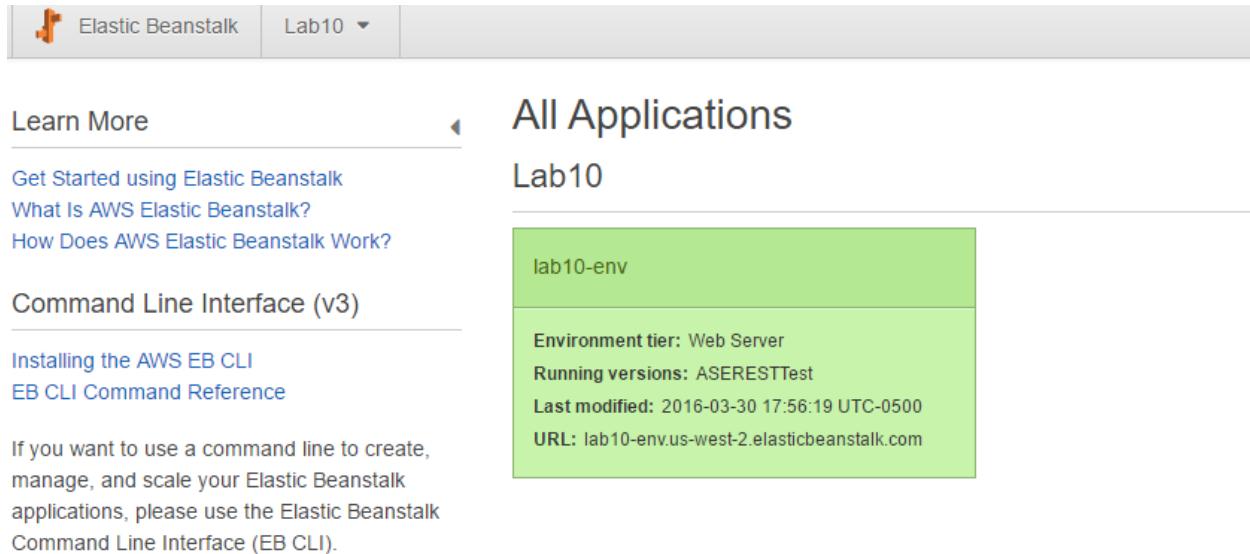


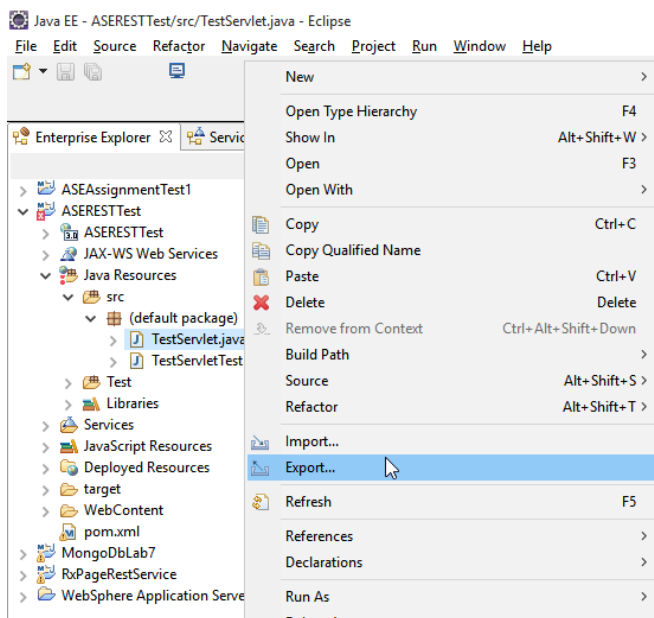
Deploying Web Application to ElasticBean Stock on AWS

1. Screen shot of application on Elastic Beanstalk



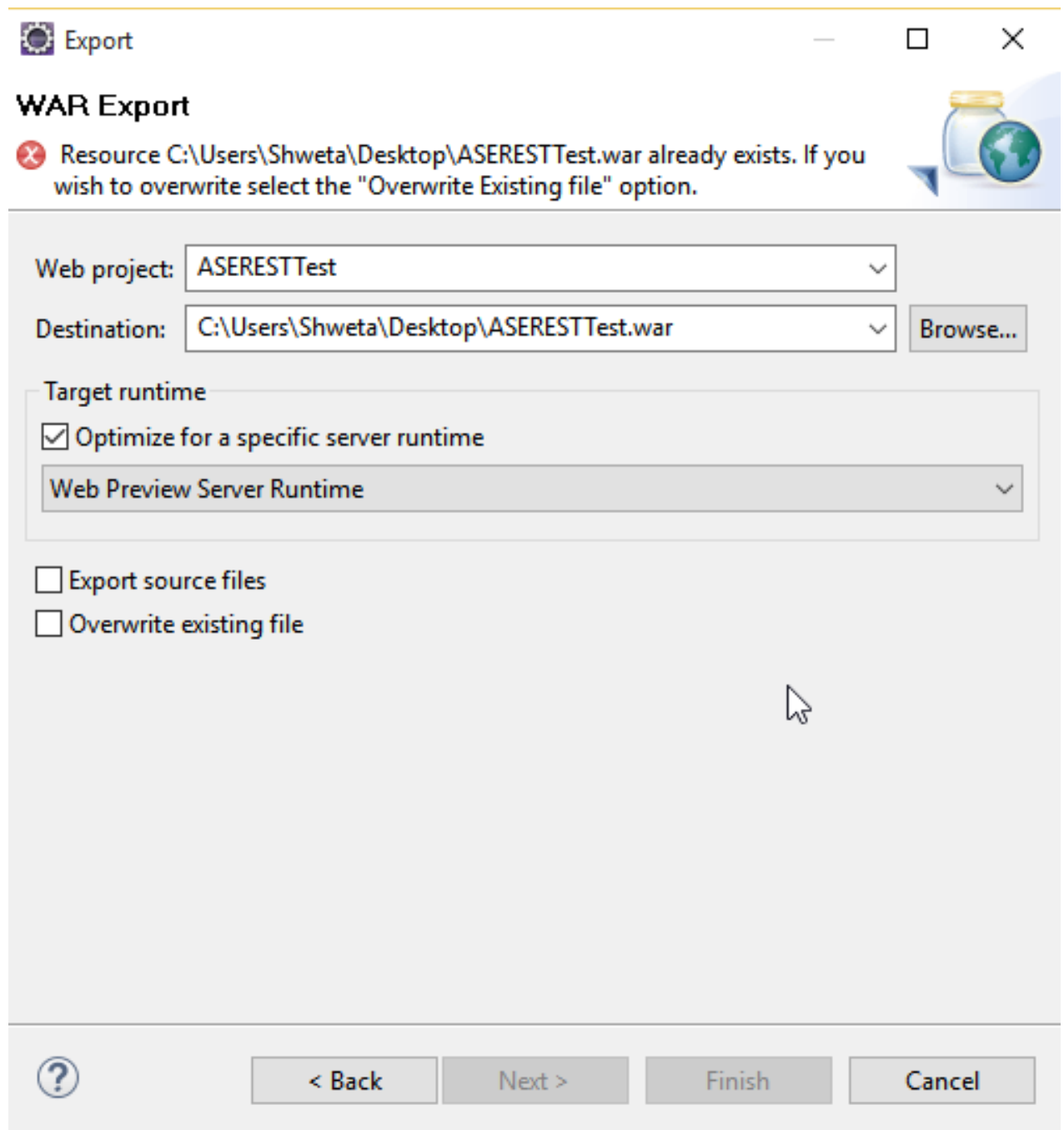
The screenshot shows the AWS Elastic Beanstalk console. At the top, there's a header with the Elastic Beanstalk logo and a dropdown menu set to 'Lab10'. Below the header, there's a 'Learn More' section with links: 'Get Started using Elastic Beanstalk', 'What Is AWS Elastic Beanstalk?', and 'How Does AWS Elastic Beanstalk Work?'. To the right of 'Learn More' is a section titled 'All Applications' with a sub-section 'Lab10'. This section contains a green box with the following information: 'lab10-env', 'Environment tier: Web Server', 'Running versions: ASERESTTest', 'Last modified: 2016-03-30 17:56:19 UTC-0500', and 'URL: lab10-env.us-west-2.elasticbeanstalk.com'. Below the 'Learn More' section, there's a 'Command Line Interface (v3)' section with links: 'Installing the AWS EB CLI' and 'EB CLI Command Reference'. At the bottom of this section, there's a paragraph: 'If you want to use a command line to create, manage, and scale your Elastic Beanstalk applications, please use the Elastic Beanstalk Command Line Interface (EB CLI).'

2. Exporting the REST service Java file to War file.



ASE Lab # 10 submitted by Shweta Parihar Class ID – 42

3. Export continuation



Export

WAR Export

Resource C:\Users\Shweta\Desktop\ASERESTTest.war already exists. If you wish to overwrite select the "Overwrite Existing file" option.

Web project: ASERESTTest

Destination: C:\Users\Shweta\Desktop\ASERESTTest.war [Browse...](#)

Target runtime

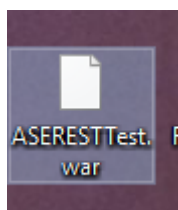
☒ Optimize for a specific server runtime

Web Preview Server Runtime

☐ Export source files

☐ Overwrite existing file

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)



ASE Lab # 10 submitted by Shweta Parihar Class ID – 42

4. Uploading and deploying the war file on the ElasticBean Stock server

Upload and Deploy

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

Upload application:

Choose File

ASERESTTest.war

Version label:

ASERESTTest

Deployment Preferences

Elastic Beanstalk will deploy to **30%** of instances in your auto scaling group at a time. Current number of instances: **1**

Batch size:

☒ Percentage

30

% of instances at a time

☐ Fixed

1

 instances at a time (max: 4)

Ignore health check:

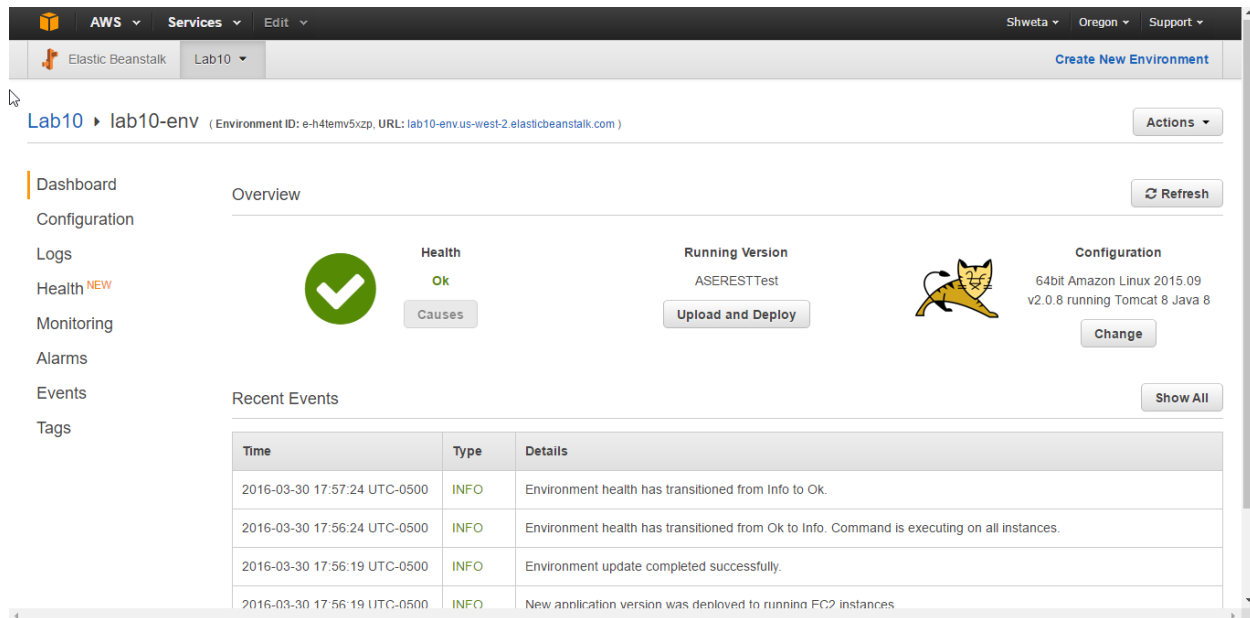
false

Cancel

Deploy

Continue to next page

ASE Lab # 10 submitted by Shweta Parihar Class ID – 42



Lab10 ▸ lab10-env (Environment ID: e-h4temy5xzp, URL: lab10-env.us-west-2.elasticbeanstalk.com)

Dashboard
Configuration
Logs
Health **NEW**
Monitoring
Alarms
Events
Tags

Overview

Health **Ok**
Causes

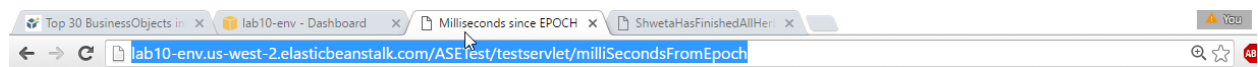
Running Version
ASERESTest
Upload and Deploy

Configuration
64bit Amazon Linux 2015.09
v2.0.8 running Tomcat 8 Java 8
Change

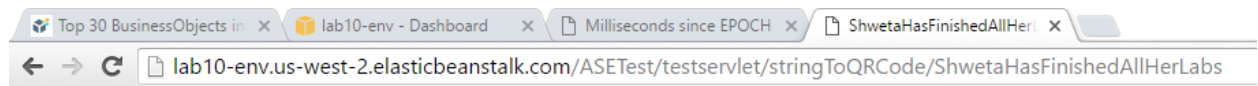
Recent Events

Time	Type	Details
2016-03-30 17:57:24 UTC-0500	INFO	Environment health has transitioned from Info to Ok.
2016-03-30 17:56:24 UTC-0500	INFO	Environment health has transitioned from Ok to Info. Command is executing on all instances.
2016-03-30 17:56:19 UTC-0500	INFO	Environment update completed successfully.
2016-03-30 17:56:19 UTC-0500	INFO	New application version was deployed to running EC2 instances.

5. REST Service working using the URL generated from AWS



Milliseconds since EPOCH : 1459378644581



6. Java Code for the REST service

```
import javax.imageio.ImageIO;  
import javax.ws.rs.GET;  
import javax.ws.rs.Path;  
import javax.ws.rs.PathParam;  
import javax.ws.rs.Produces;  
import javax.ws.rs.core.Response;
```

ASE Lab # 10 submitted by Shweta Parihar Class ID – 42

```
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

import net.glxn.qrgen.QRCode;
import net.glxn.qrgen.image.ImageType;

@Path("/testservlet")
public class TestServlet {

    @Path("/millisecondsFromEpoch")
    @GET
    @Produces("text/html")
    public String getEPOCHTime() {

        String html = "<html><head><title>Milliseconds since
EPOCH</title></head><body><h1>Milliseconds since EPOCH :
"+getMillisecondsFromEpoch()+"</h1></body></html>";
        return html;
    }

    long getMillisecondsFromEpoch()
    {
        return System.currentTimeMillis();
    }

    @Path("/stringToQRCode/{stringToConvert}")
    @GET
    @Produces("image/png")
    public Response convertStringToQRCode(@PathParam("stringToConvert") String
inputStringToConvert) {

        return
Response.ok(convertTextToQRStream(inputStringToConvert).toByteArray()).build();
    }
}
```

ASE Lab # 10 submitted by Shweta Parihar Class ID – 42

```
        ByteArrayOutputStream convertTextToQRStream(String input)
        {
            ByteArrayOutputStream baos =
QRCode.from(input).to(ImageType.PNG).stream();
//        byte[] imageData = baos;
            return baos;
        }
    }
```