Step 15-: You can delete the instance permanently by selecting instance state followed by stop

7. Conclusion:

Cloud Computing Lab

Experiment No.: 4

Platform as a Service

Experiment No. 4

1. **Aim:** To study and Implement Platform as a Service using AWS Elastic Beanstalk.
2. **Objectives:** To demonstrate the steps to deploy Web applications or Web services written in different languages on AWS Elastic Beanstalk.
3. **Outcomes:** The learner will be able to analyze various cloud computing service models and implement them to solve the given problems.
4. **Hardware / Software Required:** Internet Connection, Internet browser.
5. **Theory:**

   Amazon Web Services (AWS) comprises over one hundred 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 having to learn about the infrastructure that runs those applications. 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.

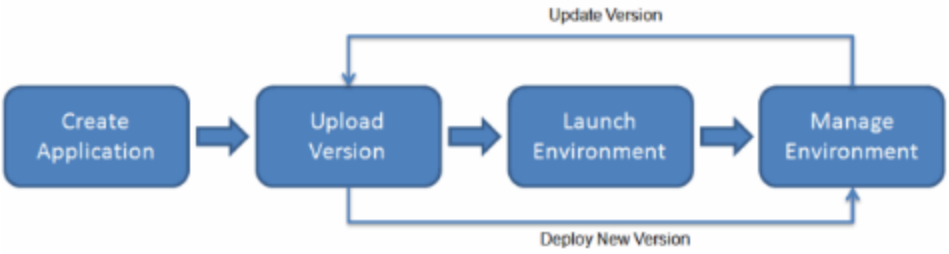
   Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.

   You can interact with Elastic Beanstalk by using the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or **eb**, a high-level CLI designed specifically for Elastic Beanstalk.

   You can also perform most deployment tasks, such as changing the size of your fleet of Amazon EC2 instances or monitoring your application, directly from the Elastic Beanstalk web interface (console).

   To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk,

and then provide some information about the application. Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions. The following diagram illustrates the workflow of Elastic Beanstalk.



After you create and deploy your application, information about the application—including metrics, events, and environment status—is available through the Elastic Beanstalk console, APIs, or Command Line Interfaces, including the unified AWS CLI.

6. **Procedure**

Setting up: Create an AWS account

If you're not already an AWS customer, you need to create an AWS account. Signing up enables you to access Elastic Beanstalk and other AWS services that you need.

To sign up for an AWS account

- Open the Elastic Beanstalk console, and in the Regions list, select your AWS Region.
- Follow the instructions shown.

Step 1: Create an example application

In this step, you create a new application. Elastic Beanstalk supports platforms for different programming languages, application servers, and Docker containers. You choose a platform when you create the application.

Create an application and an environment

To create your example application, you'll use the Create a web app console wizard. It creates an Elastic Beanstalk application and launches an environment within it. An environment is the collection of AWS resources required to run your application code.

To create an example application

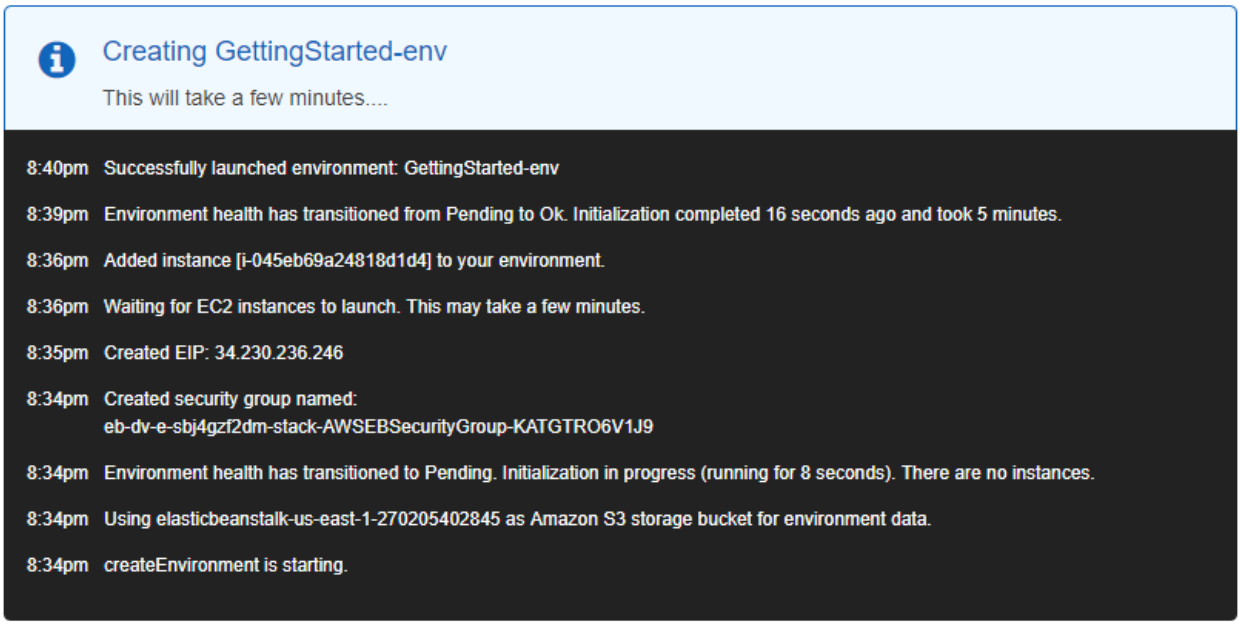
1. Open the Elastic Beanstalk console using this link: https://console.aws.amazon.com/elasticbeanstalk/home#/gettingStarted?applicationName=getting-started-app
2. Optionally add application tags.

3. For Platform, choose a platform, and then choose Create application.

To run the example application on AWS resources, Elastic Beanstalk takes the following actions. They take about five minutes to complete.

1. Creates an Elastic Beanstalk application named getting-started-app.
2. Launches an environment named GettingStartedApp-env with these AWS resources:
   - An Amazon Elastic Compute Cloud (Amazon EC2) instance (virtual machine)
   - An Amazon EC2 security group
   - An Amazon Simple Storage Service (Amazon S3) bucket
   - Amazon CloudWatch alarms
   - An AWS CloudFormation stack
   - A domain name
3. Creates a new application version named Sample Application. This is the default Elastic Beanstalk example application file.
4. Deploys the code for the example application to the **GettingStartedApp-env** environment.

During the environment creation process, the console tracks progress and displays events.



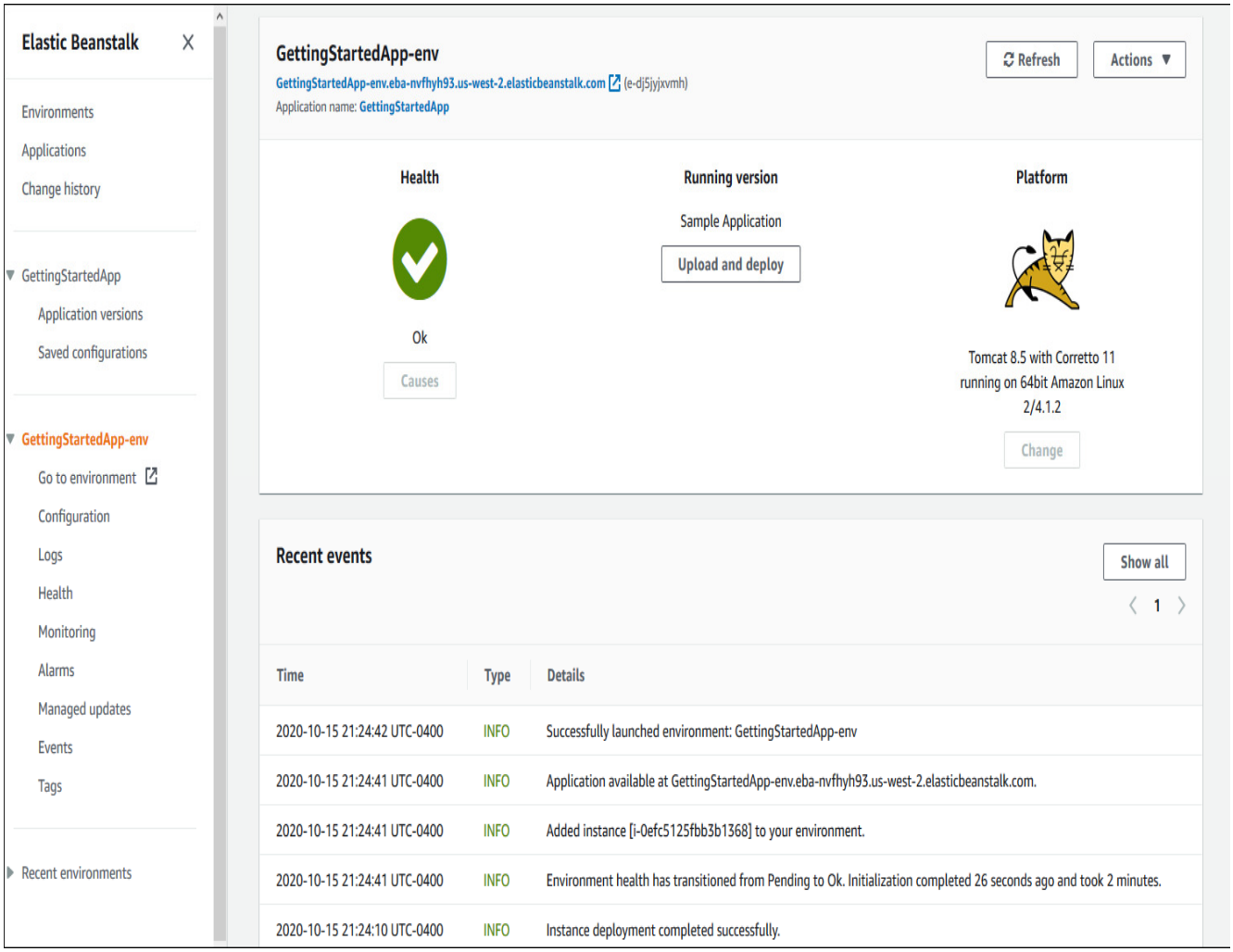
When all of the resources are launched and the EC2 instances running the application pass health checks, the environment's health changes to Ok. You can now use your web application's website.

Step 2: Explore your environment

To view the environment overview

1. Open the Elastic Beanstalk console, and in the Regions list, select your AWS Region.
2. In the navigation pane, choose Environments, and then choose the name of your environment from the list.

The environment overview pane shows top level information about your environment. This includes its name, its URL, its current health status, the name of the currently deployed application version, and the platform version that the application is running on. Below the overview pane you can see the five most recent environment events.



- While Elastic Beanstalk creates your AWS resources and launches your application, the environment is in a Pending state. Status messages about launch events are continuously added to the overview.
- The environment's URL is located at the top of the overview, below the environment name. This is the URL of the web application that the environment is running. Choose this URL to get to the example application's Congratulations page.
- The navigation page on the left side of the console links to other pages that contain more detailed information about your environment and provide access to additional features:
  - **Configuration** – Shows the resources provisioned for this environment, such as the Amazon Elastic Compute Cloud (Amazon EC2) instances that host your application. You can configure some of the provisioned resources on this page.
  - **Health** – Shows the status of and detailed health information about the Amazon EC2 instances running your application.
  - **Monitoring** – Shows statistics for the environment, such as average latency and CPU utilization. You can use this page to create alarms for the metrics that you are monitoring.
  - **Events** – Shows information or error messages from the Elastic Beanstalk service and from other services whose resources this environment uses.
  - **Tags** – Shows environment tags and allows you to manage them. Tags are key-value pairs that are applied to your environment.

Step 3: Deploy a new version of your application

Periodically, you might need to deploy a new version of your application. You can deploy a new version at any time, as long as no other update operations are in progress on your environment.

To update your application version

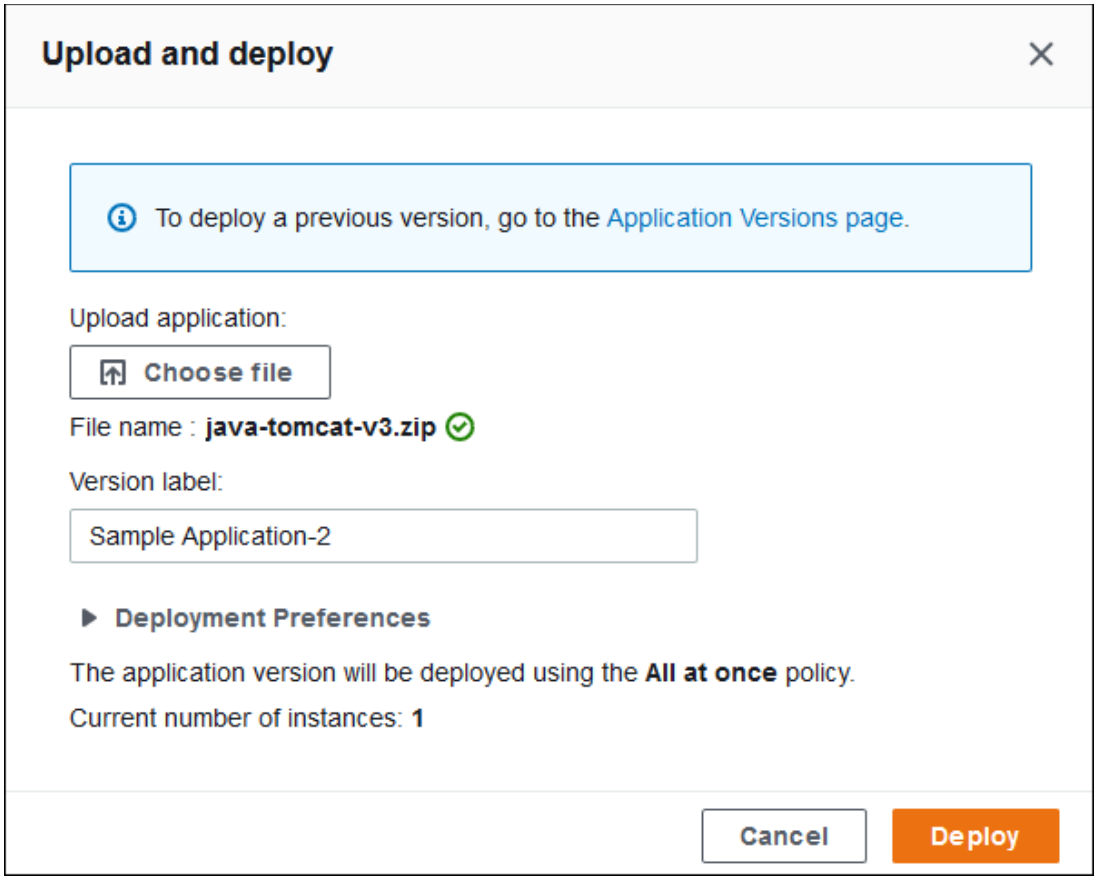
1. Create one simple web page using PHP and create zip file.

2. Open the Elastic Beanstalk console, and in the Regions list, select your AWS Region.

3. In the navigation pane, choose **Environments**, and then choose the name of your environment from the list.

4. On the environment overview page, choose **Upload and deploy**.

5. Choose **Choose** file, and then upload the zip file created in step 1. The console automatically fills in the Version label with a new unique label. If you type in your own version label, ensure that it's unique.



6. **Choose Deploy.**

While Elastic Beanstalk deploys your file to your Amazon EC2 instances, you can view the deployment status on the environment's overview. While the application version is updated, the Environment Health status is gray. When the deployment is complete, Elastic Beanstalk performs an application health check. When the application responds to the health check, it's considered healthy and the status returns to green. The environment overview shows the new Running Version—the name you provided as the Version label.

Elastic Beanstalk also uploads your new application version and adds it to the table of application versions. To view the table, choose Application versions under getting-started-app on the navigation pane.

7. **Conclusion**