

There are 2 types of Web Applications are there:

1. static webapplication
2. dynamic webapplications

1. static webapplication

Incase of static web application upon receiving the request, the webserver will locate an relevant static page like HTML and dispatches it as an response back to the user. So no matter how many times the enduser has requested the resource by sending the request, always the server will serves the same static content to the enduser. The enduser cannot interact with the application in passing the data asking to perform operation based on it.

2. dynamic webapplication

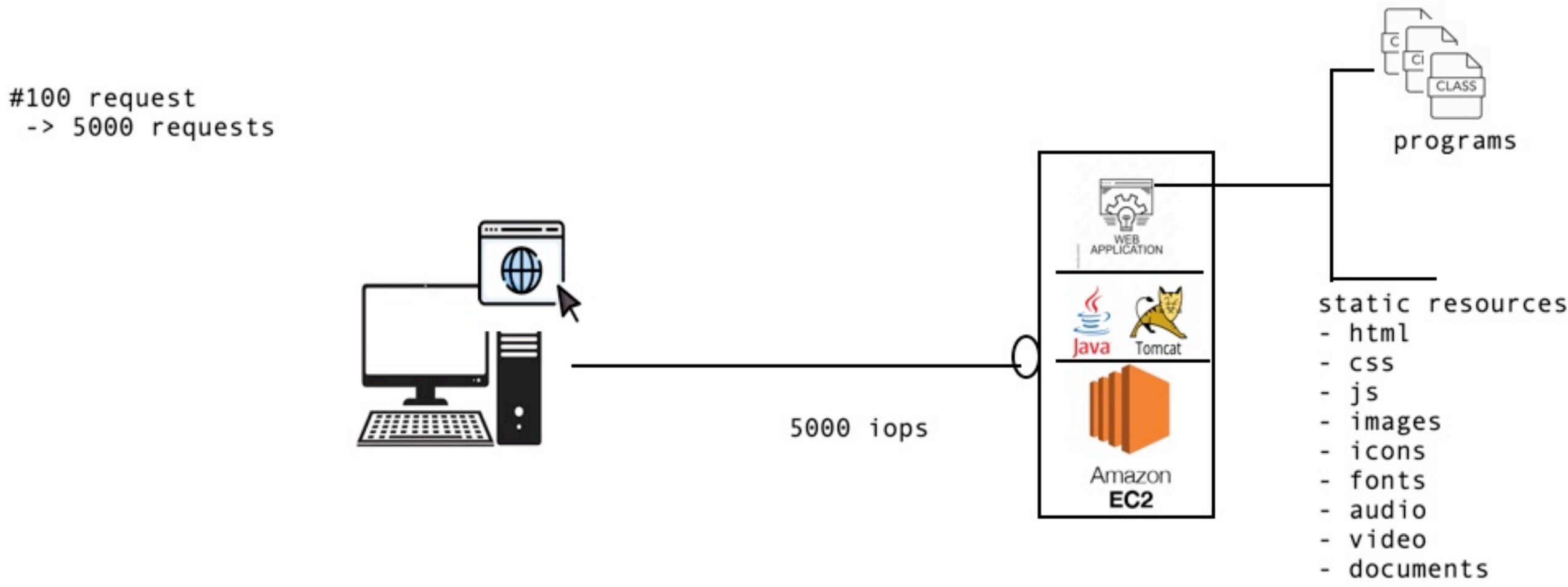
Incase of dynamic web application, the enduser can interact with the application by passing the data, asking it to perform the operation on it. upon the webserver receives the request, it gathers the enduser data and executes an program on the server with the userdata being passed and collects the output of the program and dispatches it as an response back to the user. since the response for the request being send is not fixed and is computed based on the user input data, these applications are called "Dynamic Web Applications".

The dynamic web applications requires dynamic webserver like java: tomcat, .net: IIS etc.

A dynamic web application along with programs it even comprises of static resources as well like

1. HTML pages
 2. css
 3. javascript
 4. images
 5. fonts
 6. icons
- etc

usually these resources are packaged and deployed on the dynamic webserver as below.



When the dynamic webserver received the request, it identifies the resource to whom the request has been sent for, and if the resource is found to be a dynamic program then the server executes the program by passing the user inputs it has gathered and collects the output of the program and dispatches/writes as an response to the client/browser. otherwise if the request is identified for an static resource, the webserver has nothing to do here for that request, it just need to serve appropriate static resource by dispatching it as response back to the user/browser/client over the network

```
index.html
<html>
<head>
  <title>page title</title>
</head>
<body>
  <script type="text/javascript" src="common.js"/>
  <script type="text/javascript" src="validations.js"/>
  <script type="text/javascript" src="jquery.js"/>
  
</body>
</html>
```

when the client sends the request to the above index.html to the webserver (static/dyanmic) the webserver serves only the index.html page to the browser back. while the browser is rendering the index.html contents by parsing it, when it encounters the script tag referencing an static javascript or an img tag referencing an image, then browser makes an another request to the server asking for the static resource.

It this way while rendering a typcial webpage in a browser, it would span around 50 - 100+ requests asking for serving the static resources per each page back to the server due to which we run into below issues

1. the number of network calls being send to the server for serving the static resources increases, due to which network congestion will takes place
2. The bandwidth capacity of the webserver or the machine will be occupied in serving the static resources only, there by it cannot accept dynamic requests coming to the application
3. The server might reach to the max connection limit to accept the client requests

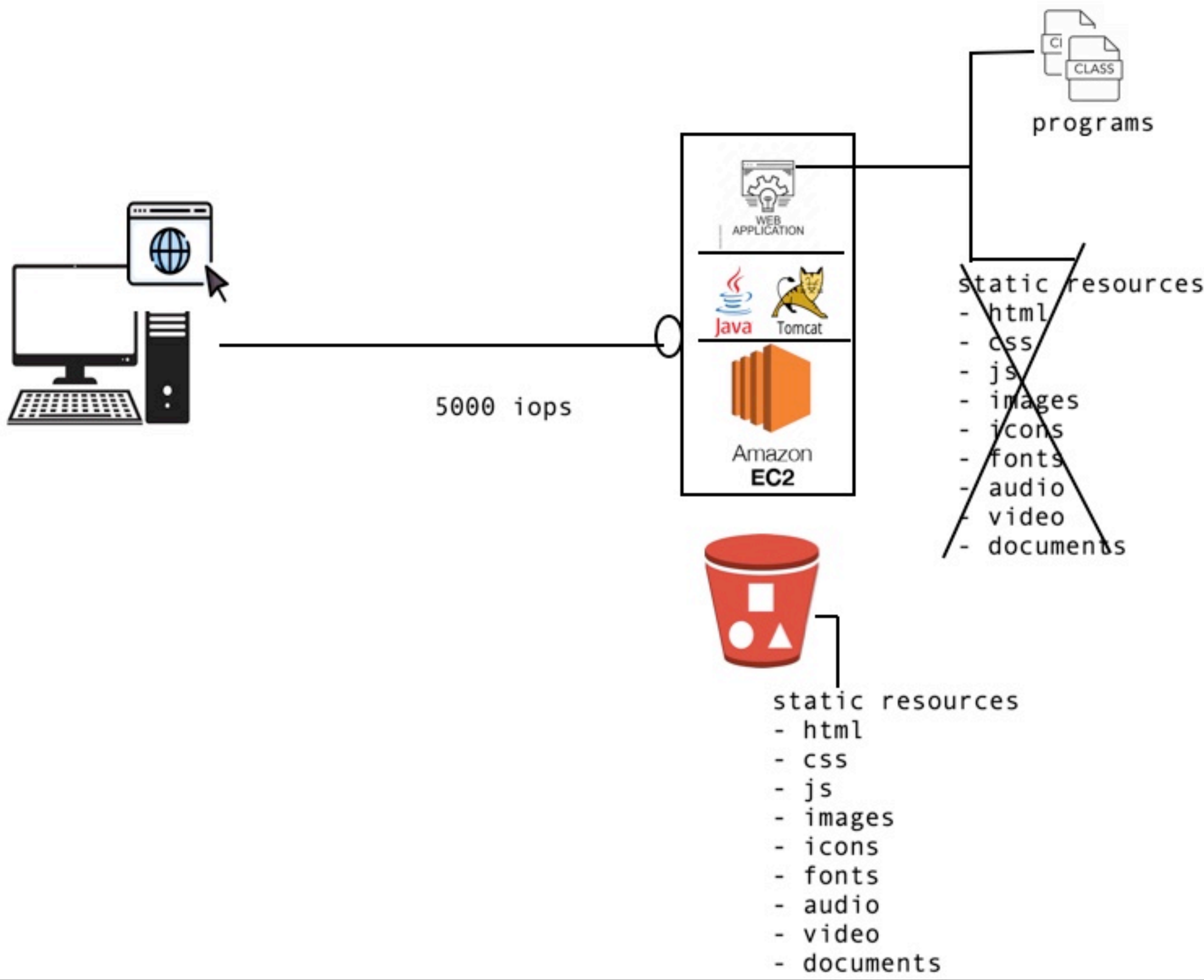
due to high network congestion

The webserver cannot accept the requests even though it has plenty of cpu/memory available and the client will be experiencing the latency in loading the pages.

At this moment it seems that we need to scale-out the application to resolve the problem, but that increases the cost of running the application without adding an business value

interms of monetary

Instead of this, externalize the static resources of your application by placing them in S3 storage.



In hosting an static website we can use S3 storage aswell.

