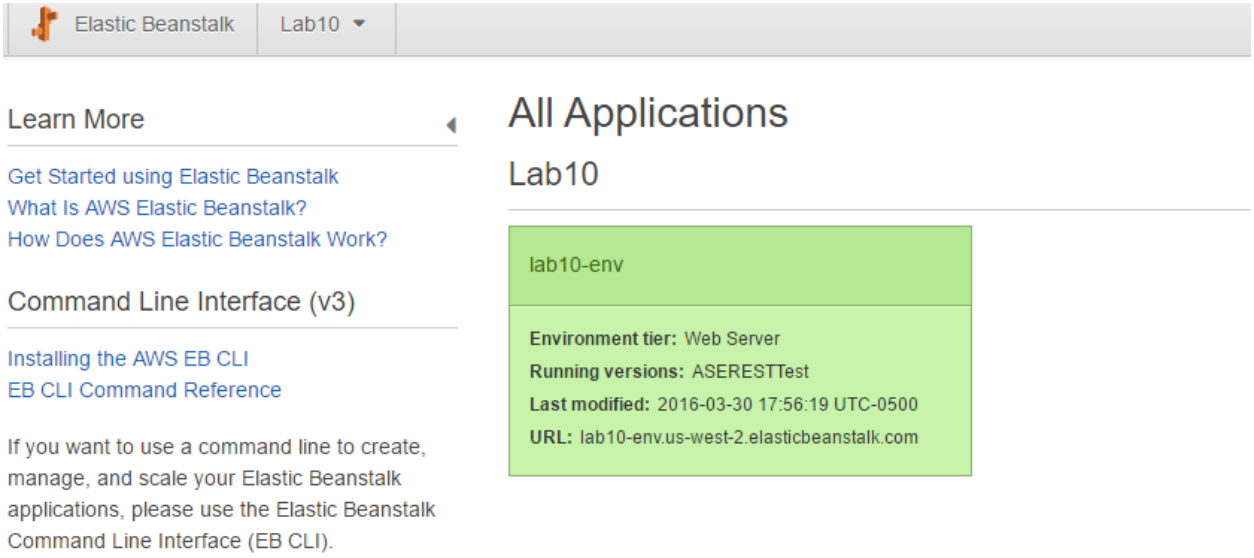
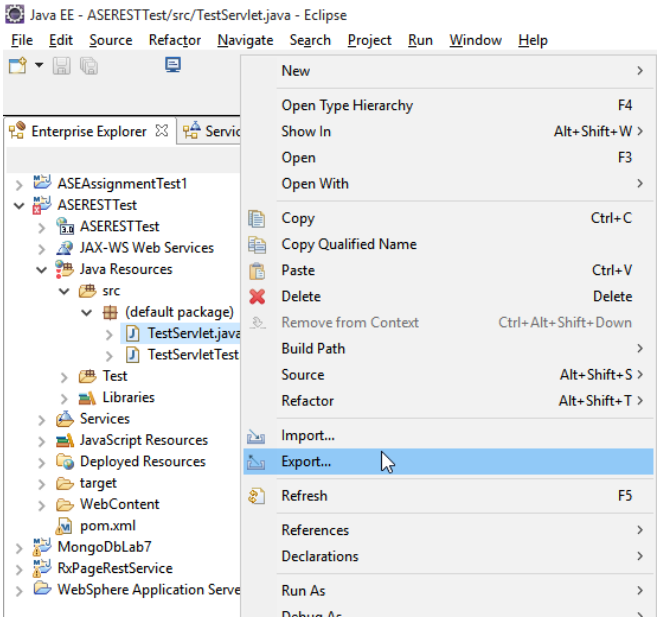
ASE Lab # 10 submitted by Shweta Parihar Class ID – 42

Deploying Web Application to ElasticBean Stock on AWS

1. Screen shot of application on Elastic Beanstalk

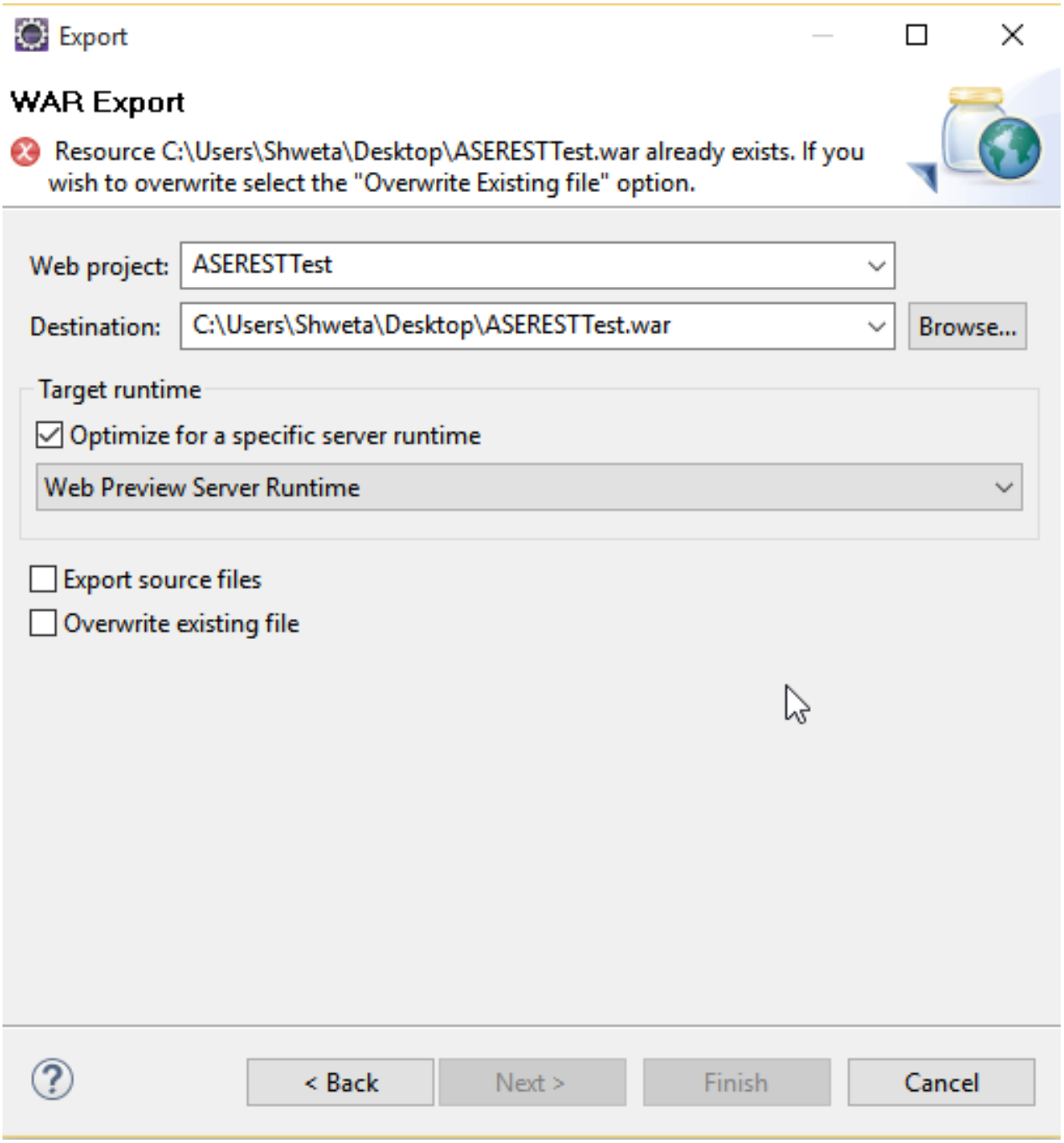


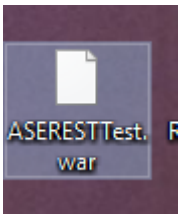
2. Exporting the REST service Java file to War file.



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

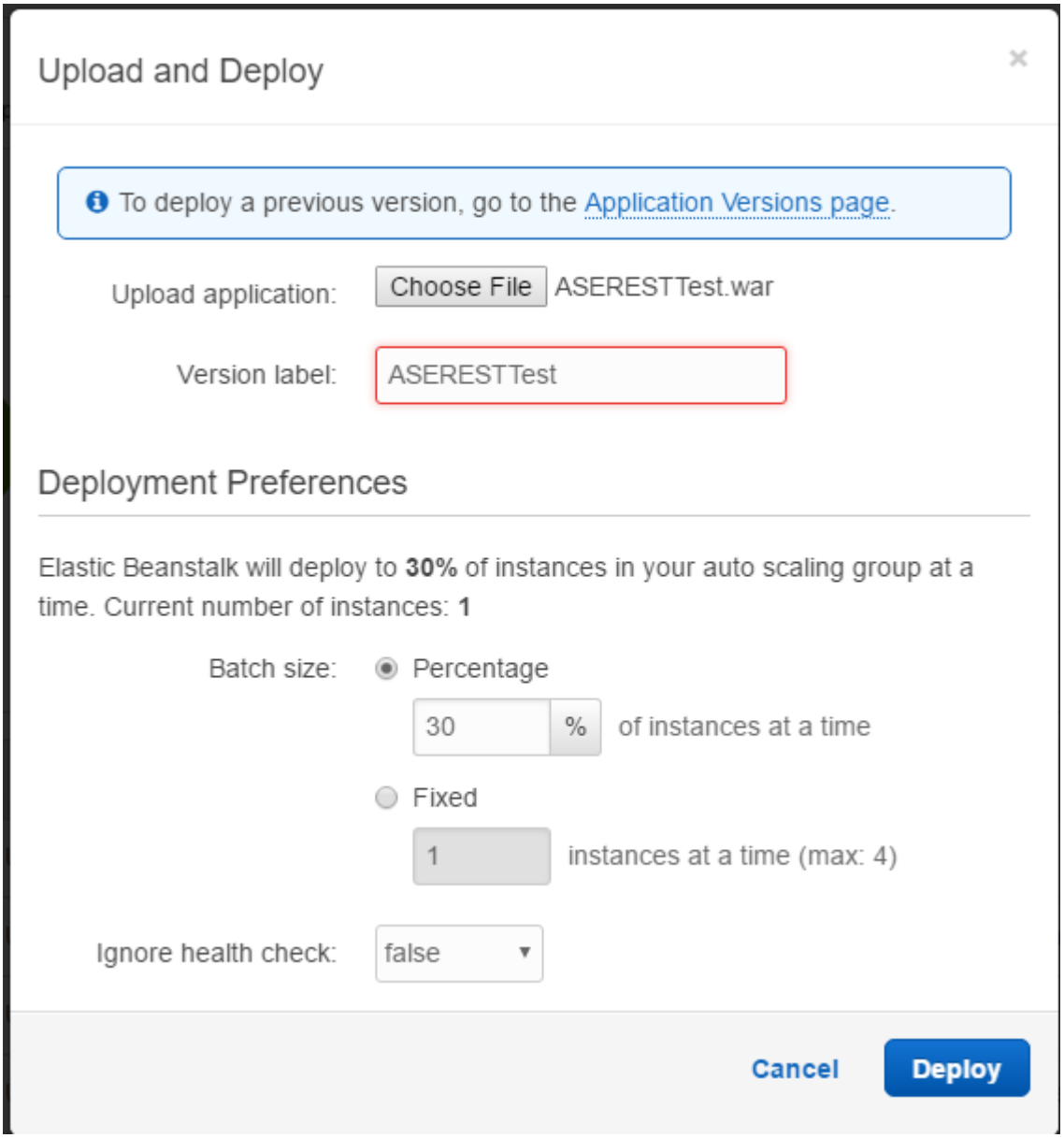
3. Export continuation





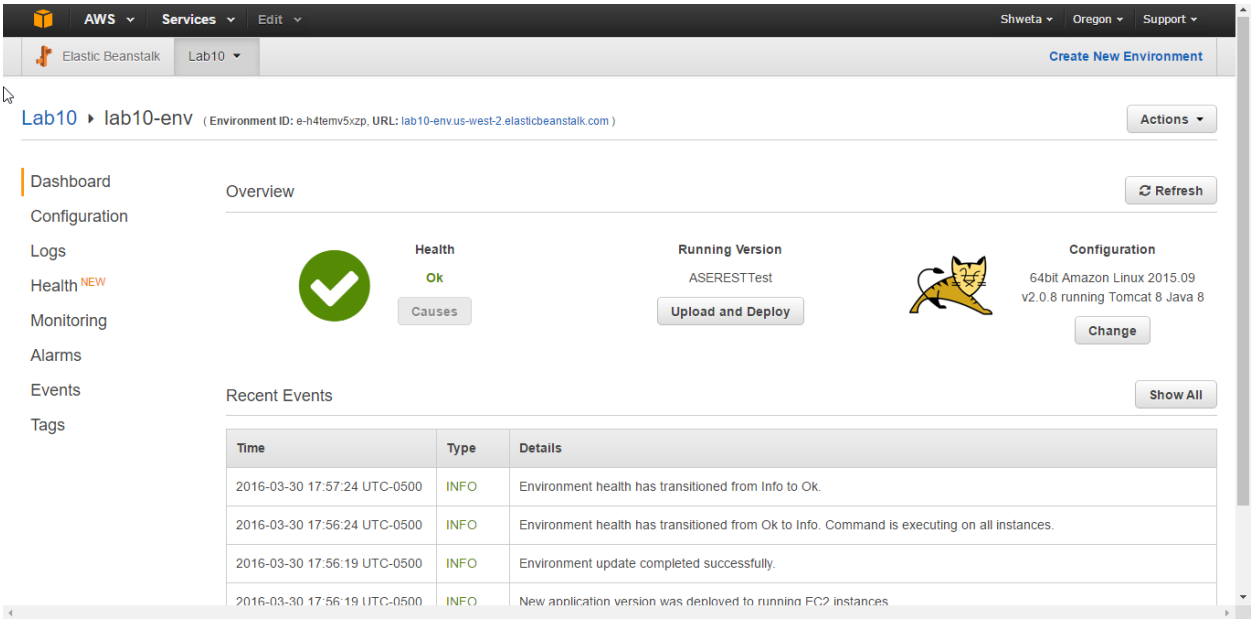
ASE Lab # 10 submitted by Shweta Parihar Class ID – 42

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



Continue to next page

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



5. REST Service working using the URL generated from AWS

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;
        }
}
```