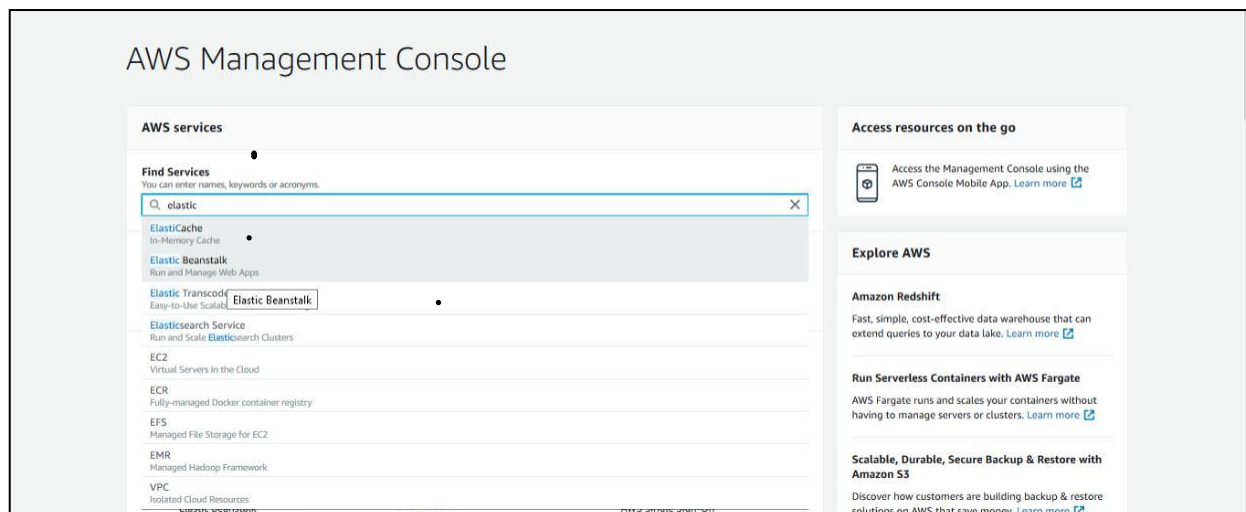


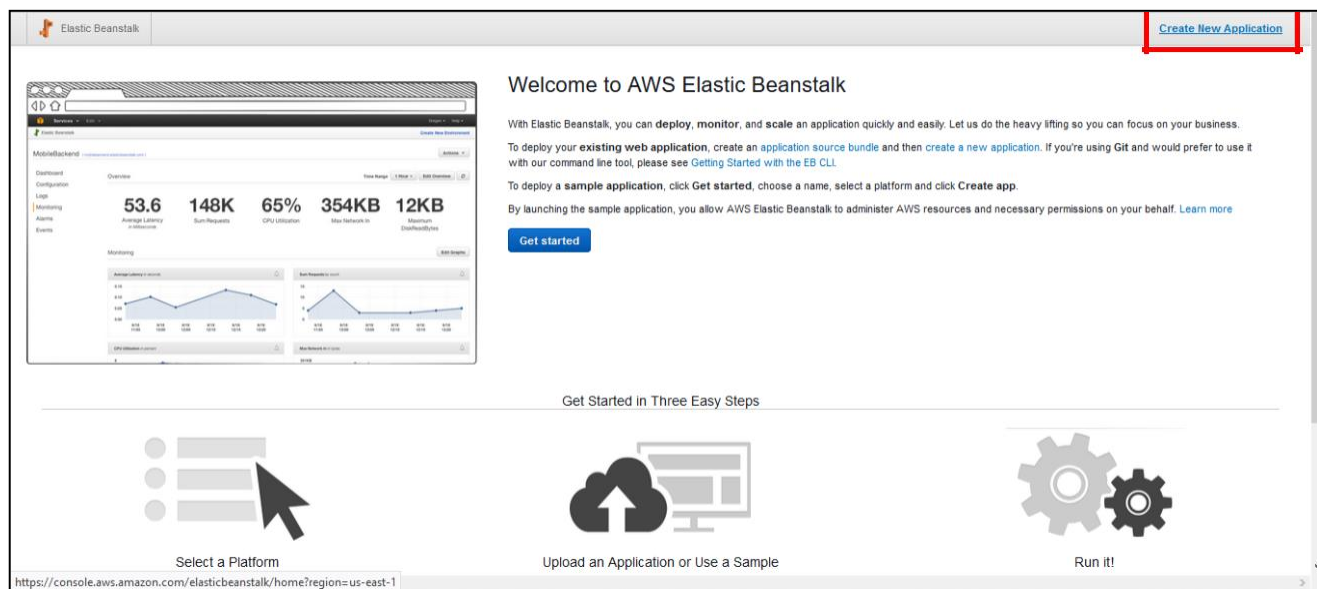
Practical 7: Deploy a Web Application on AWS.

Solution :

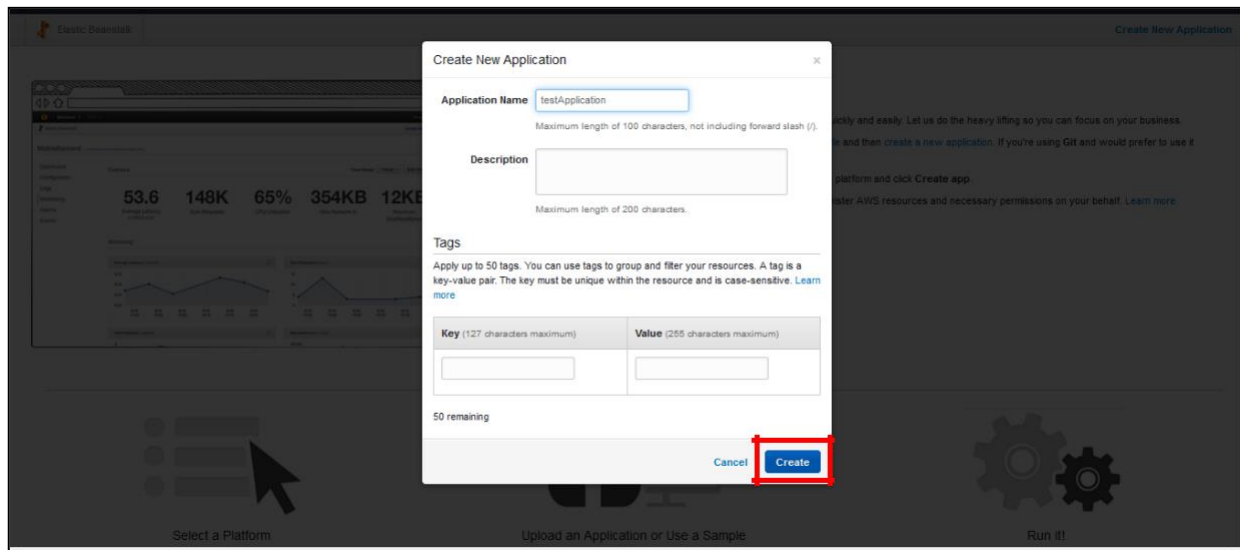
Step 1. Go to AWS dashboard, search and select **Elastic BeanStalk**



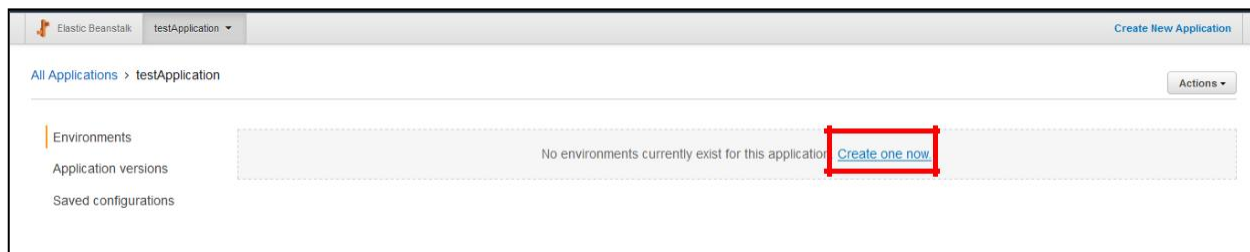
Step 2. Select Create New Application at Elastic Beanstalk Dashboard



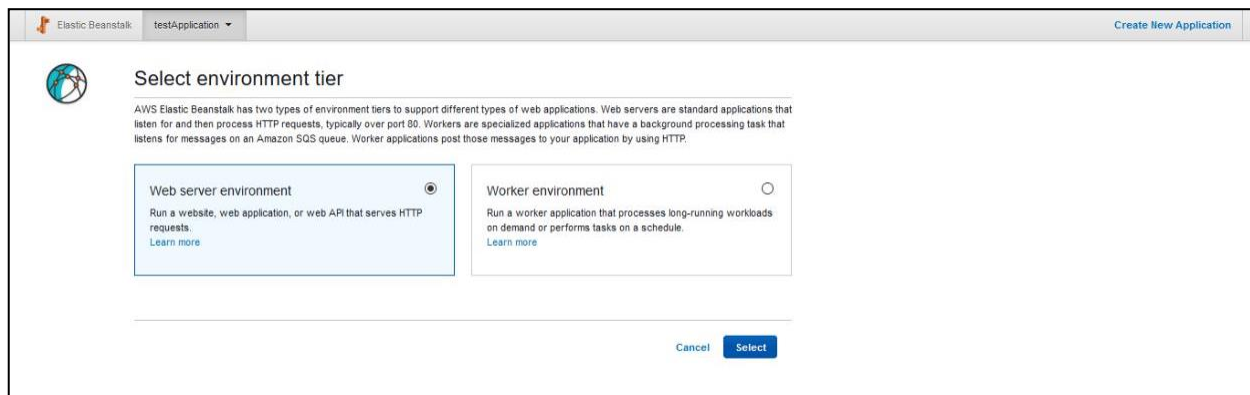
Step 3. Write any name in Application name here we have mentioned **testApplication**, then select create option.



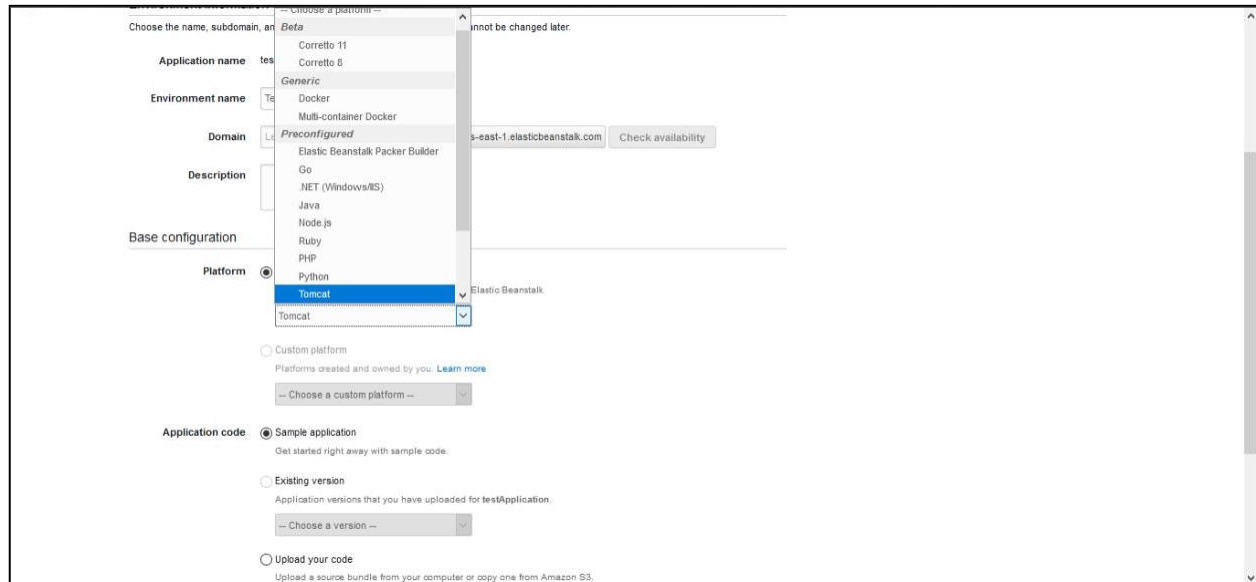
Step 4. Now the application dashboard will open. In the dashboard select **Create one now**



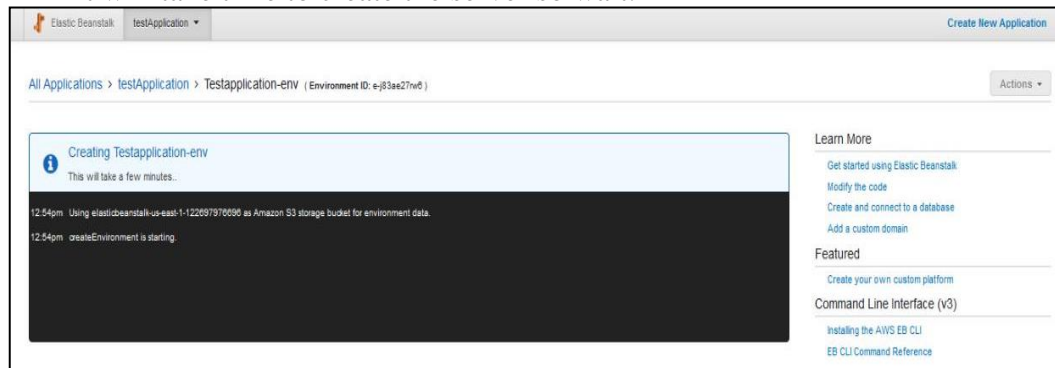
Step 5. Select Web Server Environment and proceed



Step 6. As we are deploying a java web application select **tomcat** in platform.
Continue further by selecting **Create Environment**

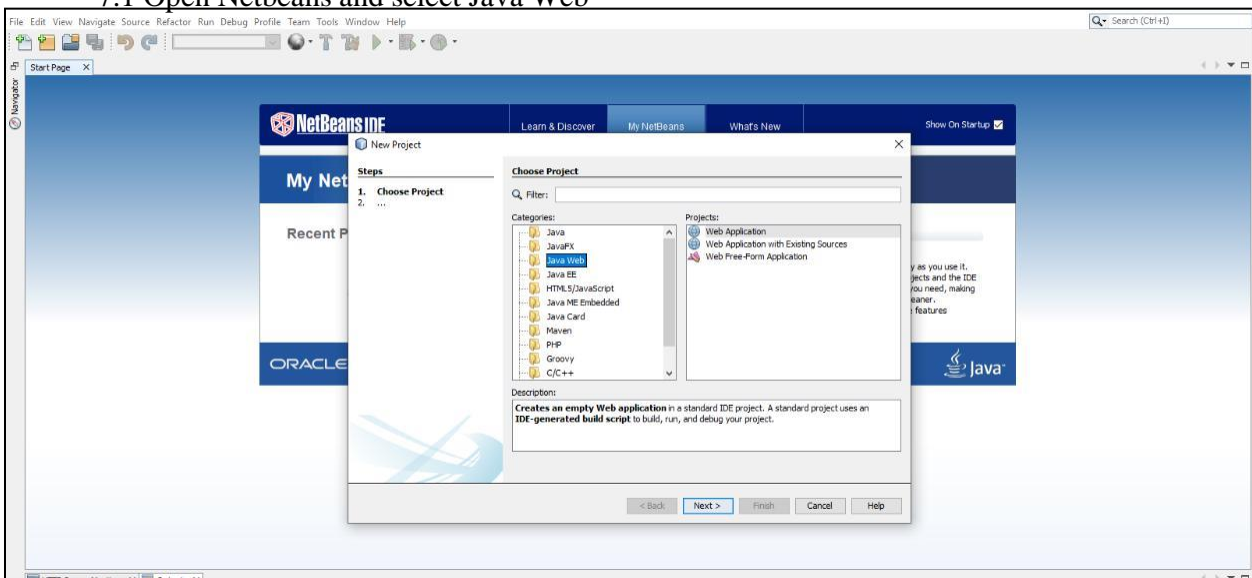


- It will take time to create the server so wait.

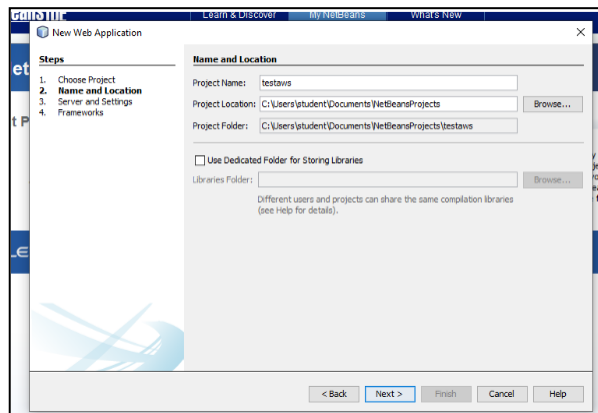


Step 7. Create a Web Application for AWS in Netbeans

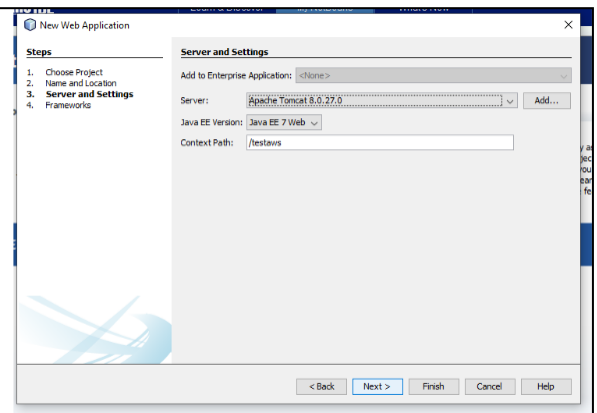
7.1 Open Netbeans and select Java Web



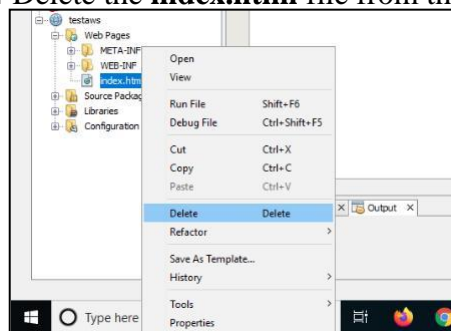
7.2 Name the app : **testaws**



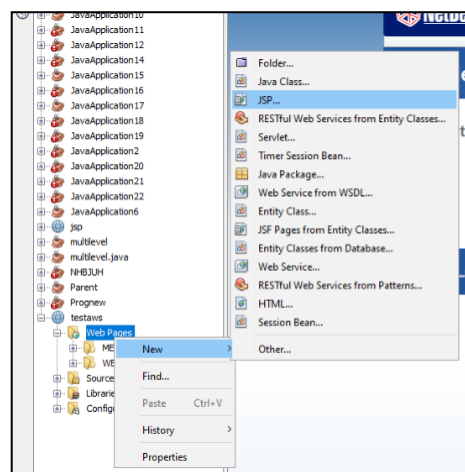
7.3 In server select **Apache tomcat** & finish



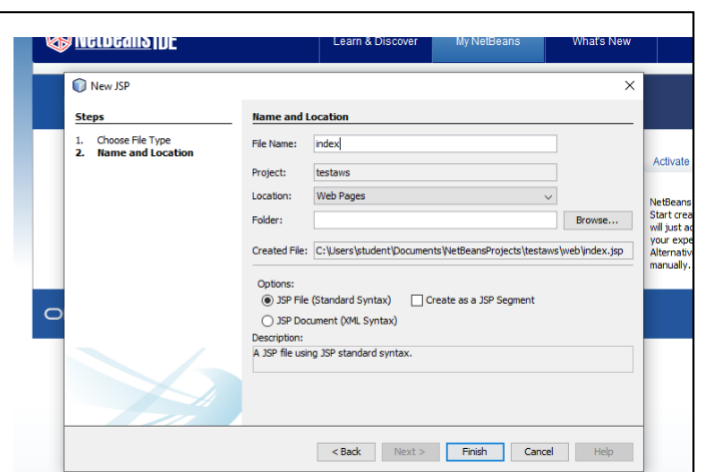
7.4 Delete the **index.html** file from the package



7.5 In the webpage create a new jsp File



7.6 Name it as index

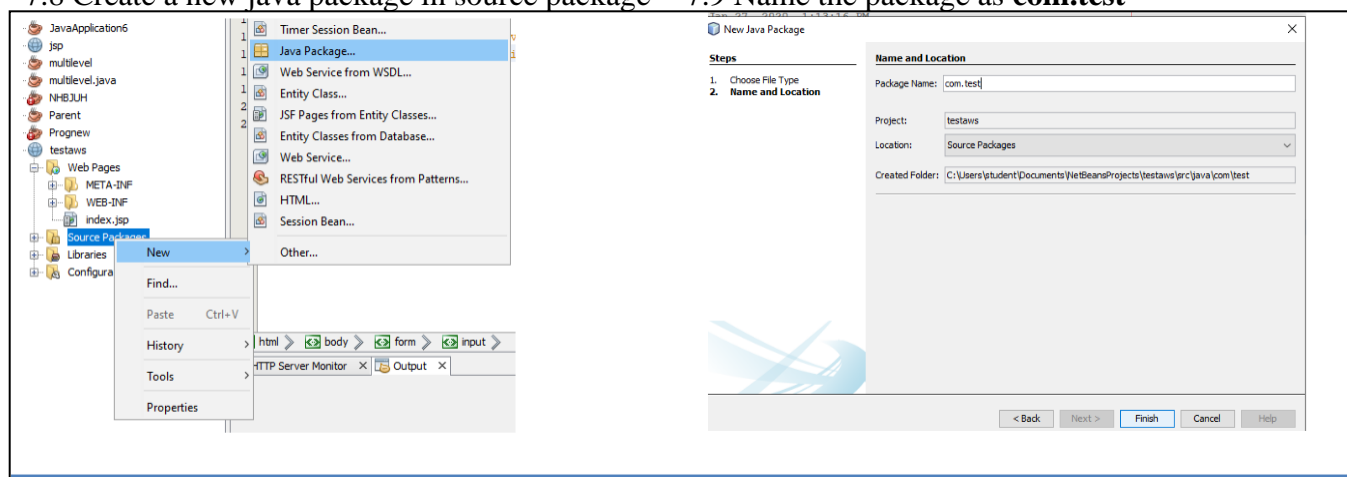


7.7 Create a form tag after hello world line

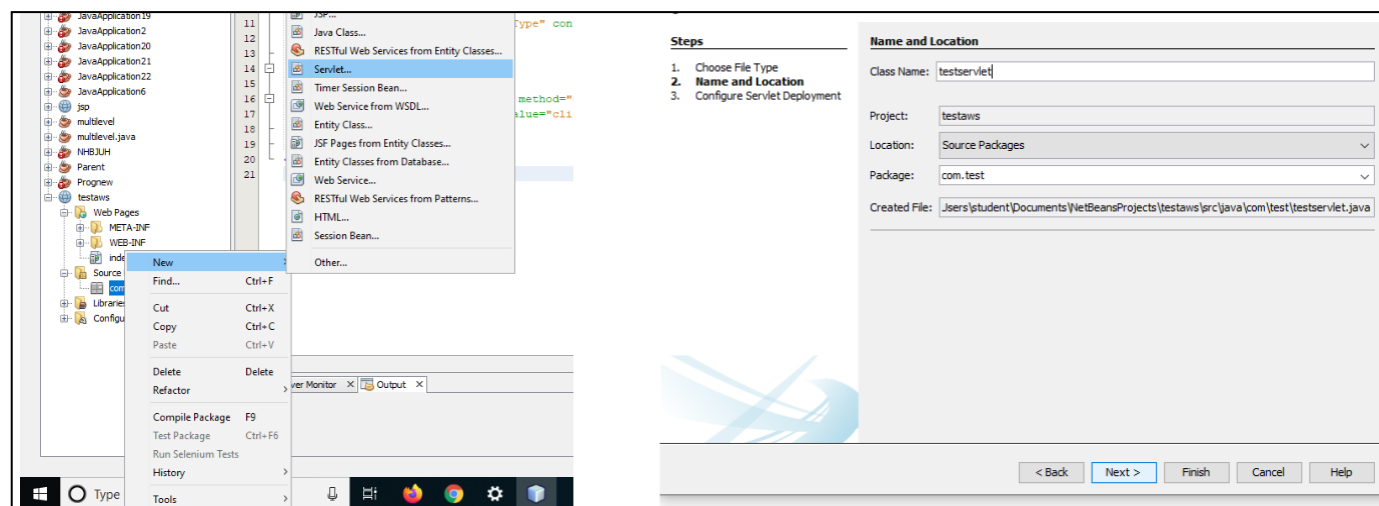
```

6
7 <%page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <h1>Hello World!</h1>
16 <form action="testServlet" method="get">
17 <input type="submit" value="click here">
18 </form>
19 </body>
20 </html>
21
  
```

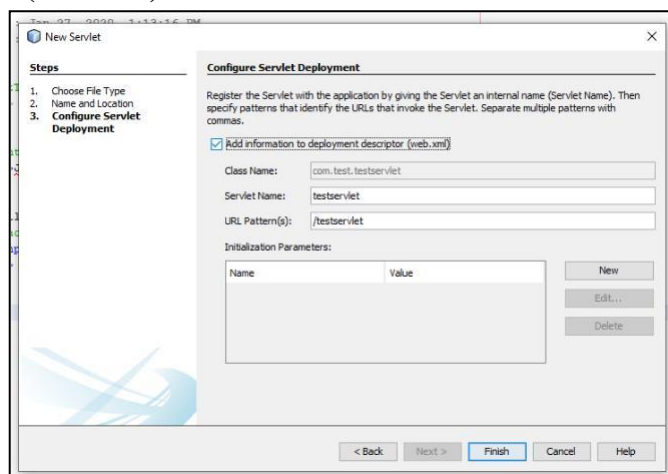
7.8 Create a new java package in source package 7.9 Name the package as **com.test**



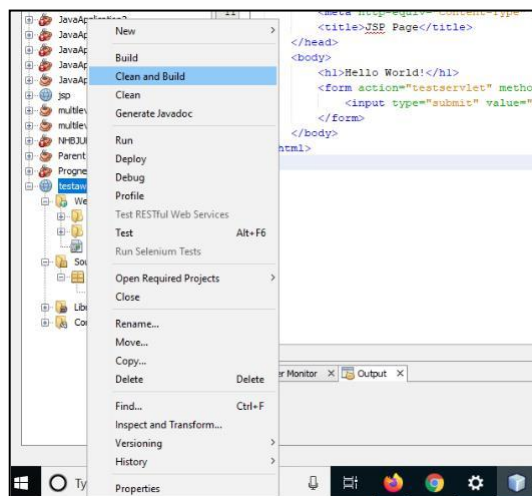
7.10 Create a new servlet in **com.test** package 7.11 Name it as **testServlet** and proceed



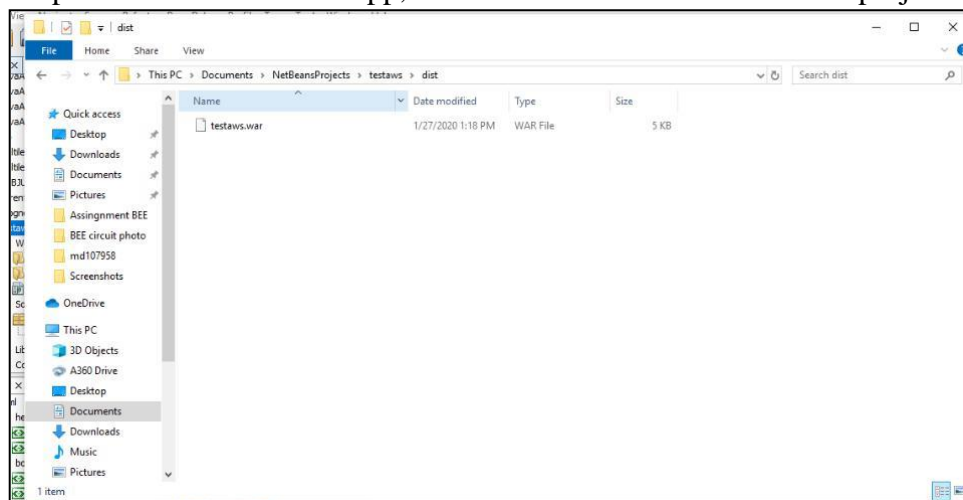
7.12 In the Configure Servlet Deployment check **Add information to deployment descriptor (web.xml)**



7.13 Now select the Web App and select option **Clean and Build**

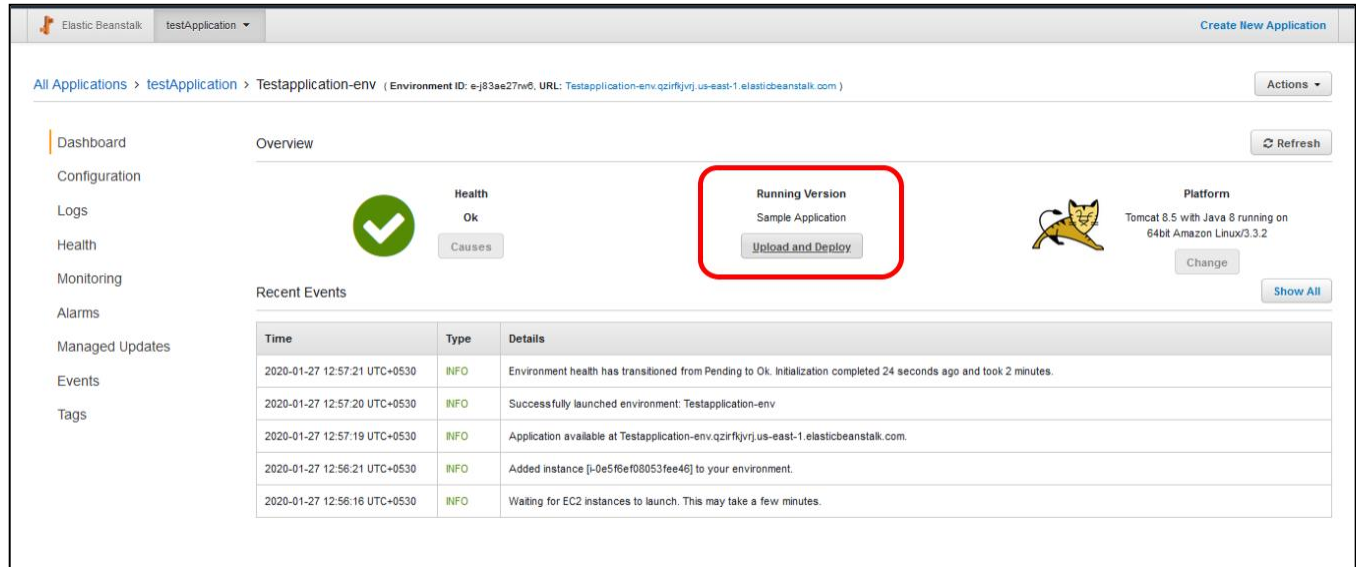


Step 8. Since we build the app, there will be a **.war** file created in project **dist** folder.



Step 9. Upload and deploy

9.1 Go to the AWS dashboard and select the application which you created. Select Upload and Deploy option.



Dashboard

Configuration

Logs

Health

Monitoring

Alarms

Managed Updates

Events

Tags

Overview

Health: Ok

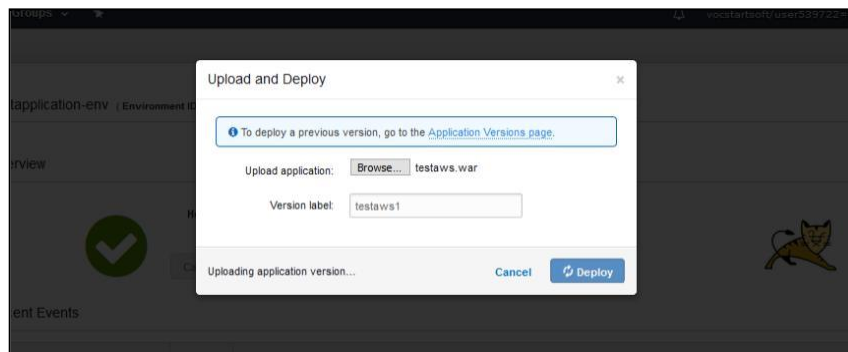
Running Version: Sample Application

Platform: Tomcat 8.5 with Java 8 running on 64bit Amazon Linux/3.3.2

Recent Events

Time	Type	Details
2020-01-27 12:57:21 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 24 seconds ago and took 2 minutes.
2020-01-27 12:57:20 UTC+0530	INFO	Successfully launched environment: Testapplication-env
2020-01-27 12:57:19 UTC+0530	INFO	Application available at Testapplication-env.qzirkfjvrij.us-east-1.elasticbeanstalk.com.
2020-01-27 12:56:21 UTC+0530	INFO	Added instance [i-0e5f6e08053fee46] to your environment.
2020-01-27 12:56:16 UTC+0530	INFO	Waiting for EC2 instances to launch. This may take a few minutes.

9.2 In the location of the **.war** file and upload it. Name it as **testaws1** or whatever you want. Deploy it after the war file is uploaded.



Upload and Deploy

To deploy a previous version, go to the [Application Versions page](#).

Upload application:

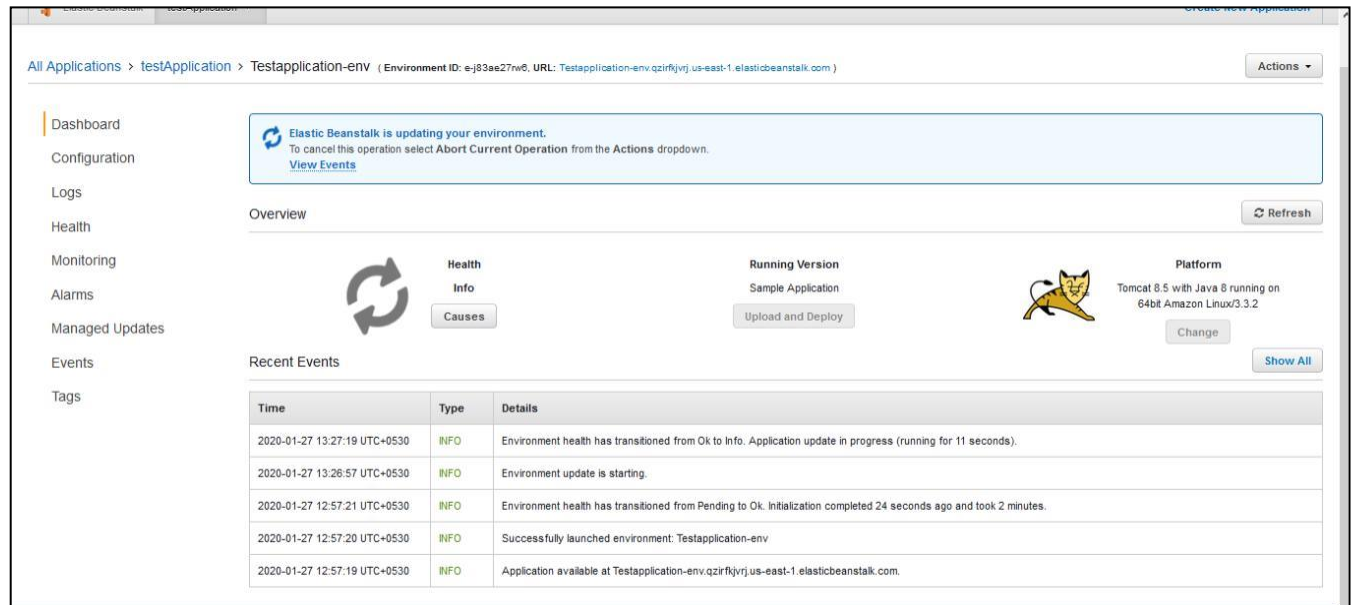
Version label:

Uploading application version...

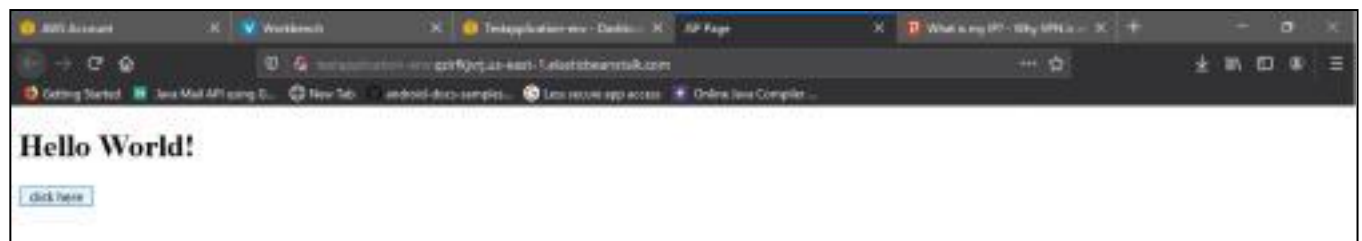
Cancel Deploy

Step 10. Checking the app in online or not ?

10.1 When the app is successfully deployed, the pop-up will be gone and a link will be generated.
Click on that link and a new tab will be opened, with the deployed application.



10.2 As we can see the application is working. We click here button which we created.



10.3 As we click on the click here button we are redirected to testservlet. Hence the application is running.

