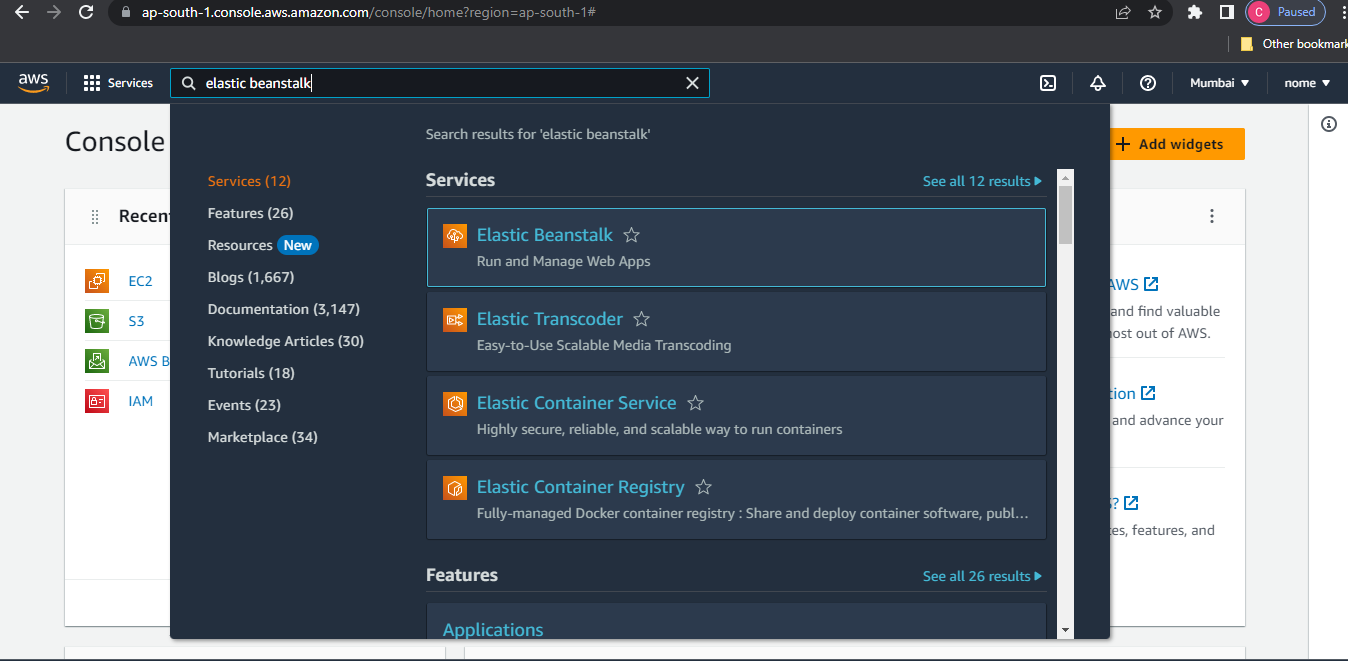
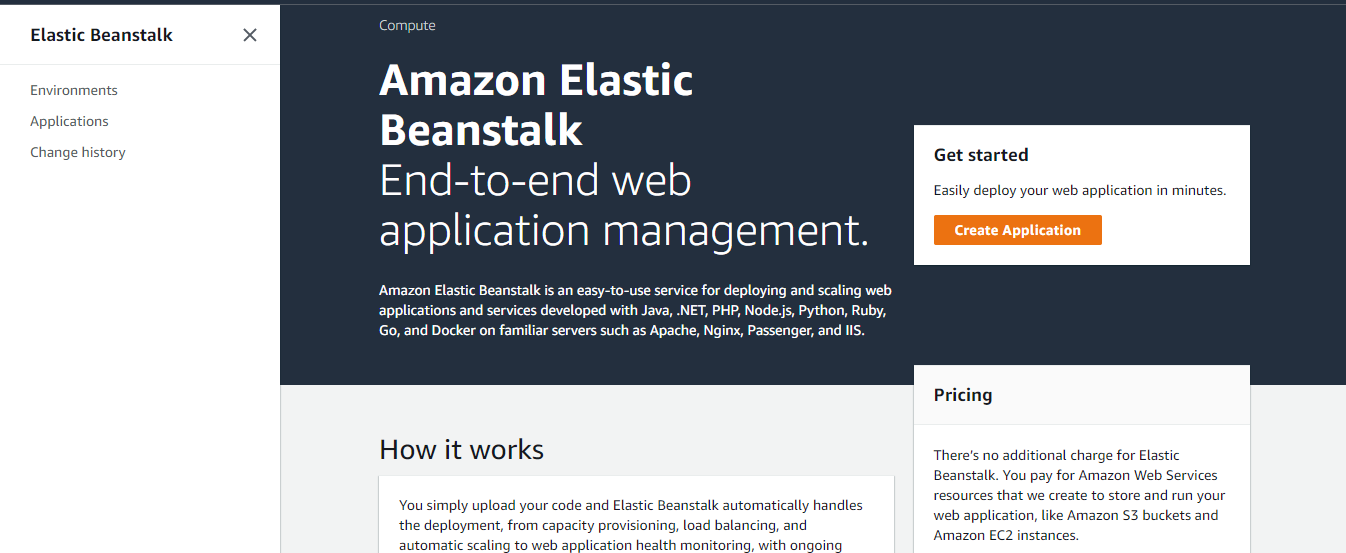
**Experiment-6:Deploy an application using AWS Elastic Beanstalk (PaaS)**

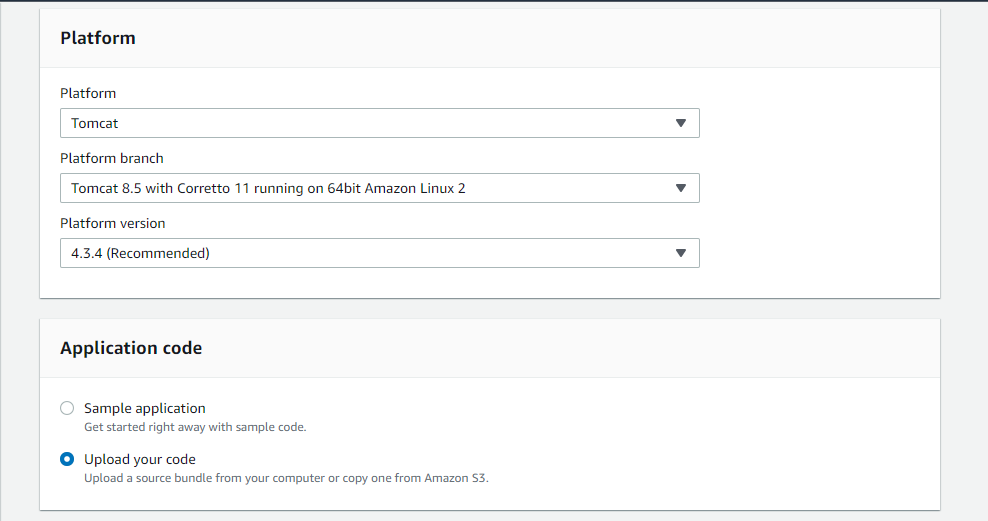
**Step 1:** Open The Elastic Beanstalk Service in the AWS management console.

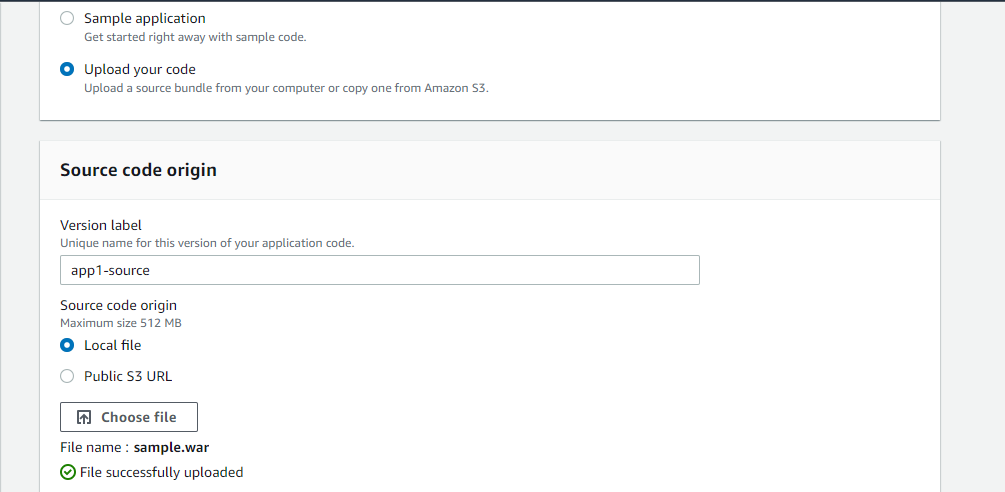


**Step 2:** Create a NewApplication.

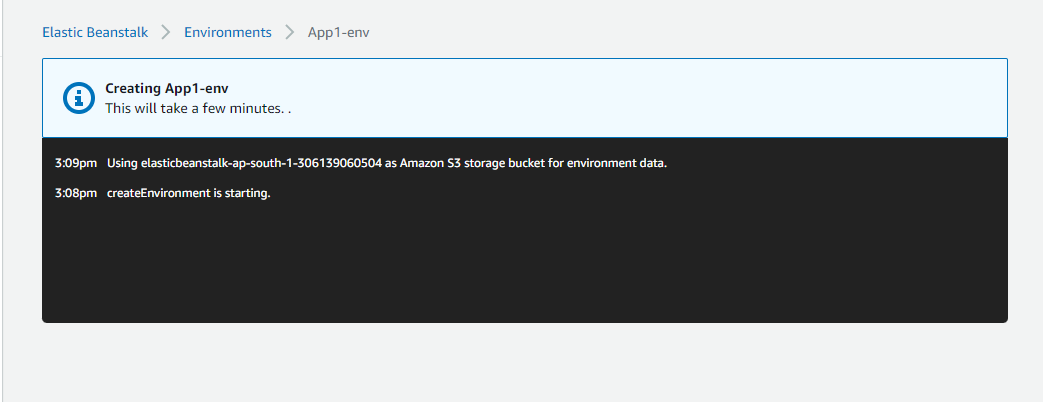


**Step 3:**Select the web server environment and fill the details about your applications and click on create environment.

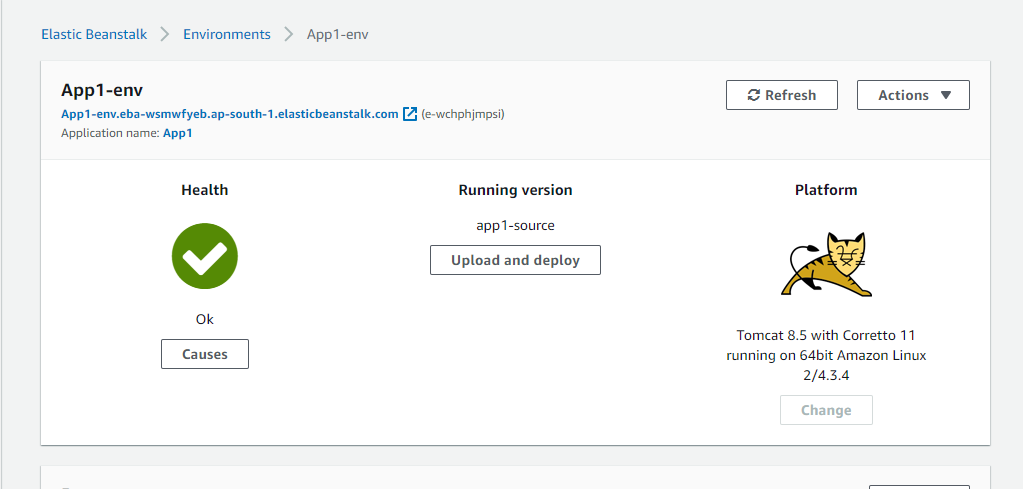


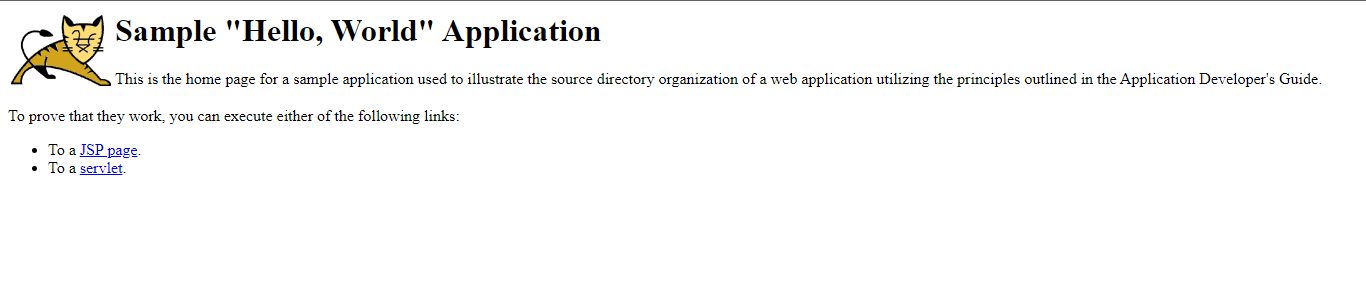


**Step 4:**It will start the process of creating the environment and will show each service it is adding to the environment.



**Step 5:** After completing it will take you to the environment page where it will show the health check of our application as OK.







**QUESTIONS**

**Ques 1:** List the features and advantages of AWS Elastic Beanstalk?

**Ans.** AWS Elastic Beanstalk is a fully-managed platform as a service (PaaS) offered by Amazon Web Services. Here are some of the features and advantages of using AWS Elastic Beanstalk:

**Features:**

1. Multi-Platform Support: AWS Elastic Beanstalk supports multiple programming languages and platforms including Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker.
2. Scalability: AWS Elastic Beanstalk allows you to easily scale your application up or down depending on the traffic.
3. Easy Deployment: You can deploy your applications easily with AWS Elastic Beanstalk, and the platform handles all the heavy lifting for you.
4. Auto-scaling: AWS Elastic Beanstalk can automatically scale up or down based on the load.
5. Monitoring: AWS Elastic Beanstalk provides detailed monitoring of your application performance and can send alerts when there are issues.

**Advantages:**

1. Easy to use: AWS Elastic Beanstalk is easy to use, and you don't need to have advanced knowledge of AWS services.
2. Cost-effective: You only pay for the resources that your application consumes, and there are no upfront costs or long-term commitments.
3. Faster time to market: With AWS Elastic Beanstalk, you can quickly deploy your application and focus on developing your product instead of managing infrastructure.
4. High Availability: AWS Elastic Beanstalk offers high availability and automatic failover, which ensures that your application is always up and running.
5. Easy to integrate: AWS Elastic Beanstalk can be easily integrated with other AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon SQS.

**Ques 2**: Explain in detail Web Server and Worker Environment of Elastic Beanstalk.

**Ans.** Elastic Beanstalk is a managed service offered by Amazon Web Services (AWS) that makes it easier to deploy and scale web applications. Elastic Beanstalk abstracts the underlying infrastructure and automates many of the tasks involved in deploying and managing applications, making it an ideal solution for developers who want to focus on writing code rather than worrying about infrastructure.

Elastic Beanstalk provides two key components for running web applications: a web server environment and a worker environment. In this answer, we'll explain each of these components in detail.

**Web Server Environment**

A web server environment in Elastic Beanstalk is a fully managed, highly available environment that is designed to run web applications. Elastic Beanstalk supports several programming languages and frameworks, including Java, .NET, Node.js, PHP, Python, Ruby, and Go.

When you create a web server environment in Elastic Beanstalk, the service automatically provisions and configures the resources necessary to run your application, including EC2 instances, load balancers, auto-scaling groups, and security groups. Elastic Beanstalk also sets up monitoring and logging for your application, which makes it easier to identify and troubleshoot issues.

Elastic Beanstalk supports several deployment options, including rolling updates, blue-green deployments, and canary deployments. With rolling updates, Elastic Beanstalk deploys new versions of your application one instance at a time, while still serving traffic from the remaining instances. Blue-green deployments involve creating a new environment with the updated application and gradually routing traffic to it. Canary deployments are similar to blue-green deployments but involve routing a small percentage of traffic to the new environment to test it before gradually increasing the traffic.

**Worker Environment**

A worker environment in Elastic Beanstalk is a fully managed environment that is designed to run background tasks or "worker" processes. Worker environments are typically used for tasks that don't require immediate user feedback or that run asynchronously from the user interface. Examples of worker tasks include image processing, database updates, and sending emails.

When you create a worker environment in Elastic Beanstalk, the service provisions and configures the resources necessary to run your application, including EC2 instances, message queues, and security groups. Elastic Beanstalk also sets up monitoring and logging for your application.

Worker environments support several programming languages and frameworks, including Java, .NET, Node.js, PHP, Python, Ruby, and Go. Elastic Beanstalk also supports several queue services, including Amazon Simple Queue Service (SQS) and RabbitMQ, which makes it easier to manage the message queues that your application depends on.

Elastic Beanstalk supports several deployment options for worker environments, including rolling updates and blue-green deployments. With rolling updates, Elastic Beanstalk deploys new versions of your application one instance at a time, while still processing tasks from the remaining instances. Blue-green deployments involve creating a new environment with the updated application and gradually routing tasks to it.

In summary, Elastic Beanstalk provides a fully managed environment for running web applications and background tasks. With Elastic Beanstalk, developers can focus on writing code rather than worrying about infrastructure, and they can take advantage of several deployment options to deploy and manage their applications with ease.