**1.Web Application/web site**

**2. Need Of web Application**

**3. Web Application Architecture.**

**4.Web Component**

**5. Web Server**

**6. Application Server**

**1.Web Application/Web Site :-** The web application is collection of web pages. Each web page contains the information/data . There are two types of web pages.

a. static web pages/Passive web pages.

b. Dynamic web pages/Active web pages.

* 1. Static web page:- The content of this web page will remain same for all the request irrespective of request generation time or input values to request.

Example:-

1. About us page
2. Wiki pedia page
3. Terms & conditions page.
   1. Dynamic Web page:- The content of this web page will be changed based on input values to request or time of request generation. On Dynamic web page,

We see some fixed/constant content and dynamic content which is changed time to time or based on input values.

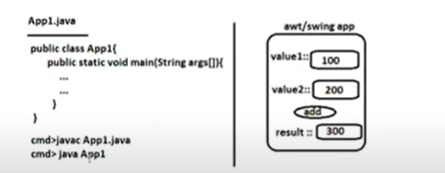
Examples:-

1. Any login page:- when user give wrong credentials(inputs) ,we see the error message dynamically in red color on login page.
2. Stock market trading web site web page.
3. Live IPL game scoring web page.

Note:- Website/web application is collection of static web pages or dynamic web pages.

1. **Need of Web Application:-**
2. The data and logic of standard alone application is specific to one computer and standard alone application can be operated by one person at a time.

Example:- class with main(-) method, awt/Swing app,calculator app, anti-virus s/w …etc.



Data which is generated by standard app or which is stored in database/file.

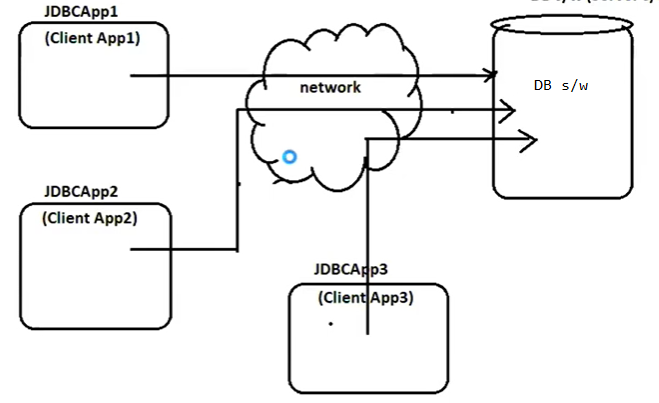
In above example, 300 is data which is generated by awt/swing app.

The logic which process data.

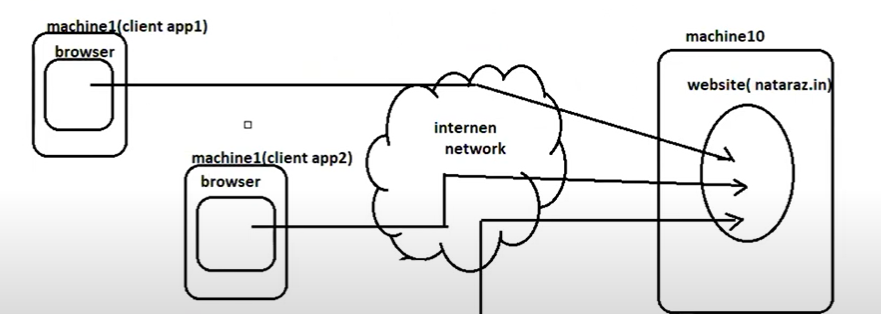
The data and logic of server application can be accessed or manipulated by multiple client apps(Browsers) that are connected to server using network.

Example:-

JDBC app talking to DB s/w.



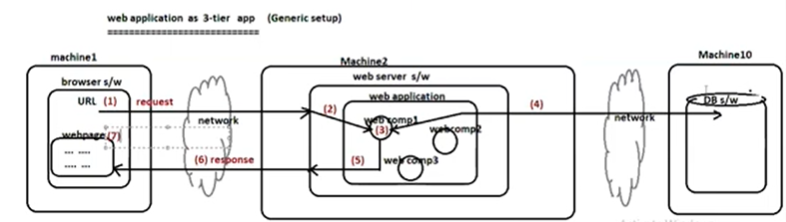
The data and logics of web applications can be accessed from different locations on 24X7 basis using client applications( browser software) through internet network.



**Technologies:-**  servlet,JSP are used to develop the Java based web application.

**FrameWork:-** structs, jsf, spring MVC, or spring BootMVC are used to develop the Java based web applications.

1. **Web Application Architectures:-**
   1. **3-tier Architecture:-**

****

1. End user generates request to web application/web site by placing the URL in address bar of Browser or By clicking the hyper link or By submitting form data.
2. Web server continuously listen to client request by having the daemon thread. After received request by web server, web server handovers the request to web container.

Note:- The web container is combination of both servlet container + JSP container.

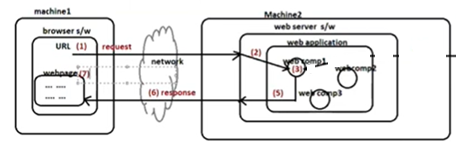
b.1. if request is for static web component( html file/html component or JS file/JS component) then web container collect and places web component code in response. That response is given to web server. The web server and send response to Browser.

b.2. If request is for Dynamic web component(servlet component)then web container maps/links request to appropriate web component. The servlet container will implement the life cycle of servlet component using JRE.

b.3. If request is for Dynamic web component(JSP component) then JSP container maps/links request to appropriate web component. The JSP container will implement the life cycle of JSP component using JRE.

1. In the process of processing request for dynamic web component, if required web component may interact with data for retrieving data from DB or to store the data in DB.
2. Web component send the result to web server.
3. Web server send result to browser as response.
4. Browser displays response in the form of web page.

3.2.2-Tier architecture:-

****

1. End user generates request to web application/web site by placing the URL in address bar of Browser or By clicking the hyper link or By submitting form data.

Note:- The web container is combination of both servlet container + JSP container.

1. Web server continously listen to client request by having the daemon thread. After received request by web server, web server handovers the request to web container.

b.1. if request is for static web component( html file/html component or JS file/JS component) then web container collect and places web component code in response. That response is given to web server. The web server and send response to Browser.

b.2. If request is for Dynamic web component(servlet component)then web container maps/links request to appropriate web component. The servlet container will implement the life cycle of servlet component using JRE.

b.3. If request is for Dynamic web component(JSP component) then JSP container maps/links request to appropriate web component. The JSP container will implement the life cycle of JSP component using JRE.

1. Web component logic is executed to generate the result(response).
2. Web component send the result to web server.
3. Web server send result to browser as response.
4. Browser displays response in the form of web page.
5. **Web Component:-** Generally, web application contains the set of web components. The web component is reusable component. **It has ability to generate the web pages.**

The html file/program, Jsp file/program , servlet file/program , JS file/program ,…etc are technically called web component.

* 1. There are 3 kinds of web components based on generated web pages.

1. Static web component:- From this type of web component, the static web page is generated.

Ex:- html files.

1. Dynamic web component:-From this type of web component, the dynamic web page is generated.

Ex:- Servlet file/program , JSP file/ program…etc

1. Helper web component:- This component does not generates web pages. This component helps to static web component, dynamic web component in the generating the web pages.The helper web component is image file, audio file, video file.
   1. Based on place where web component code is executed, web components are 2 kinds.
2. Client side web components:- These component reside in server machine . These web component code goes to browser through the response from web server. This web component execution happens at client side.

Ex:- html file/program ,JS file/program.

1. Server side web components:- These component resides in server machine and its execution happens at server machine.

Ex:- Servlet file/program, JSP file/program.

1. **Web Server:-**

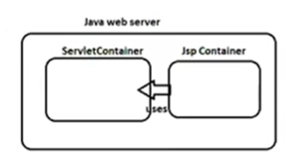
Def-1:The web server is physical machine which provide services to client machines.

Ex:- Network server machine.

Def-2The web server is logical software which provide services to client apps.

Ex:- DB s/w, web server s/w, etc…

Every java based web server like Tomcat have **Servlet Container** and **JSP container.**

****

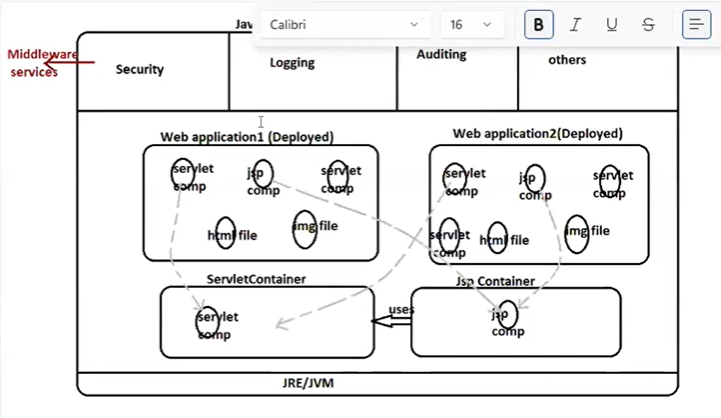
* 1. **Container:-** Container is software program/ application that manages whole life cycle of given web component from birth to death.

**Ex:**

1. Servlet Container:- It manages the servlet component life cycle.
2. JSP container:- It manages the JSP component life cycle.
3. EJB container:- It manages the EJB component life cycle.

**Web container** is combination of only Servlet container and JSP container.

* 1. **Web Server Architecture**:-

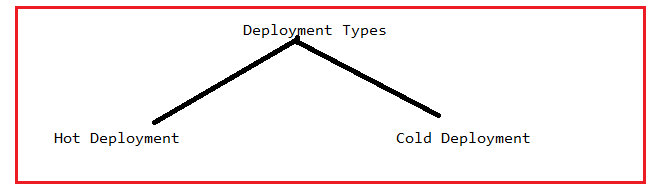
****

**Deployment:-** The process of keeping web application in web server is technically called Deployment.

In above picture, two web applications(web application1 & web application2) were deployed.

There are two types of Deployments:

1. Hot Deployment.
2. Cold Deployment.



**Hot Deployment:-** when web server is in running mode, if web application is deployed in web server , then it is called Hot Deployment.

**Cold Deployment:-** when web server is in stopped mode, if web application is deployed in web server, then it is called cold Deployment.

**Un Deployment:-** The process of removing web application from web server is technically called Un Deployment.

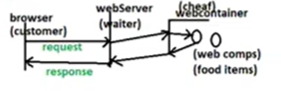
**Middle Ware Services**:- The additional, optional and configurable services that can be applied to web application to make web application as more perfect and accurate is called **middleware services/Add-On services.**

**Examples:-**

1. Security:- It protects web application from unauthorized users.
2. Logging:- It keeps tracks of Application flow.(i.e) which line is being executed, which method is being executed and In which class, control is there ..etc is application flow.
3. Auditing:- Keep the tracks of user activities.(i.e) which activities are done by user.
   * 1. **Web Server Responsibilities:-**
4. Web server continuously listen to client request by having the daemon thread. After received request by web server, web server handovers the request to web container.
5. provide add-on services /Middle ware services to deployed web applications.

c. Receiving result from web container.

d. Web server send result to browser as response.



* + 1. **Web container Responsibilities**:-

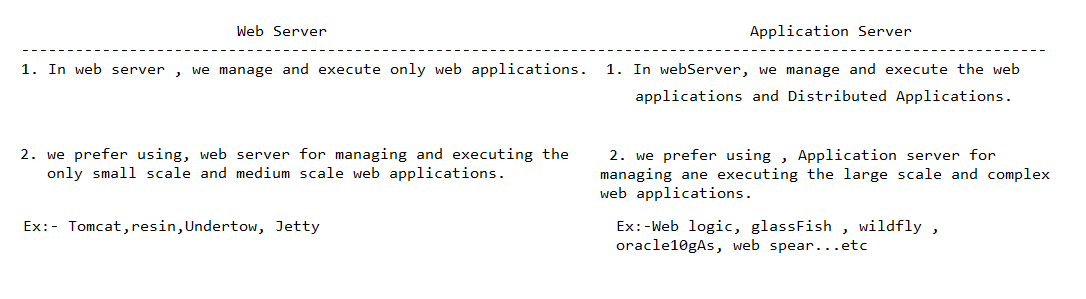
1. Receiving the request from web server.
2. Based on URL, web container maps/links the request to appropriate web component.(This is called routing).
3. Provides the environment required for inter communication between the web components.

(servlet –to- servlet, servlet-to-JSP, JSP-to-JSP ,..etc)

1. Web container does use Garbage collector to destroy the objects of web component Because GC can remove object when they are unreferenced. Where as, web container has to remove the web component object at the end of their life cycle irrespective of whether object is referenced or not. Therefore web container uses its own garbage collector to destroy the web component object.
2. Provide the environment to apply the middle-ware services on deployed web applications.
3. Provide the environment to deploy, un deploy , start , stop and load the web applications.
4. It collect the output of web component and It send the output to web server.
5. **Application Server:-**

The application server is web server++.



****

**Note:-1.All application servers are java based servers.**

**2. some web servers/application servers have their own JREs.**

**EX:- WEB LOGIC, WEB SPEAR…etc**

**3. Some web servers/ application servers does not have their own JREs and they use JRE that is already installed in current computer.**

**Ex:- Tomcat,GlashFish…etc.**