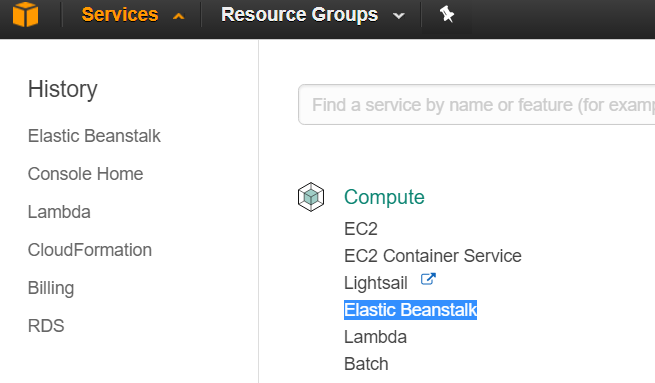
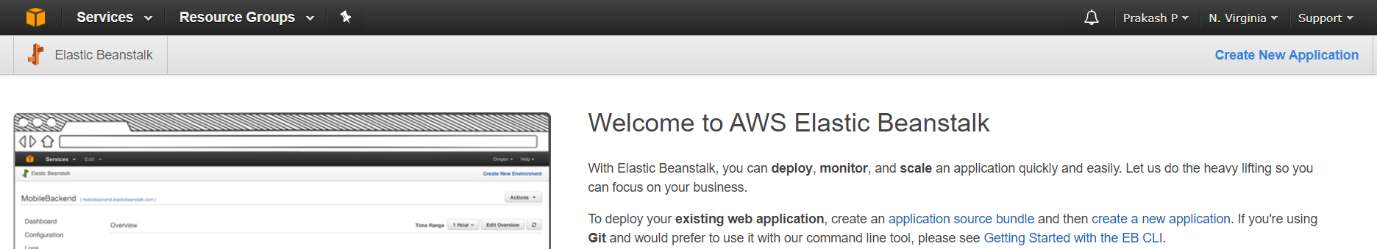
AWS Elastic Beanstalk

* Amazon Web Services (AWS) comprises dozens of services, each of which exposes an area of functionality.
* While the variety of services offers flexibility for how you want to manage your AWS infrastructure, it can be challenging to figure out which services to use and how to provision them.
* With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications.
* AWS Elastic Beanstalk reduces management complexity without restricting choice or control.
* You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.
* Elastic Beanstalk uses highly reliable and scalable services.
* Elastic Beanstalk supports applications developed in Java, PHP, .NET, Node.js, Python, and Ruby, as well as different container types for each language.

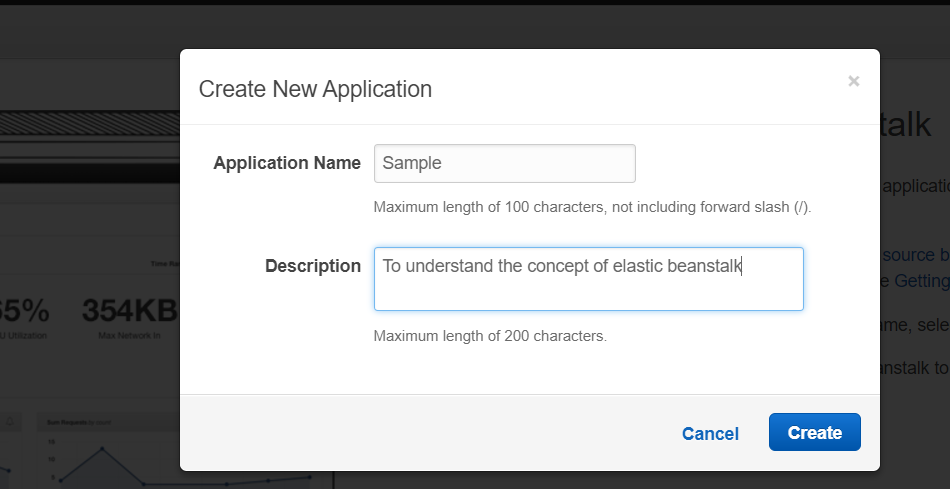
Step 1: Select Elastic Beanstalk from compute



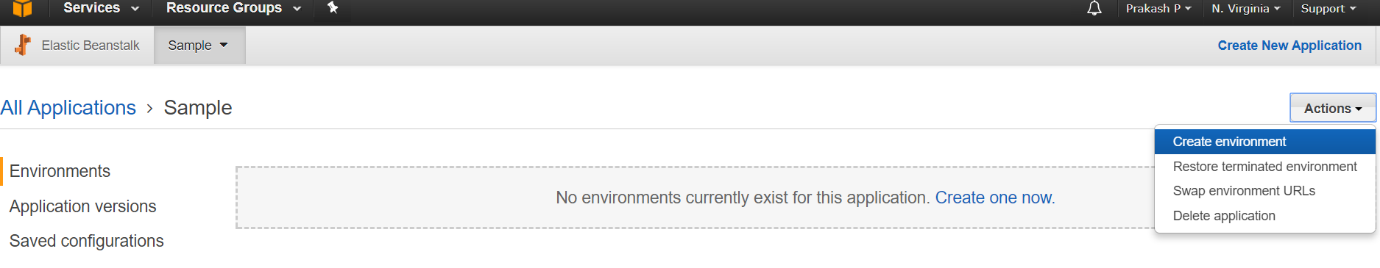
Step 2: Click Create New Application in the Elastic Beanstalk dashboard.

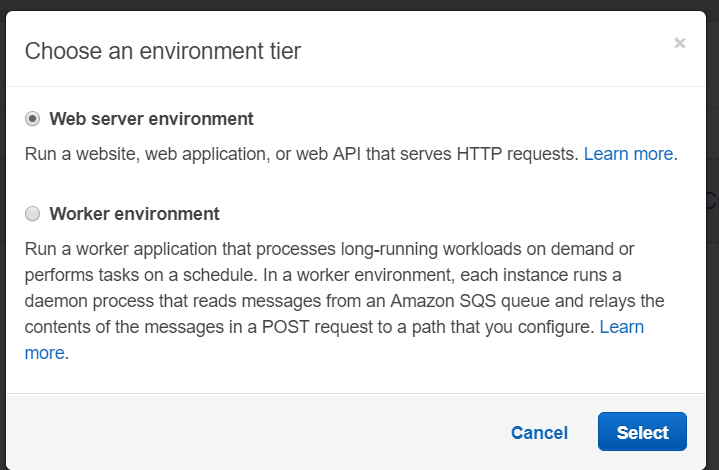


Step 3: Enter the application name ( Eg. Sample) and click the create button. It will open an application window without any environment.

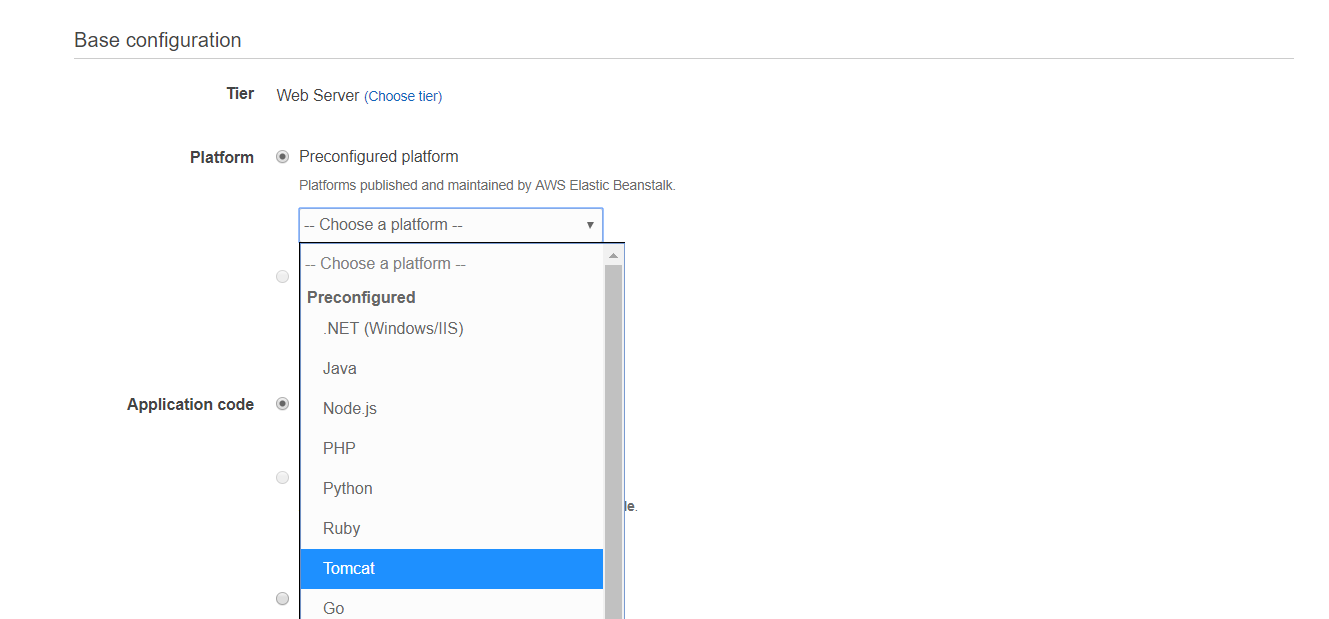


Step 4 : Go to Actions and select the Create environment and choose web server environment and click select.





Step 5: It opens Create a new environment page, Give some environment and domain name.

In the base configuration, Choose >> Platform as >> Tomcat

And the Application code as sample application and click the Create Environment button.

It will take some time to create the environment.

Step 6 : Create a new java web application using Tomcat as server in NetBeans IDE.

Step 7: To Change the newjsp.jsp as a welcome page instead of index.html , please do the following steps:

In Netbeans by default, if you create a project with no added frameworks, no deployment descriptor(web.xml) is provided.

To change it, right click on the project and select New>Other>web>Standard Deployment Descriptor(web.xml)

Now edit the web.xml and set

<welcome-file-list>

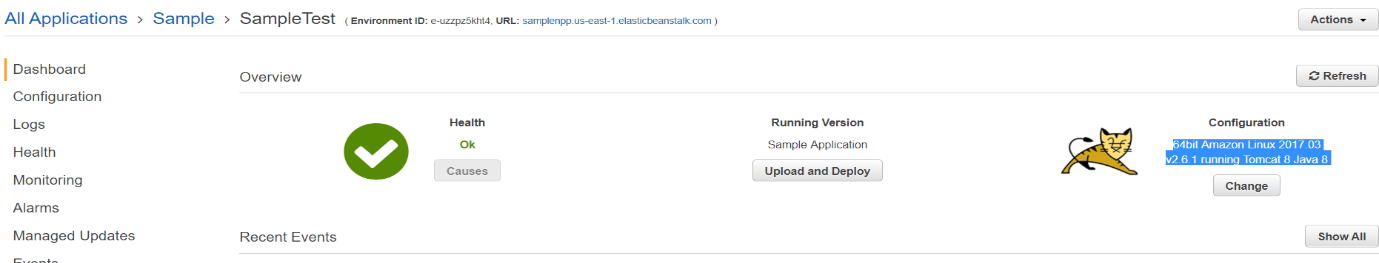
<welcome-file>newjsp.jsp</welcome-file>

</welcome-file-list>

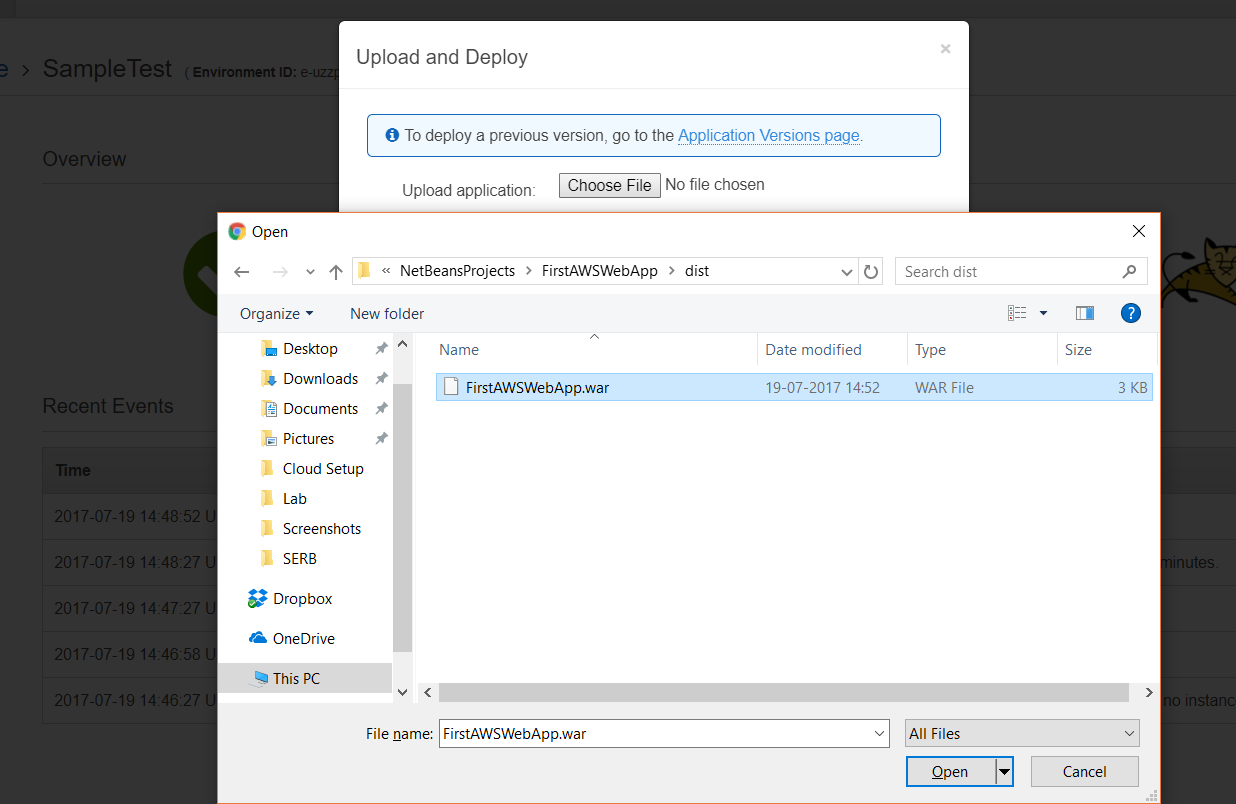
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 8: Clean and build. Finally run the project, it will open newjsp.jsp as starting page, instead of index.html.

Step 9: Check you environment status in AWS console, it is ready to deploy the web application developed from the step 6 to 8.



Step 10 : Click the upload and deploy button and select the war file from the location ( Project location) C:\Users\..................... \NetBeansProjects\FirstAWSWebApp\dist and click the deploy .



Step 11: Once the deployment was successful, Copy the URL from the URL >> and copy the link into the browser.

