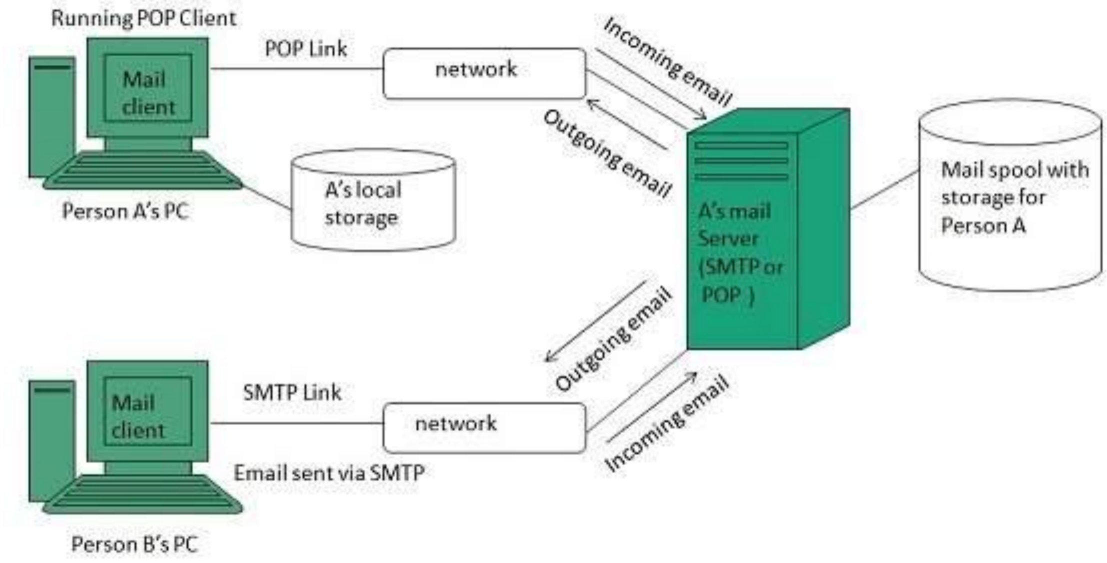
It is also called **mail program, mail application** or **mail client.** It allows us to manage, read and compose e-mail.

Mail Server

The function of mail server is to receive, store and deliver the email. It is must for mail servers to be Running all the time because if it crashes or is down, email can be lost.

Mailboxes

Mailbox is generally a folder that contains emails and information about them.



Example: Creating Email Account

E-mail Hacking

Email hacking can be done in any of the following ways:

- Spam
- Virus
- Phishing

Spam

E-mail spamming is an act of sending **Unsolicited Bulk E-mails (UBI)** which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services.

Virus

Some emails may incorporate with files containing malicious script which when run on your computer may lead to destroy your important data.

DOMAIN NAME SYSTEM

The Domain Name System (**DNS**) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private **network**. It associates various information with domain names assigned to each of the participating entities.

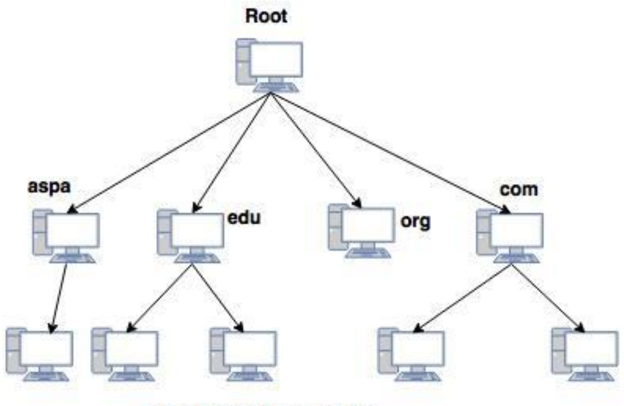


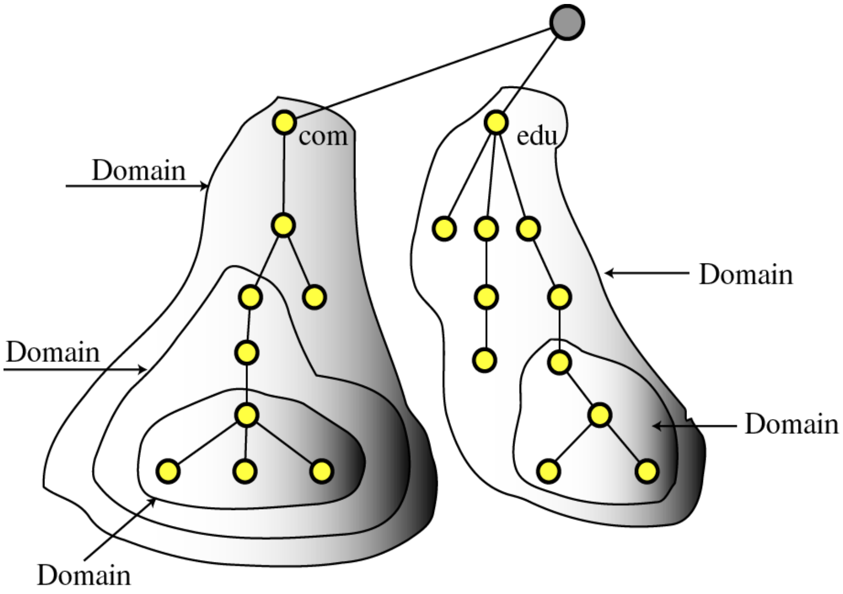
Fig: Hierarchy of DNS

DNS Components

There are 3 components:

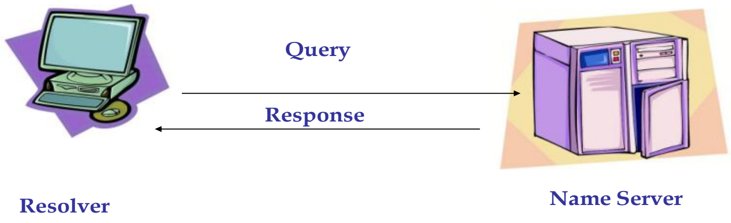
- ❖ Name Space:
  - Specifications for a structured name space and data associated with the names
- ❖ Resolvers:
  - Client programs that extract information from Name Servers.
- ❖ Name Servers:
  - Server programs which hold information about the structure and the names.

Name Space

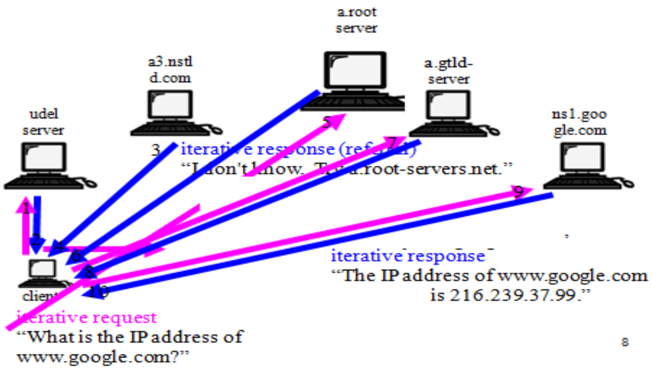


Resolvers

A Resolver maps a name to an address and vice versa.



Iterative Resolution

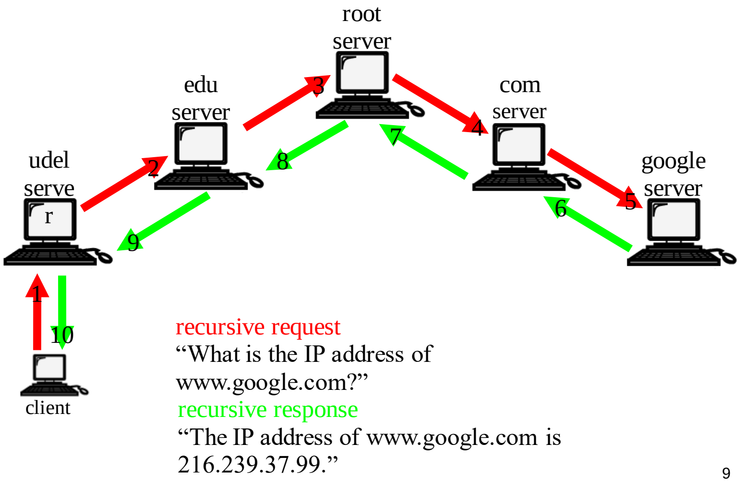


iterative response (referral)
"I don't know. Try a.root-servers.net."

iterative response
"The IP address of www.google.com
is 216.239.37.99."

iterative request
"What is the IP address of
www.google.com?"

Recursive Resolution



recursive request
"What is the IP address of
www.google.com?"
recursive response
"The IP address of www.google.com is
216.239.37.99."

Structure

Domain name space

The domain name space consists of a tree data structure. Each node or leaf in the tree has a *label* and zero or more *resource records* (RR), which hold information associated with the domain name.

Domain name syntax, internationalization

- A domain name consists of one or more parts, technically called *labels*, that are conventionally concatenated, and delimited by dots, such as example.com.
- The right-most label conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain *com*.

Name servers

- The Domain Name System is maintained by a distributed database system, which uses the client–server model.
- The nodes of this database are the name servers.

Authoritative name server

- An *authoritative* name server is a name server that only gives answers to DNS queries from data that has been configured by an original source, for example, the domain administrator or by dynamic DNS methods, in contrast to answers obtained via a query to another name server that only maintains a cache of data.

Subject : PROGRAMMING IN PHP

Sub.Code : 16SCCCA9

Staff in Charge : Asst.Prof. P.Jayalakshmi

Asst.Prof. D.Arokia Mary

V UNIT

I. AJAX –Introduction

- AJAX is about updating parts of a web page, without reloading the whole page.

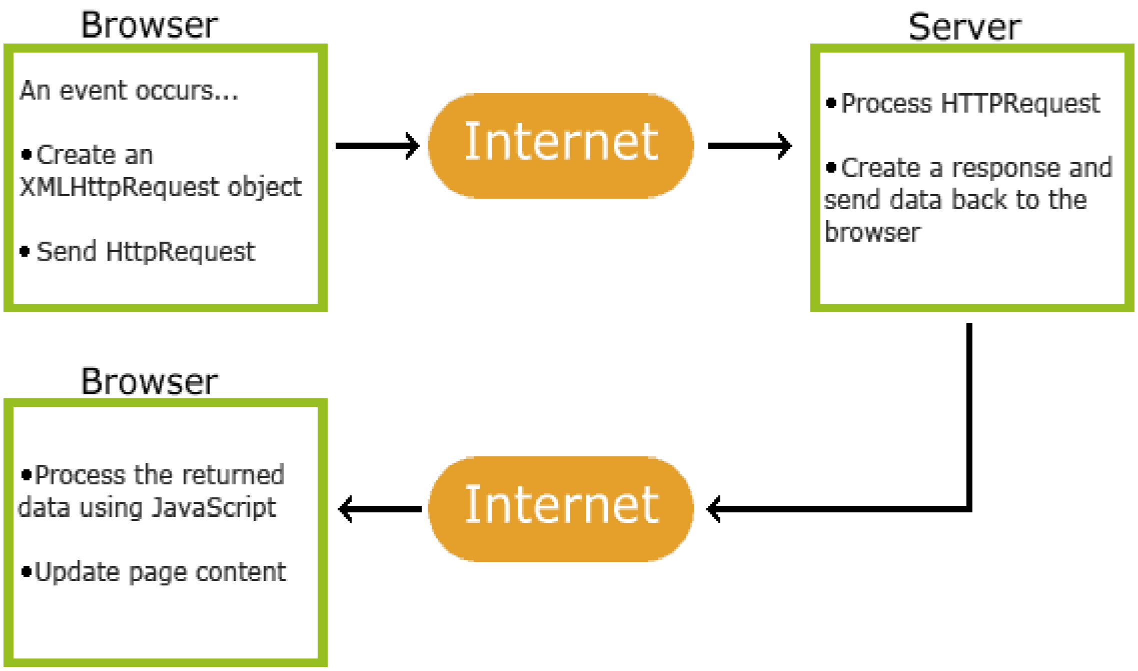
What is AJAX?

- AJAX = Asynchronous JavaScript and XML.
- AJAX is a technique for creating fast and dynamic web pages.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.
- Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

JavaScript and XML.

- **JavaScript** is a client side scripting language. It is executed on the client side by the web browsers that support JavaScript.JavaScript code only works in browsers that have JavaScript enabled.
- **XML** is the acronym for Extensible Markup Language. It is used to encode messages in both human and machine readable formats. It’s like HTML but allows you to create your custom tags. For more details on XML, see the article on XML

How AJAX Works


Browser
An event occurs...
•Create an XMLHttpRequest object
•Send HttpRequest
Internet
Server
•Process HTTPRequest
•Create a response and send data back to the browser
Browser
•Process the returned data using JavaScript
•Update page content
Internet

AJAX is Based on Internet Standards

AJAX is based on internet standards, and uses a combination of:

- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML (often used as the format for transferring data)

AJAX applications are browser- and platform-independent!

Why use AJAX?

- It allows developing rich interactive web applications just like desktop applications.

- Validation can be performed done as the user fills in a form without submitting it. This can be achieved using auto completion. The words that the user types in are submitted to the server for processing. The server responds with keywords that match what the user entered.
- It can be used to populate a dropdown box depending on the value of another dropdown box
- Data can be retrieved from the server and only a certain part of a page updated without loading the whole page. This is very useful for web page parts that load things like
- Tweets
- comments
- Users visiting the site etc.

<u>How to Create an PHP Ajax application</u>

- Create a simple application that allows users to search for popular PHP MVC frameworks.
- Our application will have a text box that users will type in the names of the framework.
- We will then use mvc AJAX to search for a match then display the framework's complete name just below the search form.

<u>Step 1) Creating the index page</u>

<u>Index.php</u>

```
<html>
<head>
<title>PHP MVC Frameworks - Search Engine</title>
<script type="text/javascript" src="/auto_complete.js"></script>
</head>
<body>
<h2>PHP MVC Frameworks - Search Engine</h2>
<p><b>Type the first letter of the PHP MVC Framework</b>
</p><form method="POST" action="index.php">
<p><input type="text" size="40" id="txtHint"  onkeyup="showName(this.value)"></p>
  </form><p>Matches: <span id="txtName"></span></p>
</body>
</html>
```

<u>Step 2) Creating the frameworks page</u>

frameworks.php

```
<?php
$frameworks = array("CodeIgniter","ZendFramework","Cake PHP","Kohana");
$name = $_GET["name"];
if (strlen($name) > 0) {
 $match = "";
```

```
  for ($i = 0; $i < count($frameworks); $i++) {
   if (strtolower($name) == strtolower(substr($frameworks[$i], 0, strlen($name)))) {

            if ($match == "") {
            $match = $frameworks[$i]; } else
          {

                $match = $match . " , " . $frameworks[$i];

            }

        }

    }

}
echo ($match == "") ? 'no match found' : $match;
?>
```

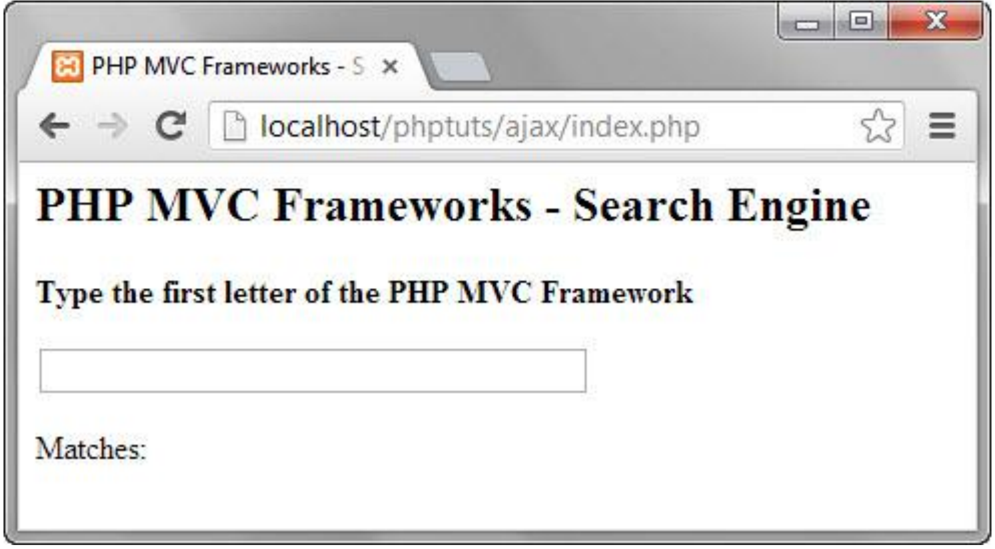
Step 3) Creating the JS script

auto_complete.js

```
functionshowName(str){
if (str.length == 0){ //exit function if nothing has been typed in the textbox
        document.getElementById("txtName").innerHTML=""; //clear previous results
return;
   }
  if (window.XMLHttpRequest) {// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
 } else {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
 }
  xmlhttp.onreadystatechange=function() {
if (xmlhttp.readyState == 4 &&xmlhttp.status == 200){
   document.getElementById("txtName").innerHTML=xmlhttp.responseText;
 }
 }
   xmlhttp.open("GET","frameworks.php?name="+str,true);
   xmlhttp.send();

}
```

Step 4) Testing our PHP Ajax application

Assuming you have saved the file index.php In phututs/ajax, browse to the URL http://localhost/phptuts/ajax/index.php

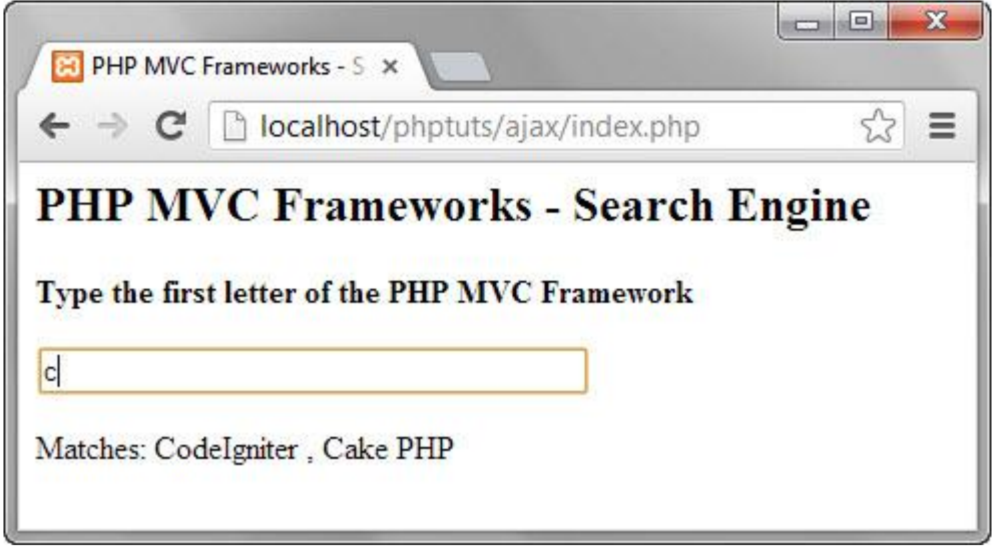


PHP MVC Frameworks - Search Engine

Type the first letter of the PHP MVC Framework

Matches:

Type the letter C in the text box You will get the following results.



PHP MVC Frameworks - Search Engine

Type the first letter of the PHP MVC Framework

c

Matches: CodeIgniter , Cake PHP

Ajax Image Upload using PHP and jQuery

blog post we have covered one of the topic of ajax i.e. to **upload images using Ajax and php** without page refresh.

Short Description :

In this process the image is selected first and previewed before storing it in to any location . Then using jQuery Ajax, it is send to php script on submit button event.

PHP script stores the image in a defined location and returns the message of the success and failure of the process.

Here we have applied some validations on the selected file, to check it whether it is an image file (jpg, jpeg or png) or any other file type.

Upload File size allowed is 100 Kb.

Following steps are pointed under this process :

1. Include jQuery library.

2. HTML page with upload field.

3. jQuery Ajax code.

3. PHP script to store image.

Ajax Code :

```
$.ajax({

url: "ajax_php_file.php", // Url to which the request is send

type: "POST",             // Type of request to be send, called as method

data: new FormData(this), // Data sent to server, a set of key/value pairs (i.e. form fields and values)

contentType: false,       // The content type used when sending data to the server.

cache: false,             // To unable request pages to be cached

processData:false,        // To send DOMDocument or non processed data file it is set to false
success: function(data)   // A function to be called if request succeeds

{

$('#loading').hide();

$("#message").html(data);

}

});
```

PHP code used to store the image :

```


$sourcePath = $_FILES['file']['tmp_name'];       // Storing source path of the file in a variable

$targetPath = "upload/".$_FILES['file']['name']; // Target path where file is to be stored

move_uploaded_file($sourcePath,$targetPath) ;    // Moving Uploaded file
```

Creating and Drawing Images

- An *image* is a rectangle of pixels that have various colors. Colors are identified by their position in the *palette*, an array of colors. Each entry in the palette has three separate color values—one for red, one for green, and one for blue. Each value ranges from 0 (this color not present) to 255 (this color at full intensity).
- Image files are rarely a straightforward dump of the pixels and the palette. Instead, various *file formats* (GIF, JPEG, PNG, etc.) have been created that attempt to compress the data somewhat to make smaller files.
- For now, let's start with the simplest possible GD example. Example 1 is a script that generates a black filled square. The code works with any version of GD that supports the PNG image format.

Example 1. A black square on a white background (black.php)

```
<?php

 $im = ImageCreate(200,200);

 $white = ImageColorAllocate($im,0xFF,0xFF,0xFF);

 $black = ImageColorAllocate($im,0x00,0x00,0x00);

ImageFilledRectangle($im,50,50,150,150,$black);

header('Content-Type: image/png');

ImagePNG($im);

?>
```

Example 1 illustrates the basic steps in generating any image: creating the image, allocating colors, drawing the image, and then saving or sending the image. Figure 1 shows the output of Example 1.

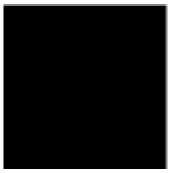


Figure 1. A black square on a white background

To see the result, simply point your browser at the *black.php* PHP page. To embed this image in a web page, use:

```
<imgsrc="black.php">
```

The Structure of a Graphics Program

Most dynamic image-generation programs follow the same basic steps outlined in Example 1.

You can create a 256-color image with the `ImageCreate( )` function, which returns an image handle:

- $image = ImageCreate(*width*, *height*);

All colors used in an image must be allocated with the `ImageColorAllocate( )` function. The first color allocated becomes the background color for the image.[4]

[4]This is true only for images with a color palette. True color images created using `ImageCreateTrueColor( )` do not obey this rule.

- $color = ImageColorAllocate(*image*, *red*, *green*, *blue*);

The arguments are the numeric RGB (red, green, blue) components of the color. In Example 1, we wrote the color values in hexadecimal, to bring the function call closer to the HTML color representation `"#FFFFFF"` and `"#000000"`.

There are many drawing primitives in GD. Example 1 uses `ImageFilledRectangle( )`, in which you specify the dimensions of the rectangle by passing the coordinates of the top-left and bottom-right corners:

- ImageFilledRectangle(*image*, *tlx*, *tly*, *brx*, *bry*, *color*);

The next step is to send a Content-Type header to the browser with the appropriate content type for the kind of image being created. Once that is done, we call the appropriate output function. The `ImageJPEG( )`, `ImagePNG( )`, and `ImageWBMP( )` functions create JPEG, PNG, and WBMP files from the image, respectively:

- ImageJPEG(*image* [, *filename* [, *quality*]]);

ImagePNG(*image* [, *filename*]);

ImageWBMP(*image* [, *filename*]);

If no *`filename`* is given, the image is sent to the browser. The *`quality`* argument for JPEGs is a number from 0 (worst-looking) to 10 (best-looking). The lower the quality, the smaller the JPEG file. The default setting is 7.5.

In Example 1, we set the HTTP header immediately before calling the output-generating function `ImagePNG( )`. If, instead, you set the Content-Type at the very start of the script, any errors that are generated are treated as image data and the browser displays a broken image icon. Table 1 lists the image formats and their Content-Type values.

Table 1. Content-Type values for image formats

Format	Content-Type
GIF	image/gif
JPEG	image/jpeg
PNG	image/png
WBMP	image/vnd.wap.wbmp

Changing the Output Format

As you may have deduced, generating an image stream of a different type requires only two changes to the script: send a different Content-Type and use a different image-generating function. Example 2 shows Example 1 modified to generate a JPEG instead of a PNG image.

Example 2. JPEG version of the black square

```
<?php

 $im = ImageCreate(200,200);

 $white = ImageColorAllocate($im,0xFF,0xFF,0xFF);

 $black = ImageColorAllocate($im,0x00,0x00,0x00);

ImageFilledRectangle($im,50,50,150,150,$black);

header('Content-Type: image/jpeg');

ImageJPEG($im);

?>
```

Testing for Supported Image Formats

If you are writing code that must be portable across systems that may support different image formats, use the `ImageTypes( )` function to check which image types are supported. This function returns a bitfield; you can use the bitwise AND operator (`&`) to check if a given bit is set. The constants `IMG_GIF`, `IMG_JPG`, `IMG_PNG`, and `IMG_WBMP` correspond to the bits for those image formats.

Example 3 generates PNG files if PNG is supported, JPEG files if PNG is not supported, and GIF files if neither PNG nor JPEG are supported.

Example 3. Checking for image format support

```
<?php

 $im = ImageCreate(200,200);

 $white = ImageColorAllocate($im,0xFF,0xFF,0xFF);

 $black = ImageColorAllocate($im,0x00,0x00,0x00);

ImageFilledRectangle($im,50,50,150,150,$black);

if (ImageTypes( ) & IMG_PNG) {

header("Content-Type: image/png");

ImagePNG($im);

 } elseif (ImageTypes( ) & IMG_JPG) {

header("Content-Type: image/jpeg");

ImageJPEG($im);

 } elseif (ImageTypes( ) & IMG_GIF) {

header("Content-Type: image/gif");

ImageGIF($im);
}

?>
```

Reading an Existing File

If you want to start with an existing image and then modify it, use either `ImageCreateFromJPEG( )` or `ImageCreateFromPNG( ):`

- $image = ImageCreateFromJPEG(*filename*);

$image = ImageCreateFromPNG(*filename*);

Basic Drawing Functions

GD has functions for drawing basic points, lines, arcs, rectangles, and polygons. This section describes the base functions supported by GD 1.x.

The most basic function is `ImageSetPixel( )`, which sets the color of a specified pixel:

- ImageSetPixel(*image*, *x*, *y*, *color*);

There are two functions for drawing lines, `ImageLine( )` and `ImageDashedLine():`

- ImageLine(*image*, *start_x*, *start_ y*, *end_x*, *end_ y*, *color*);
- ImageDashedLine(*image*, *start_x*, *start_ y*, *end_x*, *end_ y*, *color*);

There are two functions for drawing rectangles, one that simply draws the outline and one that fills the rectangle with the specified color:

- ImageRectangle(*image*, *tlx*, *tly*, *brx*, *bry*, *color*);

ImageFilledRectangle(*image*, *tlx*, *tly*, *brx*, *bry*, *color*);

Specify the location and size of the rectangle by passing the coordinates of the top-left and bottom-right corners.

You can draw arbitrary polygons with the `ImagePolygon( )` and `ImageFilledPolygon( )` functions:

- ImagePolygon(*image*, *points*, *number*, *color*);

ImageFilledPolygon(*image*, *points*, *number*, *color*);

Both functions take an array of points. This array has two integers (the `x and y` coordinates) for each vertex on the polygon. The *`number`* argument is the number of vertices in the array (typically `count($points)/2`).

The `ImageArc( )` function draws an arc (a portion of an ellipse):

- ImageArc(*image*, *center_x*, *center_ y*, *width*, *height*, *start*, *end*, *color*);

The ellipse is defined by its center, width, and height (height and width are the same for a circle). The start and end points of the arc are given as degrees counting counterclockwise from 3 o'clock. Draw the full ellipse with a *`start`* of 0 and an *`end`* of 360.

There are two ways to fill in already-drawn shapes. The `ImageFill( )` function performs a flood fill, changing the color of the pixels starting at the given location. Any change in pixel color marks the limits of the fill. The `ImageFillToBorder( )` function lets you pass the particular color of the limits of the fill:

```
ImageFill(image, x, y, color);

ImageFillToBorder(image, x, y, border_color, color);
```

Images with Text

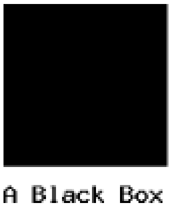
Often it is necessary to add text to images. GD has built-in fonts for this purpose. Example 4 adds some text to our black square image.

4. Adding text to an image

```
<?php

 $im = ImageCreate(200,200);

 $white = ImageColorAllocate($im,0xFF,0xFF,0xFF);

 $black = ImageColorAllocate($im,0x00,0x00,0x00);

ImageFilledRectangle($im,50,50,150,150,$black);

ImageString($im,5,50,160,"A Black Box",$black);

Header('Content-Type: image/png');
```

```
ImagePNG($im);

?>
```

Figure 2 shows the output of Example 4.



Subject : CLOUD COMPUTING

Sub.Code : 16SMBECA2:1

Staff in Charge : Asst.Prof. S.Sathiyavani

Asst.Prof. S.Indira

V UNIT

I. CLOUD APPLICATIONS

- **A cloud application, or cloud app, is a software program where cloud-based and local components work together.**

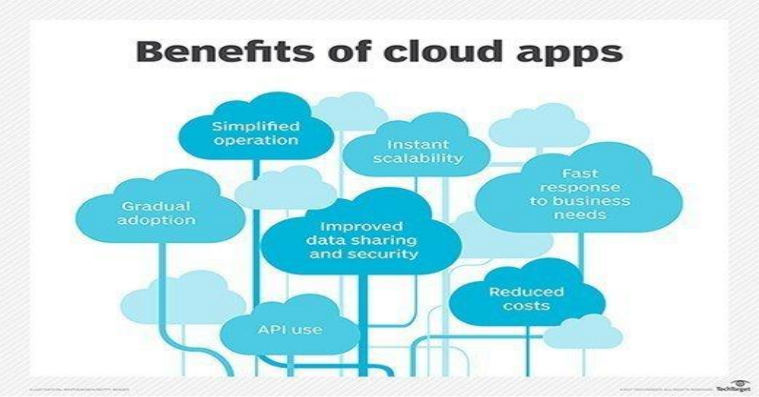
- **This model relies on remote** servers **for processing logic that is accessed through a web browser with a continual internet connection.**

- **A cloud application, or cloud app, is a software program where cloud-based and local components work together.**

- **This model relies on remote** servers **for processing logic that is accessed through a web browser with a continual internet connection.**

BENEFITS OF CLOUD APPS

- Fast response to business needs
- Simplified operation
- Instant scalability
- API use
- Gradual adoption
- Reduced costs.
- Improved data sharing and security.



II. What is Cloud Computing?

- Cloud computing is a term referred to storing and accessing of data over the internet. It doesn't store any data on the hard disk of your personal computer. In cloud computing, you access data from a remote server.

What is cloud migration? An introduction to moving to the cloud

- Cloud migration is the process of moving data, applications or other business elements to a cloud computingenvironment.

Benefits of cloud migration

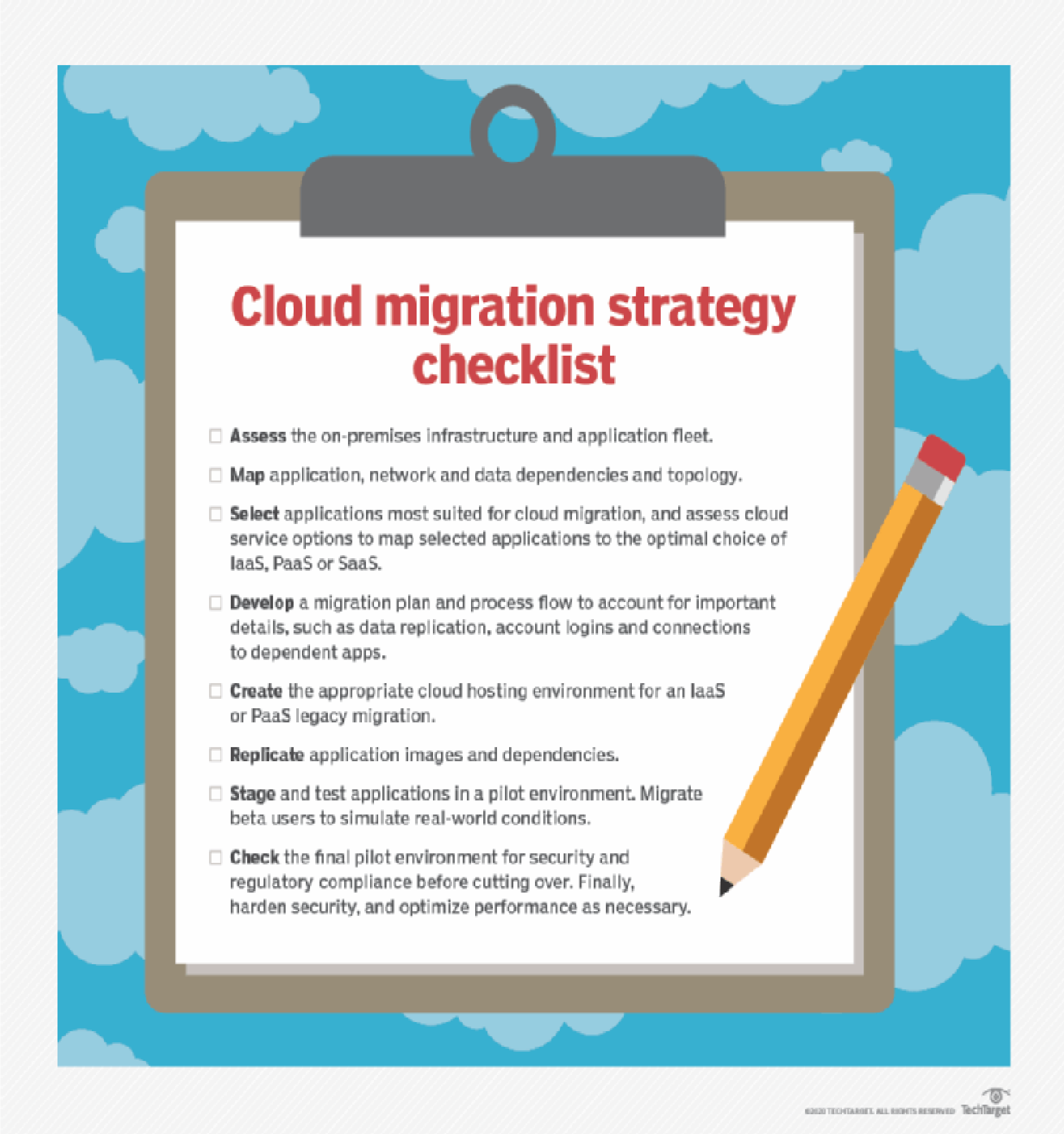
- The general goal or benefit of any cloud migration is to host applications and data in the most effective IT environment possible, based on factors such as cost, performance and security.

- For example, many organizations perform the migration of on-premises applications and data from their local data center to public cloud infrastructure to take advantage of benefits such as greater elasticity, self-service provisioning, redundancy and a flexible, pay-per-use model.

Cloud migration strategies

- Moving workloads to the cloud requires a well-thought-out strategy that includes a complex combination of management and technology challenges as well as staff and resource realignment.

- There are choices in the type of migration to perform as well as the type of data that should move. It's important to consider the following cloud migration steps before taking action
    1. Assess
    2. Map
    3. Select
    4. Develop
    5. Create
    6. Replicate
    7. Stage
    8. check



Cloud migration strategy checklist

□ **Assess** the on-premises infrastructure and application fleet.

□ **Map** application, network and data dependencies and topology.

□ **Select** applications most suited for cloud migration, and assess cloud service options to map selected applications to the optimal choice of IaaS, PaaS or SaaS.

□ **Develop** a migration plan and process flow to account for important details, such as data replication, account logins and connections to dependent apps.

□ **Create** the appropriate cloud hosting environment for an IaaS or PaaS legacy migration.

□ **Replicate** application images and dependencies.

□ **Stage** and test applications in a pilot environment. Migrate beta users to simulate real-world conditions.

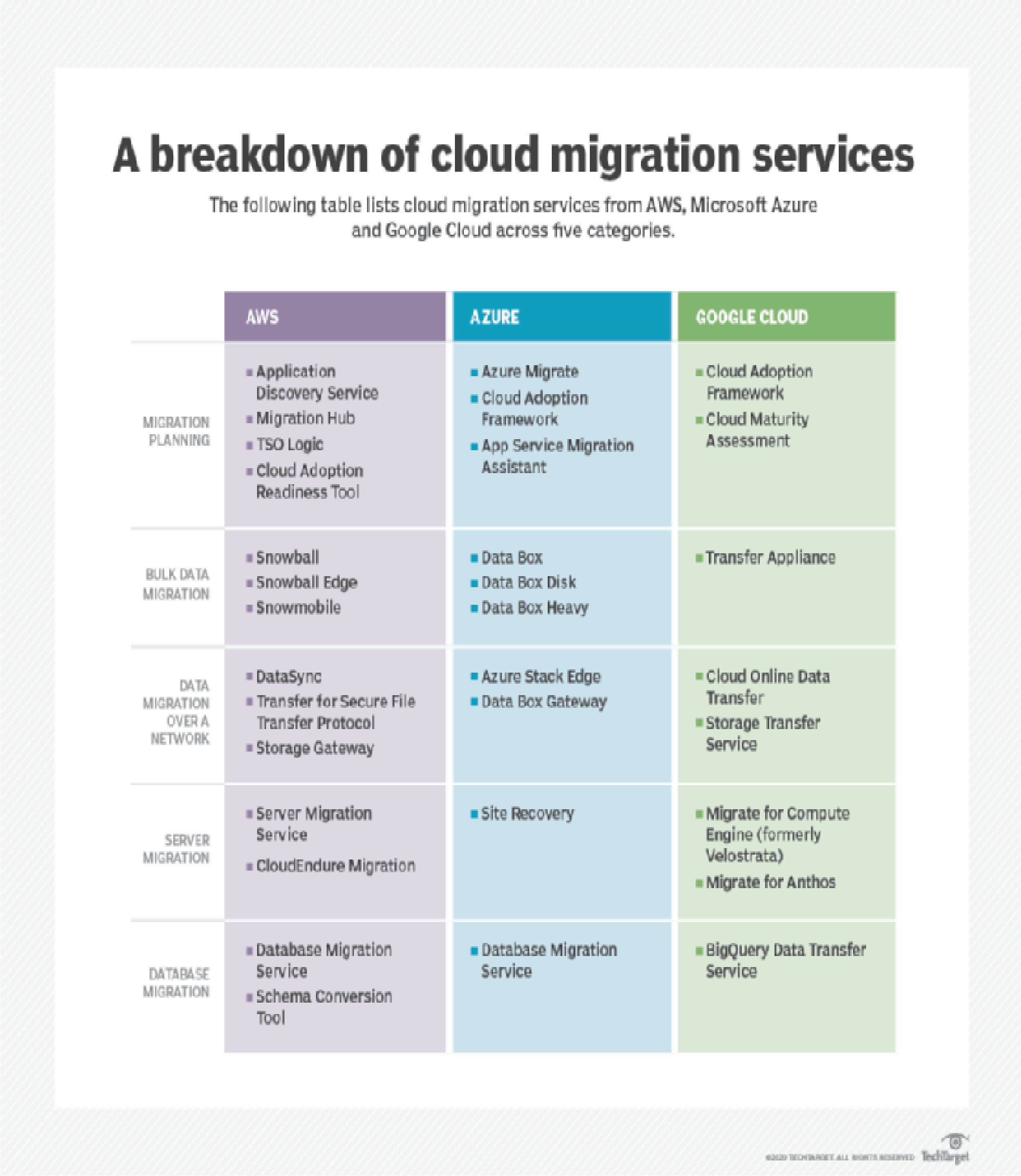
□ **Check** the final pilot environment for security and regulatory compliance before cutting over. Finally, harden security, and optimize performance as necessary.

©2020 TECHTARGET. ALL RIGHTS RESERVED TechTarget

you examine the application, it may be prudent to reconsider its strategic architecture to set it up for what could potentially be a longer life. A handful of platforms are now mainstream among hybrid and multi-cloud environments, including the following:

- Microsoft Azure Stack;

- Google Cloud Anthos;
- AWS Outposts;
- VMware Cloud on AWS; and
- a container-based PaaS, such as Cloud Foundry or Red Hat OpenShift.



A breakdown of cloud migration services

The following table lists cloud migration services from AWS, Microsoft Azure and Google Cloud across five categories.

	AWS	AZURE	GOOGLE CLOUD
MIGRATION PLANNING	▪ Application Discovery Service ▪ Migration Hub ▪ TSO Logic ▪ Cloud Adoption Readiness Tool	▪ Azure Migrate ▪ Cloud Adoption Framework ▪ App Service Migration Assistant	▪ Cloud Adoption Framework ▪ Cloud Maturity Assessment
BULK DATA MIGRATION	▪ Snowball ▪ Snowball Edge ▪ Snowmobile	▪ Data Box ▪ Data Box Disk ▪ Data Box Heavy	▪ Transfer Appliance
DATA MIGRATION OVER A NETWORK	▪ DataSync ▪ Transfer for Secure File Transfer Protocol ▪ Storage Gateway	▪ Azure Stack Edge ▪ Data Box Gateway	▪ Cloud Online Data Transfer ▪ Storage Transfer Service
SERVER MIGRATION	▪ Server Migration Service ▪ CloudEndure Migration	▪ Site Recovery	▪ Migrate for Compute Engine (formerly Velostrata) ▪ Migrate for Anthos
DATABASE MIGRATION	▪ Database Migration Service ▪ Schema Conversion Tool	▪ Database Migration Service	▪ BigQuery Data Transfer Service

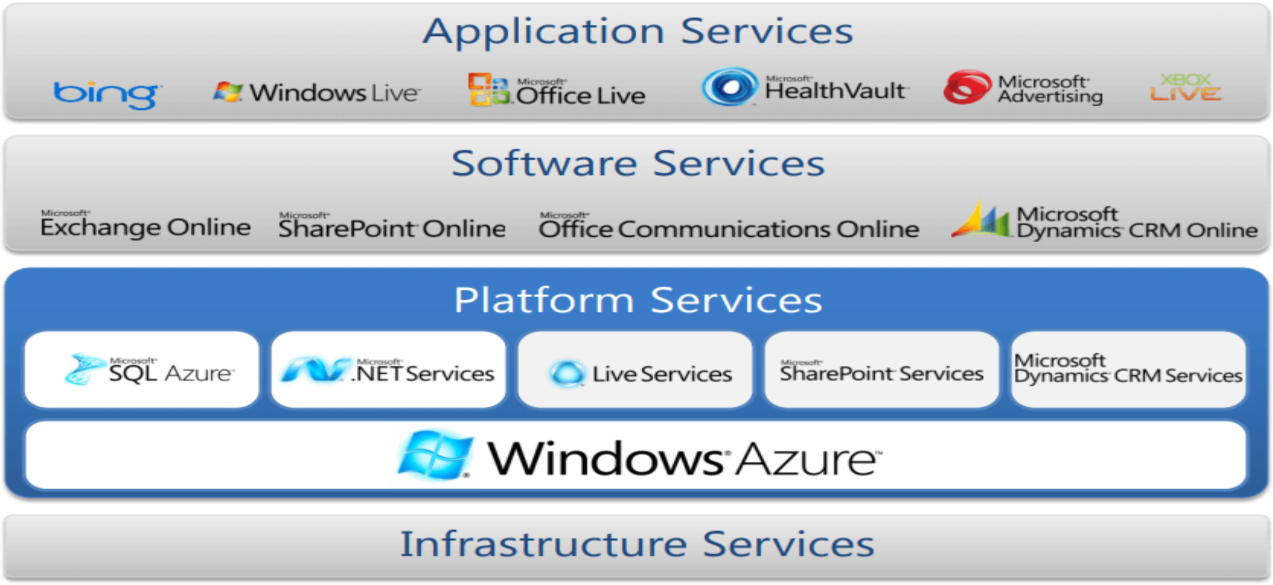
III. WHAT IS MICROSOFT AZURE

Azure is a cloud computing platform which was launched by Microsoft in February 2010. It is an open and flexible cloud platform which helps in development, data storage, service hosting, and service management. The Azure tool hosts web applications over the internet with the help of Microsoft data centers.

TYPES OF AZURE CLOUDS

There are mainly three types of clouds in Microsoft Azure are:

1. PAAS
2. SAAS
3. IASS



Azure as IaaS

IaaS(Infrastructure as a Service) is the foundational cloud platform layer. This Azure service is used by IT administrators for processing, storage, networks or any other fundamental computer operations. It allows users to run arbitrary software.

Advantages:

- It offers efficient design time portability
- It is advisable for the application which needs complete control
- IaaS offers quick transition of services to clouds

- The apparent benefit of laaS is that it frees you from the concerns of setting up many physical or virtual machines.
- Helps you to access, monitor and manage datacenters

- **Disadvantages of Iaas:**

- Plenty of security risks from unpatched servers
- Some companies have defined processes for testing and updating on-premise servers vulnerabilities. This cannot be done with Azure.

- **<u>Azure as PaaS</u>**

PaaS is a computing platform which includes an operating system, programming language execution environment, database or web services. This Azure service is used by developers and application providers.

As its name suggests, this platform is provided to the client to develop and deploy software. It allows the client to focus on application development instead of worrying about hardware and infrastructure. It also takes care of operating systems, networking and servers issues.

- **Advantages:**

- The total cost is low as the resources are allocated on demand and servers are automatically added or subtracted.
- Azure is less vulnerable because servers are automatically checked for all known security issues
- The entire process is not visible to the developer, so it does not have a risk of a data breach

- **Disadvantages:**

- Portability issues can occur when you use PaaS services
- There may be different environment at Azure, so the application needs to adapt accordingly.

<u>Azure AsSaaS</u>

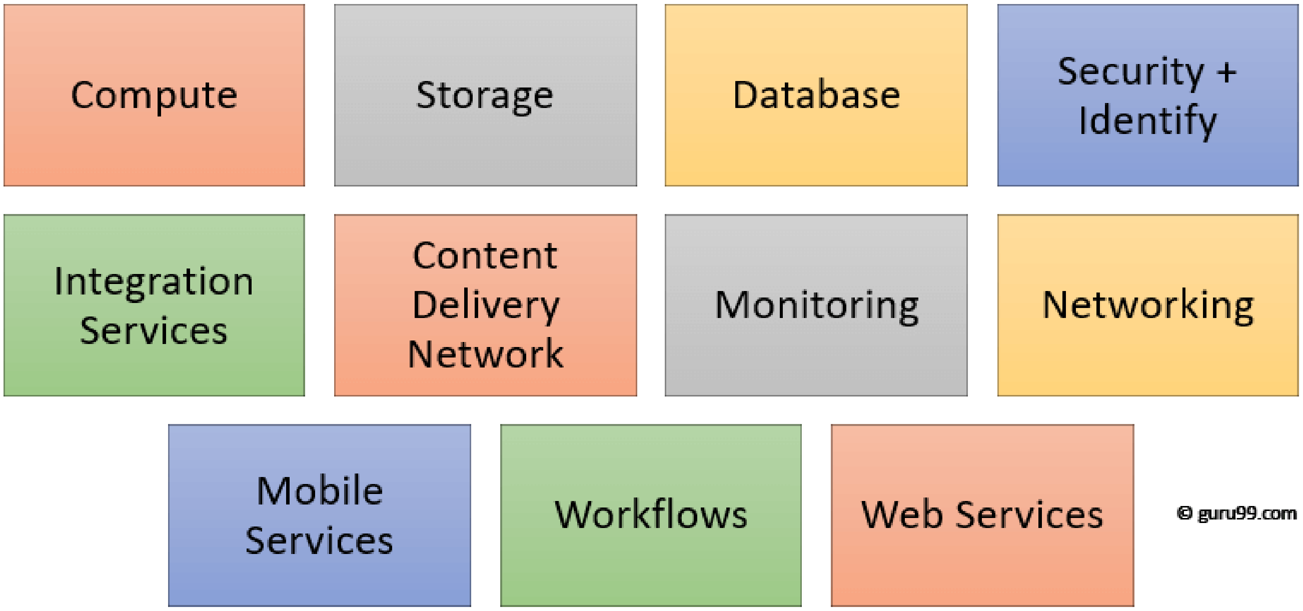
SaaS (Software as a Service) is software which is centrally hosted and managed. It is a single version of the application is used for all customers. You can scale out to multiple instances. This helps you to ensure the best performance in all locations. The software is licensed through a monthly or annual subscription. MS Exchange, Office, Dynamics are offered as a SaaS

Azure key Concepts

Concept Name	Description
Regions	Azure is a global cloud platform which is available across various regions around the world. When you request a service, application, or VM in Azure, you are first asked to specify a region. The selected region represents datacenter where your application runs.
Datacenter	In Azure, you can deploy your applications into a variety of data centers around the globe. So, it is advisable to select a region which is closer to most of your customers. It helps you to reduce latency in network requests.
Azure portal	The Azure portal is a web-based application which can be used to create, manage and remove Azure resource and services. It is located at https://portal.azure.com.
Resources	Azure resource is an individual computer, networking data or app hosting services which charged individually. Some common resources are virtual machines(VM), storage account, or SQL databases.
Resource groups	An Azure resource group is a container which holds related resource for an Azure solution. It may include every resource or just resource which you wants to manage.
Resource Manager templates	It is a JSON which defines one or more resource to deploy to a resource group. It also establishes dependencies between deployed resources.
Automation:	Azure allows you to automate the process of creating, managing and deleting resource by using PowerShell or the Azure command-line Interface(CLI).

Azure PowerShell	PowerShell is a set of modules that offer cmdlets to manage Azure. In most cases, you are allowed to use, the cmdlets command for the same tasks which you are performing in the Azure portal.
Azure command-line interface(CLI)	The Azure CLI is a tool that you can use to create, manage, and remove Azure resources from the command line.
REST APIs	Azure is built on a set of REST APIs help you perform the same operation that you do in Azure portal UI. It allows your Azure resources and apps to be manipulated via any third party software application.

- **Azure Domains (Components)**



Key Azure Components

Compute

It offers computing operations like app hosting, development, and deployment in Azure Platform. It has the following components:

- Virtual Machine: Allows you to deploy any language, workload in any operating system
- Virtual Machine Scale Sets: Allows you to create thousands of similar virtual machines in minutes

- Azure Container Service: Create a container hosting solution which is optimized for Azure. You scale and arrange applications using Kube, DC/OS, Swarm or Docker
- Azure Container Registry: This service store and manage container images across all types of Azure deployments
- Functions: Let's you write code regardless of infrastructure and provisioning of servers. In the situation when your functions call rate scales up.
- Batch: Batch processing helps you scale to tens, hundreds or thousands of virtual machines and execute computer pipelines.
- Service Fabric: Simplify microservice-based application development and lifecycle management. It supports Java, PHP, Node.js, Python, and Ruby.

Storage

Azure store is a cloud storage solution for modern applications. It is designed to meet the needs of their customer's demand for scalability. It allows you to store and process hundreds of terabytes of data. It has the following components:

- Blob Storage: Azure Blob storage is a service which stores unstructured data in the cloud as objects/blobs. You can store any type of text or binary data, such as a document, media file, or application installer.
- Queue Storage: It provides cloud messaging between application components. It delivers asynchronous messaging to establish communication between application components.
- File Storage: Using Azure File storage, you can migrate legacy applications. It relies on file shares to Azure quickly and without costly rewrites.
- Table Storage: Azure Table storage stores semi-structured NoSQL data in the cloud. It provides a key/attribute store with a schema-less design

Database

This category includes Database as a Service (DBaaS) which offers SQL and NoSQL tools. It also includes databases like Azure Cosmos DB and Azure Database for PostgreSQL. It has the following components:

- SQL Database: It is a relational database service in the Microsoft cloud based on the market-leading Microsoft SQL Server engine.
- DocumentDB: It is a fully managed NoSQL database service which is It built for fast and predictable performance and ease of development.
- Redis Cache: It is a secure and highly advanced key-value store. It stores data structures like strings, hashes, lists, etc.

Content Delivery Network

Content Delivery Network (CDN) caches static web content at strategically placed locations. This helps you to offer speed for delivering content to users. It has the following components:

- VPN Gateway: VPN Gateway sends encrypted traffic across a public connection.
- Traffic Manager: It helps you to control and allows you to do the distribution of user traffic for services like WebApps, VM, Azure, and cloud services in different Datacenters
- Express Route: Helps you to extend your on-premises networks into the Microsoft cloud over a dedicated private connection to Microsoft Azure, Office 365, and CRM Online.

Security + Identify sevices

It provides capabilities to identify and respond to cloud security threats. It also helps you to manage encryption keys and other sensitive assets. It has the following components:

- Key Vault: Azure Key Vault allows you to safeguard cryptographic keys and helps you to create secrets used by cloud applications and services.
- Azure Active Directory: Azure Active Directory and identity management service. This includes multi-factor authentication, device registration, etc.
- Azure AD B2C: Azure AD B2C is a cloud identity management solution for your consumer-facing web and mobile applications. It allows you to scales hundreds of millions of consumer identities.

Enterprise Integration Services:

- Service Bus: Service Bus is an information delivery service which works on the third-party communication system.
- SQL Server Stretch Database: This service helps you migrates any cold data securely and transparently to the Microsoft Azure cloud
- Azure AD Domain Services: It offers managed domain services like domain join, group policy, LDAP, etc. This authentication which is compatible with Windows Server Active Directory.
- Multi-Factor Authentication: Azure Multi-Factor Authentication (MFA) is two-step verification. It helps you to access data and applications to offers a simple sign-in process.

Monitoring + Management Services

These services allow easy management of Azure deployment.

- Azure Resource Manager: It makes it easy for you to manage and visualize resource in your app. You can even control who is your organization can act on the resources.
- Automation: Microsoft Azure Automation is a way to automate the manual, long-running, error-free, and constantly repeated tasks. These tasks are commonly performed in a cloud and enterprise environment.

Azure Networking

- Virtual Network: Perform Network isolation and segmentation. It offers filter and Route network traffic.
- Load Balancer: Offers high availability and network performance of any application. Load balance information Internet traffic to Virtual machines.
- Application Gateway: It is a dedicated virtual appliance that offers an Application Delivery Controller (ADC) as a service.
- Azure DNS: Azure DNS hosting service offers name resolution using Microsoft Azure infrastructure.

Web and Mobile Services:

- Web Apps: Web Apps allows you to build and host websites in the programming language of your choice without the need to manage its infrastructure.
- Mobile Apps: Mobile Apps Service offers a highly scalable, globally available mobile app development platform for users.
- API Apps: API apps make it easier to develop, host and consume APIs in the cloud and on-premises.
- Logic Apps: Logic Apps helps you to simplify and implement scalable integrations

Workflows in the cloud

It provides a visual designer to create and automate your process as a series of steps known as a workflow

- Notification Hubs: Azure Notification Hubs offers an easy-to-use, multi-platform, scaled-out push engine
- Event Hubs: Azure Event Hubs is data streaming platform which can manage millions of events per second. Data sent to an event hub can be transformed and stored using any real-time analytics offers batching/storage adapters.
- Azure Search: It is a cloud search-as-a-service solution which offers server and infrastructure management. It offers ready-to-use service that you can populate with your data. This can be used to add search to your web or mobile application.

Migration

Migration tools help an organization estimate workload migration costs. It also helps to perform the migration of workloads from your local data centers to the Azure cloud.

Traditional vs. Azure Cloud Model

Traditional	**Azure Cloud Model**
Dedicated infrastructure for each application	Loosely coupled apps and micro-services
Purpose-built hardware	Industry-standard hardware
Distinct infrastructure and operations teams	Service-focused DevOps teams
Customized processes & configurations	Standardized processes & configurations

Applications of Azure

Microsoft Azure is used in a broad spectrum of applications like:

- Infrastructure Services
- Mobile Apps
- Web Applications
- Cloud Services
- Storage, Backup, and Recovery
- Data Management
- Media Services

Advantages of Azure

Here, are advantages of using Azure:

- Azure infrastructure will cost-effectively enhance your business continuity strategy
- It allows you to access the application without buying a license for the individual machine
- Windows Azure offers the best solution for your data needs, from SQL database to blobs to tables
- Offers scalability, flexibility, and cost-effectiveness
- Helps you to maintain consistency across clouds with familiar tools and resources
- Allows you to extend data center with a consistent management toolset and familiar development and identity solutions.

- You can deploy premium virtual machines in minutes which also include Linux and Windows servers
- Helps you to scale your IT resources up and down based on your needs
- You are not required to run the high-powered and high-priced computer to run cloud computing's web-based applications.
- You will not require processing power or hard disk space if you are using Azure
- Cloud computing offers virtually limitless storage
- If your personal computer or laptop crashes, all your data is still out there in the cloud, and it is still accessible
- Sharing documents leads directly to better collaboration
- If you change your device your computers, applications and documents follow you through the cloud

DisAdvantages of Azure

- Cloud computing is not possible if you can't connect to the Internet

- Azure is a web-based application which requires a lot of bandwidth to download, as do large documents

- Web-based applications can sometimes be slower compared to accessing a similar software program on your desktop PC

IV. GOOGLE CLOUD PLATFORM

- **Google Cloud Platform (GCP)**, offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube.[1] Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning.[2] Registration requires a credit card or bank account details.[3]
- Google Cloud Platform provides Infrastructure as a service, Platform as a service, and Serverless computing environments.
- In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company. The service became generally

available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform.
- Google Cloud Platform is a part [4] of **Google Cloud**, which includes the Google Cloud Platform public cloud infrastructure, as well as **G Suite**, enterprise versions of Android and Chrome OS, and application programming interfaces (APIs) for machine learning and enterprise mapping services.

☑ Products

Compute

- App Engine - Platform as a Service to deploy Java, PHP, Node.js, Python, C#, .Net, Ruby and Go applications.
- Compute Engine - Infrastructure as a Service to run Microsoft Windows and Linux virtual machines.
- Kubernetes Engine - Containers as a Service based on Kubernetes.
- Cloud Functions - Functions as a Service to run event-driven code written in Node.js or Python.

Storage & Databases

- Cloud Storage - Object storage with integrated edge caching to store unstructured data.
- Cloud SQL - Database as a Service based on MySQL and PostgreSQL.
- Cloud BigTable- ManagedNoSQL database service.
- Cloud Spanner - Horizontally scalable, strongly consistent, relational database service.
- Cloud Datastore - NoSQL database for web and mobile applications.
- Persistent Disk - Block storage for Compute Engine virtual machines.
- Cloud MemoryStore - Managed in-memory data store based on Redis.

Networking

- VPC - Virtual Private Cloud for managing the software defined network of cloud resources.
- Cloud Load Balancing - Software-defined, managed service for load balancing the traffic.
- Cloud Armor - Web application firewall to protect workloads from DDoS attacks.
- Cloud CDN - Content Delivery Network based on Google's globally distributed edge points of presence. As of June 2018, the service is in Beta.
- Cloud Interconnect - Service to connect a data center with Google Cloud Platform
- Cloud DNS - Managed, authoritative DNS service running on the same infrastructure as Google.
- Network Service Tiers - Option to choose Premium vs Standard network tier for higher performing network.

Big Data

- BigQuery - Scalable, managed enterprise data warehouse for analytics.
- Cloud Dataflow - Managed service based on Apache Beam for stream and batch data processing.
- Cloud Dataproc - Big data platform for running Apache Hadoop and Apache Spark jobs.
- Cloud Composer - Managed workflow orchestration service built on Apache Airflow.
- Cloud Datalab - Tool for data exploration, analysis, visualization and machine learning. This is a fully managed Jupyter Notebook service.
- Cloud Dataprep - Data service based on Trifacta to visually explore, clean, and prepare data for analysis.
- Cloud Pub/Sub - Scalable event ingestion service based on message queues.
- Cloud Data Studio - Business intelligence tool to visualize data through dashboards and reports.

Cloud AI

- Cloud AutoML - Service to train and deploy custom machine learning models. As of September 2018, the service is in Beta.
- Cloud TPU - Accelerators used by Google to train machine learning models.
- Cloud Machine Learning Engine - Managed service for training and building machine learning models based on mainstream frameworks.
- Cloud Job Discovery - Service based on Google's search and machine learning capabilities for recruiting ecosystem.
- Dialogflow Enterprise - Development environment based on Google's machine learning for building conversational interfaces.

- Cloud Natural Language - Text analysis service based on Google Deep Learning models.
- Cloud Speech-to-Text - Speech to text conversion service based on machine learning.
- Cloud Text-to-Speech - Text to speech conversion service based on machine learning.
- Cloud Translation API - Service to dynamically translate between thousands of available language pairs
- Cloud Vision API - Image analysis service based on machine learning
- Cloud Video Intelligence - Video analysis service based on machine learning

Management Tools

- Stackdriver - Monitoring, logging, and diagnostics for applications on Google Cloud Platform and AWS.
- Cloud Deployment Manager - Tool to deploy Google Cloud Platform resources defined in templates created in YAML, Python or Jinja2.
- Cloud Console - Web interface to manage Google Cloud Platform resources.
- Cloud Shell - Browser-based shell command line access to manage Google Cloud Platform resources.
- Cloud Console Mobile App - Android and iOS application to manage Google Cloud Platform resources.
- Cloud APIs - APIs to programmatically access Google Cloud Platform resources

Identity & Security

- Cloud Identity - Single sign-on (SSO) service based on SAML 2.0 and OpenID.
- Cloud IAM - Identity & Access Management (IAM) service for defining policies based on role-based access control.
- Cloud Identity-Aware Proxy - Service to control access to cloud applications running on Google Cloud Platform without using a VPN.
- Cloud Data Loss Prevention API - Service to automatically discover, classify, and redact sensitive data.
- Security Key Enforcement - Two-step verification service based on a security key.
- Cloud Key Management Service - Cloud-hosted key management service integrated with IAM and audit logging.
- Cloud Resource Manager - Service to manage resources by project, folder, and organization based on the hierarchy.
- Cloud Security Command Center - Security and data risk platform for data and services running in Google Cloud Platform.
- Cloud Security Scanner - Automated vulnerability scanning service for applications deployed in App Engine.

- Access Transparency - Near real-time audit logs providing visibility to Google Cloud Platform administrators.

IoT

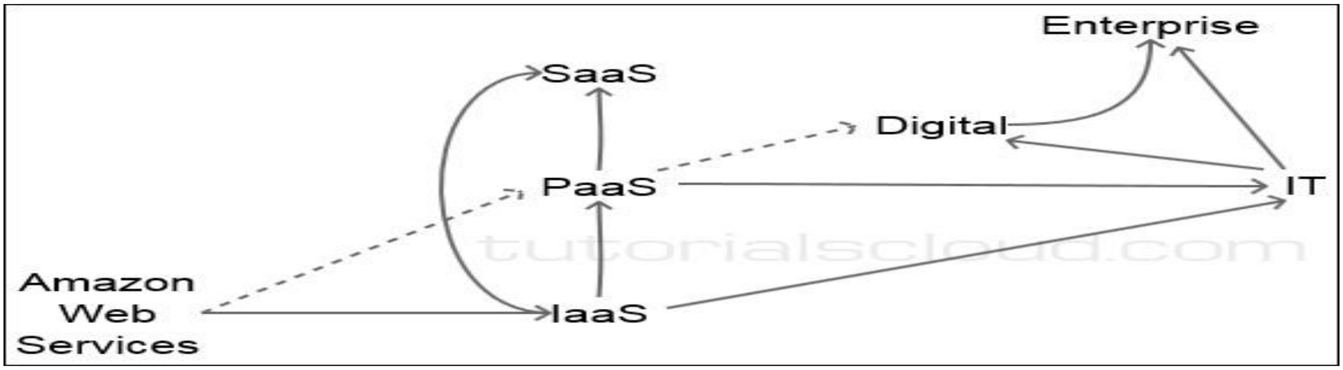
- Cloud IoT Core - Secure device connection and management service for Internet of Things.
- Edge TPU - Purpose-built ASIC designed to run inference at the edge. As of September 2018, this product is in private beta.
- Cloud IoT Edge - Brings AI to the edge computing layer.

API Platform

- Maps Platform - APIs for maps, routes, and places based on Google Maps.
- Apigee API Platform - Lifecycle management platform to design, secure, deploy, monitor, and scale APIs.
- API Monetization - Solution for API providers to create revenue models, reports, payment gateways, and developer portal integrations.
- Developer Portal - Self-service platform for developers to publish and manage APIs.
- API Analytics - Service to analyze API-driven programs through monitoring, measuring, and managing APIs.
- Apigee Sense - Enables API security by identifying and alerting administrators to suspicious API behaviors.
- Cloud Endpoints - An NGINX-based proxy to deploy and manage APIs

V. AWS (AMAZON WEB SERVICES)

- **Amazon Web Services (AWS)** is a collection of **remote computing services (web services)** that together make up a **cloud computing platform**, offered over the Internet by Amazon.com.
- Amazon Web Service is a supplementary of Amazon.com. Amazon has made it possible to develop private virtual servers that can run worldwide via 'Hardware Virtualization' on Xen hypervisor.
- These servers can be provisioned with different types of application software that user might predict along with a range of support services that not only makes cloud-computing applications possible but also make them strong to withstand computation.



AWS is Amazon's umbrella description of all of their web-based technology services.

- Mainly infrastructure services:
  - Amazon Elastic Compute Cloud (EC2)
  - Amazon Simple Storage Service (S3)
  - Amazon Simple Queue Service (SQS)
  - Amazon CloudFront
  - Amazon SimpleDB
  -

Amazon EC2 is hiring

- EC2 provides web services API for provisioning, managing, and deprovioning virtual servers inside amazon cloud.
- Applications anywhere on the Internet can launch a virtual server in the amazon cloud with a single web services call (either REST or SOAP WS call)
- Website: http://aws.amazon.com

EC2

Virtual Servers in the Cloud

- A web service that provides **resizable compute capacity** in the cloud.
- EC2 allows **creating Virtual Machines (VM) on-demand**. Pre-configured **templated Amazon Machine Image (AMI)** can be used get running immediately. Creating and sharing your own AMI is also possible via the **AWS Marketplace.**

Amazon Elastic Block Store (EBS)

- Provides **block level storage** volumes (**1 GB to 1 TB**) for **use with Amazon EC2 instances**.
  - **Multiple volumes** can be mounted to the **same instance**.

    - EBS volumes are **network-attached**, and **persist independently** from the life of an instance.
    - Storage volumes behave like **raw, unformatted block devices**, allowing users to **create a file system** on top of Amazon EBS volumes, or use them in any other way you would use a block device (**like a hard drive**).
- EBS volumes are **placed in a specific Availability Zone**, and can then be **attached to instances also in that same Availability Zone**.
- Each storage volume is **automatically replicated within the same Availability Zone**.
- EBS provides the ability to **create point-in-time snapshots of volumes**, which are **persisted to Amazon S3**.
    - These snapshots can be **used as the starting point for new Amazon EBS volumes**, and protect data for long-term durability.
    - The **same snapshot can be used to instantiate as many volumes** as you wish.
    - These snapshots **can be copied across AWS regions**.

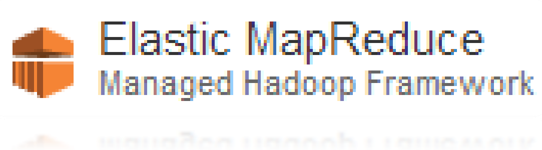
S3
Scalable Storage in the Cloud

- Amazon S3 provides a simple web services interface that can be **used to store and retrieve any amount of data, at any time, from anywhere on the web**.
- Write, read, and delete objects containing from **1 byte to 5 terabytes of data each**. The **number of objects** you can store is **unlimited**.

VPC
Isolated Cloud Resources

- **Amazon VPC** lets you **provision** a **logically isolated section** of the Amazon Web Services (AWS) Cloud.
- You have **complete control** over your virtual networking environment, including:
    - selection of your own **IP address range**,
    - **creation of subnets**, and
    - **Configuration of route tables** and **network gateways**.
- VPC allows **bridging with an onsite IT infrastructure** with an **encrypted VPN connection** with an **extra charge per VPN Connection-hour**.

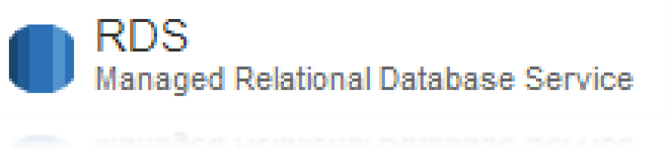
- There is **no additional charge** for using Amazon Virtual Private Cloud, aside from the normal Amazon EC2 usage charges.



Elastic MapReduce
Managed Hadoop Framework

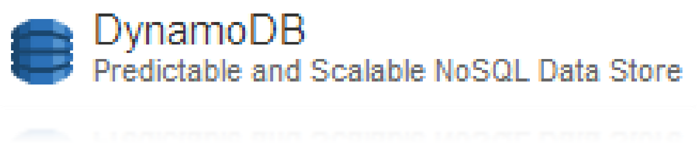
- **Amazon EMR** is a web service that makes it easy to **quickly and cost-effectively process vast amounts of data** using **Hadoop**.
- Amazon EMR **distribute** the **data and processing** across a resizable cluster of Amazon **EC2 instances**.

.



RDS
Managed Relational Database Service

- **Amazon RDS** is a web service that makes it easy to set up, operate, and scale a **relational database in the cloud**.
- Amazon RDS gives access to the capabilities of a **familiar MySQL, Oracle or Microsoft SQL Server database engine.**
  - Code, applications, and tools already **used with existing databases can be used with RDS**.



DynamoDB
Predictable and Scalable NoSQL Data Store

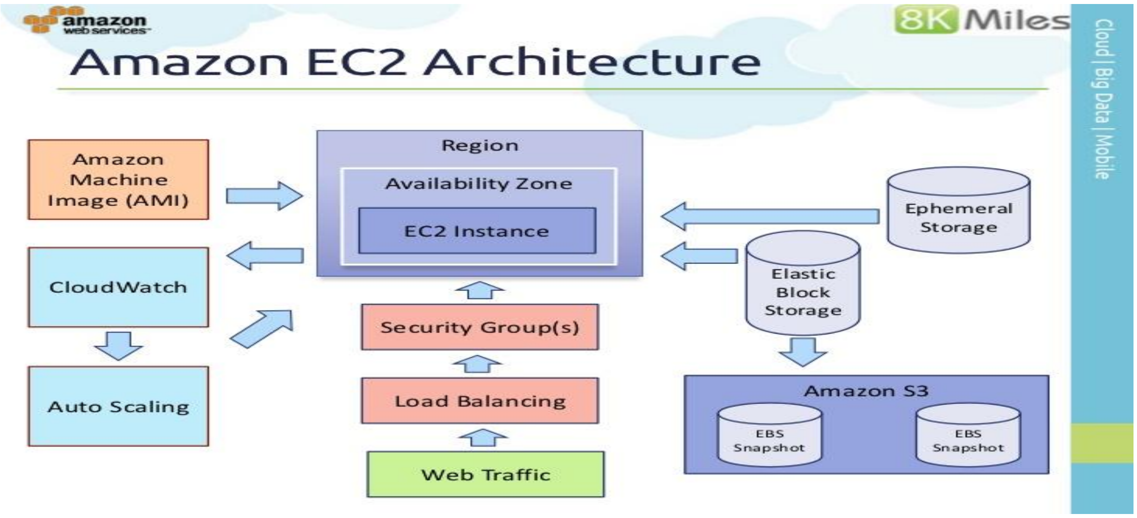
- **DynamoDB** is a fast, fully managed **NoSQL database service** that makes it simple and cost-effective to store and retrieve any amount of data, and serve any level of request traffic.
- All data items are stored on **Solid State Drives (SSDs)**, and are **replicated** across **3 Availability Zones** for high availability and durability.
- DynamoDB**tables do not have fixed schemas**, and each item may have a **different number of attributes**.
- DynamoDB has **no upfront costs** and implements a **pay as you go** plan as a. **a flat hourly rate** based on the **capacity reserved**.

SWF
Workflow Service for Coordinating Application Components

- **Amazon SWF** is a **task coordination** and **state management service** for cloud applications.

- Using Amazon SWF, you **structure** the various **processing steps** in an application that **runs across one or more machines** as a set of "**tasks**."
- Amazon SWF **manages dependencies** between the tasks, **schedules** the tasks for execution, and runs any logic that needs to be **executed in parallel**.
- The service also **tracks** the tasks' **progress**.

AMAZON EC2 ARCHITECTURE:


amazon web services
8K Miles
Cloud | Big Data | Mobile
Amazon EC2 Architecture
Amazon Machine Image (AMI)
CloudWatch
Auto Scaling
Region
Availability Zone
EC2 Instance
Security Group(s)
Load Balancing
Web Traffic
Ephemeral Storage
Elastic Block Storage
Amazon S3
EBS Snapshot
EBS Snapshot