

## Homework #7 Amazon Web Services (AWS) with Node.js

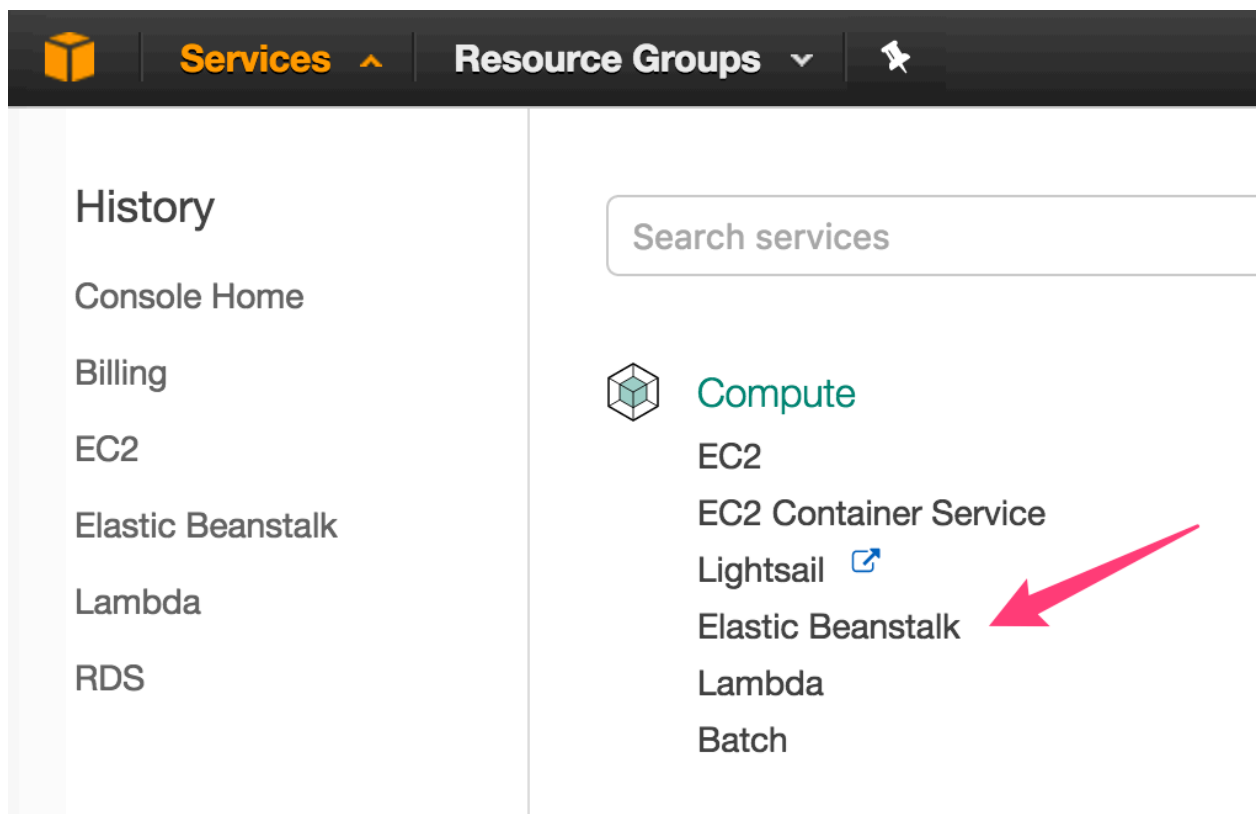
Using the instructions below one can establish a service at AWS. Once established, you will be able to move your Node.js program developed for Assignment #8 to your AWS instance and have it executed there.

### 1-4. Sign ups

This section assumes that you have performed the installation for homework #5, including AWS Educate Sign Up.

### 5. Set up the Default Elastic Beanstalk Application

- Click the top left menu **Services**
- From the list of Amazon Web Services, select **Elastic Beanstalk**, under **Compute**.



- Select **Create New Application** in the top right, right underneath your account name, and follow the Wizard.
- In the **Application name** field, enter a name for your application. Click **Create**.

**Create New Application**

**Application Name**   
Maximum length of 100 characters, not including forward slash (/).

**Description**   
Maximum length of 200 characters.

**Tags**  
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive.  
[Learn more](#)

Key (127 characters maximum)	Value (255 characters maximum)
<input type="text"/>	<input type="text"/>

50 remaining

[Cancel](#) [Create](#)

- In the **Environment** section click on the **Create One Now** hyperlink

[All Applications](#) > **NewPHP**

[Actions](#)

**Environments**

Application versions

Saved configurations

No environments currently exist for this application. [Create one now.](#)

- In the **Choose an environment tier** dialog select **Web server environment** and click on **Select** button.

**Select environment tier**

AWS Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web servers are standard applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue. Worker applications post those messages to your application by using HTTP.

**Web server environment**

Run a website, web application, or web API that serves HTTP requests.  
[Learn more](#)

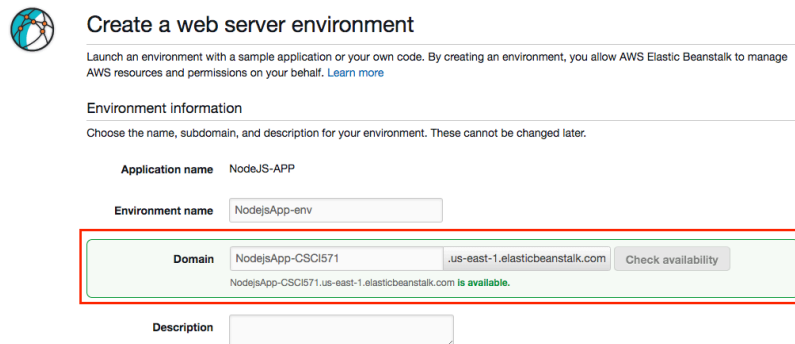
**Worker environment**

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.  
[Learn more](#)

[Cancel](#) [Select](#)

- In the **Environment Information** section, select a **Domain** (use the default by leaving the field blank, or check availability of your own subdomain of

elasticbeanstalk.com). Click on **“Check availability”** button. Your URL should be green. Otherwise you should change the environment URL.



**Create a web server environment**

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)

Environment information

Choose the name, subdomain, and description for your environment. These cannot be changed later.

Application name NodeJS-APP

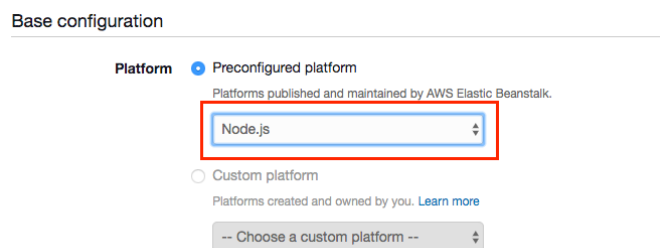
Environment name NodejsApp-env

Domain NodejsApp-CSCI571 .us-east-1.elasticbeanstalk.com [Check availability](#)

NodejsApp-CSCI571.us-east-1.elasticbeanstalk.com **is available.**

Description

- In the **Base configuration** section, choose the **Preconfigured platform**, and select the platform from the drop-down list:
  - Platform: **Node.js**



Base configuration

Platform ☒ Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

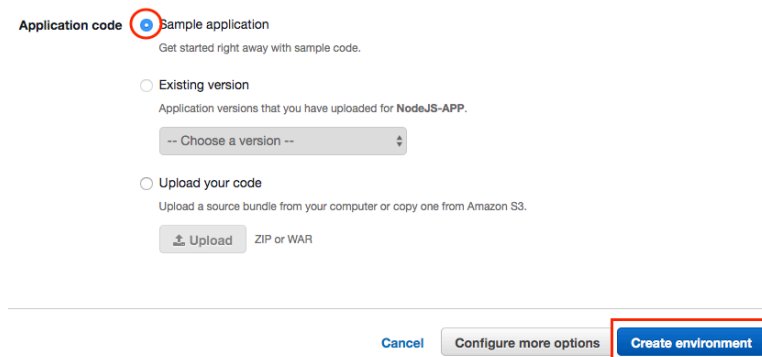
Node.js

☐ Custom platform

Platforms created and owned by you. [Learn more](#)

-- Choose a custom platform --

- In the **Application Code** section, select **Sample application**.



Application code ☒ Sample application

Get started right away with sample code.

☐ Existing version

Application versions that you have uploaded for NodeJS-APP.

-- Choose a version --

☐ Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

[Upload](#) ZIP or WAR

[Cancel](#) [Configure more options](#) [Create environment](#)

- Click **Create environment**.
- After a minute or so the **“Creating <environment-name>”** dialog appears, with the message “This will take a few minutes...”

### Creating NodejsApp-env

This will take a few minutes..

```

9:45am Created EIP: 3.232.16.111
9:45am Created security group named:
       awseb-e-kjudbpqmsc-stack-AWSEBSecurityGroup-2OKUFL6J075
9:44am Using elasticbeanstalk-us-east-1-034721222574 as Amazon S3 storage bucket for environment data.
9:44am createEnvironment is starting.
                    
```

#### Learn More

- [Get started using Elastic Beanstalk](#)
- [Modify the code](#)
- [Create and connect to a database](#)
- [Add a custom domain](#)

#### Featured

- [Create your own custom platform](#)

#### Command Line Interface (v3)

- [Installing the AWS EB CLI](#)
- [EB CLI Command Reference](#)

You will need to wait for several minutes as your Linux + Nginx + Node.js instance is created and launched. You will see several messages appear as the instance is being created and deployed. Once creation and launch are completed, you will see **Overview** dashboard appear.

Overview

Refresh

**Health**  
**Ok**  
[Causes](#)

**Running Version**  
Sample Application  
[Upload and Deploy](#)

**Platform**  
Node.js running on 64bit Amazon Linux/4.11.0  
[Change](#)

Recent Events

Show All

Time	Type	Details
2019-10-22 09:47:54 UTC-0700	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 11 seconds ago and took 3 minutes.
2019-10-22 09:47:42 UTC-0700	INFO	Successfully launched environment: NodejsApp-env
2019-10-22 09:47:42 UTC-0700	INFO	Application available at NodejsApp-CSCI571.us-east-1.elasticbeanstalk.com.
2019-10-22 09:47:06 UTC-0700	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
2019-10-22 09:46:54 UTC-0700	INFO	Added instance [i-0ec03d138015b4afc] to your environment.

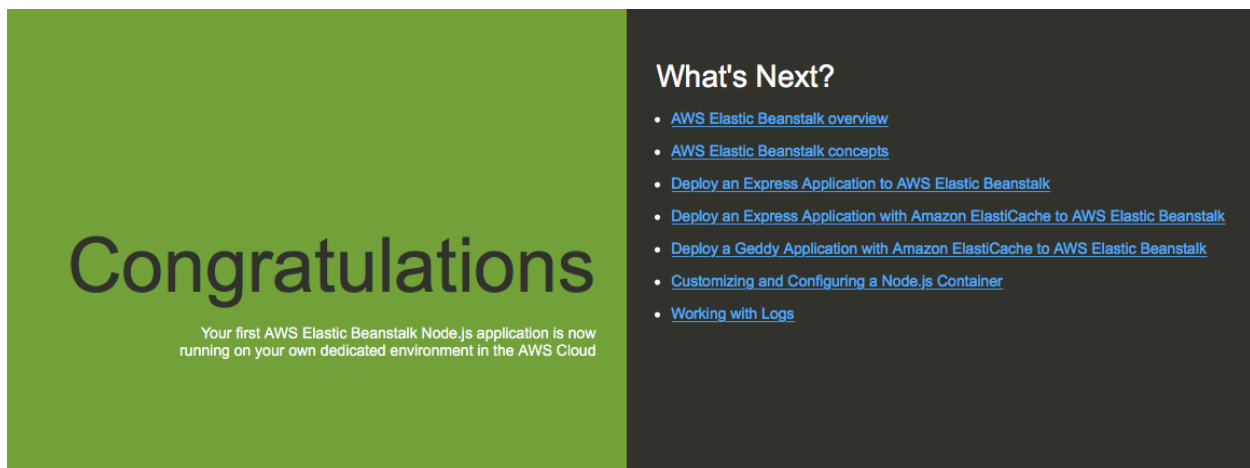
## Node.js Instance Dashboard

You may see INFO, WARNINGS and ERRORS in the Events Type field. Anything except for ERRORS is fine.

Beside “<YourEnvironment>” subtitle there is a **URL** such as *YourAppName-env.elasticbeanstalk.com*.



Click on it. You should see the "Congratulations" page. If you see it as shown below, your application and environment have been created properly. Then go back to Elastic Beanstalk console.



## Node.js Sample Application

### 6. Upload your Node.js application

Develop your Node.js server application, and make sure that you name the command used to start the Node.js application **server.js** or **app.js**. Compress the file with ZIP so that the resulting “source bundle” is named something like **nodejs-v1.zip**. On a Mac, you can right click the file or folder and choose compress. On Windows, there are several free programs, such as 7-Zip or FreeZip, etc. that you can use.

While PHP on AWS comes with all needed libraries included, Node.js comes bare, with no libraries (also known as “packages”) installed. You can include a **package.json** file in your source bundle to install packages during deployment. You use a `package.json` file in the root of your project source to use **npm** to install packages that your application requires.

This is an example `package.json` file:

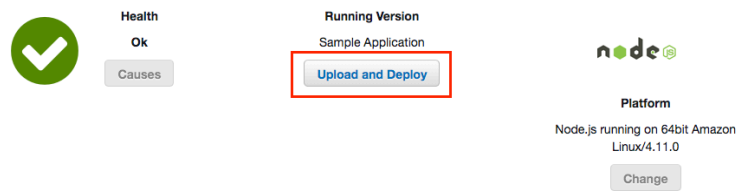
```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

When a `package.json` file is present in your bundle, Elastic Beanstalk runs **npm install** to install dependencies.

The Node.js platform on AWS includes a proxy server to serve static assets, forward traffic to your application, and compress responses. The default proxy server is **Nginx**.

## 6.1 Upload and Deploy

From the Elastic Beanstalk console, select your environment and click on the **Upload and Deploy** button.



The **Upload and Deploy** popup will display. Enter a Version **label** (e.g., version 1.0). Click on the **Choose File** button and select the **nodejs-v1.zip** file. Then click on **Deploy** button.

Upload and Deploy

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

Upload application: Choose File nodejs-v1.zip nodejs-v1.zip

Version label: nodejs-v1

Cancel Deploy

Again, wait several minutes for the *rotating wheel* to finish and the green circle with checkmark to appear. Click again on the link “**YourAppName-env.elasticbeanstalk.com**”. Check that your Node.js app is running correctly.

- **Important Note:** in the future if you want to upload an updated version of the source bundle *nodejs-v1.zip*, you should enter a different version label. Otherwise, you will get an error. It is recommended that you use increasing version numbers (2.0, 2.1, 3.0, or labels such as nodejs-v1, nodejs-v2, etc.)

For additional information, please check the AWS Developer Guide article titled “*Using the AWS Elastic Beanstalk Node.js Platform*” at:

[http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create\\_deploy\\_nodejs.container.html](http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_nodejs.container.html)

and here (new docs look):

[https://docs.aws.amazon.com/en\\_pv/elasticbeanstalk/latest/dg/create\\_deploy\\_nodejs.container.html](https://docs.aws.amazon.com/en_pv/elasticbeanstalk/latest/dg/create_deploy_nodejs.container.html)