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

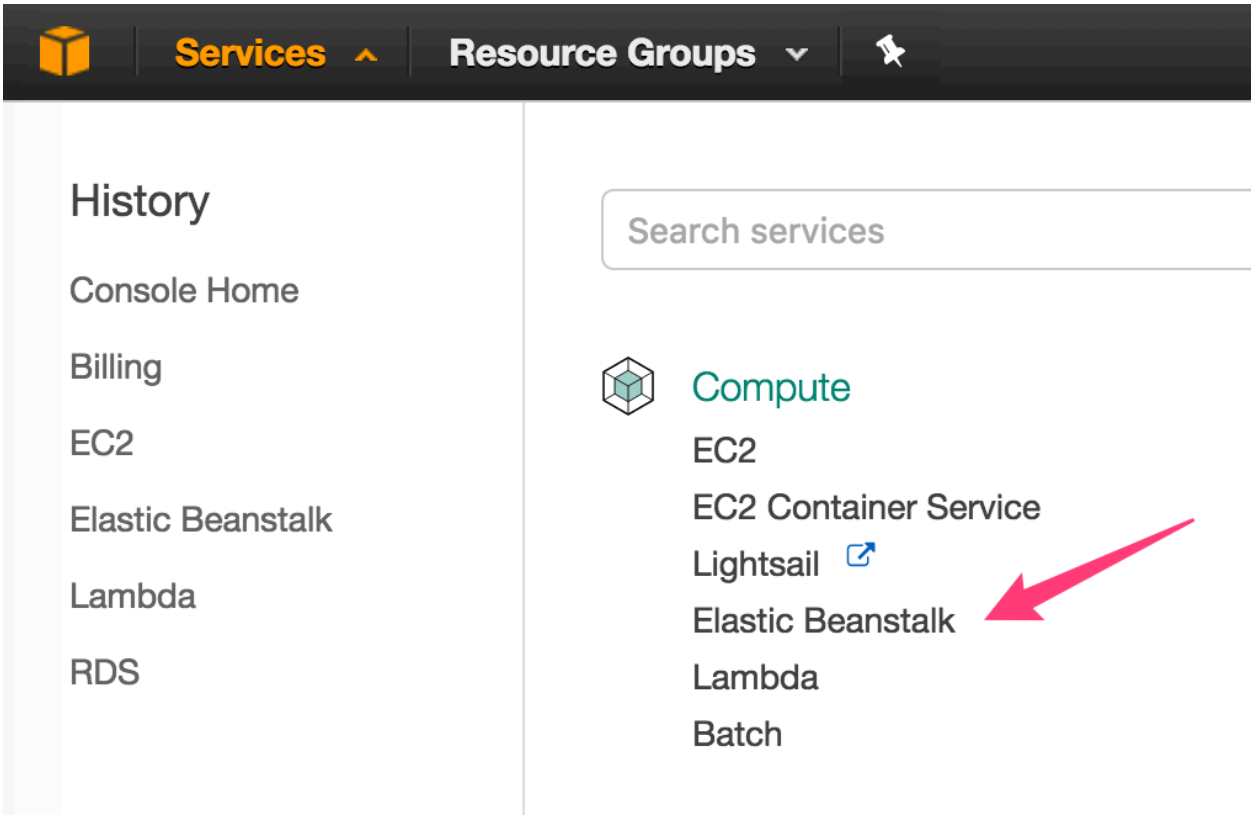
Using the instructions below one can establish a service at AWS. Once established, you will be able to move your Node.js program developed for Assignment #8 to your AWS instance and have it executed there.

1-4. Sign ups

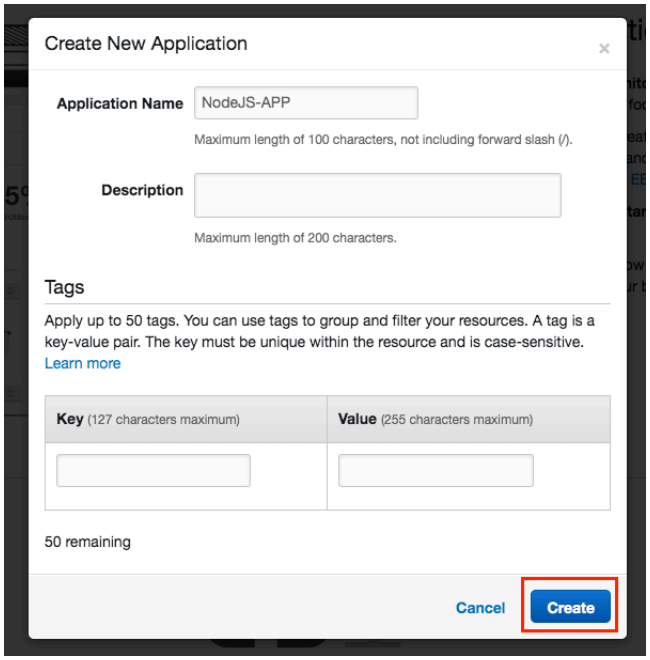
This section assumes that you have performed the installation for homework #5, including AWS Educate Sign Up.

5. Set up the Default Elastic Beanstalk Application

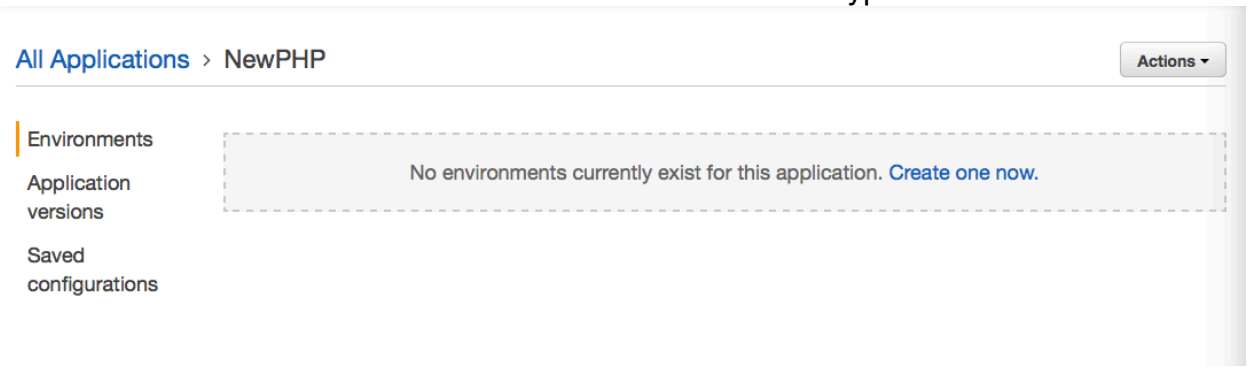
- Click the top left menu **Services**
- From the list of Amazon Web Services, select **Elastic Beanstalk,** under **Compute**.



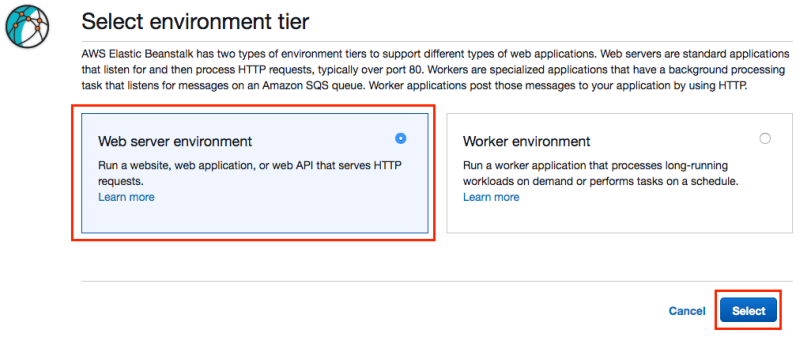
- Select **Create New Application** in the top right, right underneath your account name, and follow the Wizard.
- In the **Application name** field, enter a name for your application. Click **Create**.



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

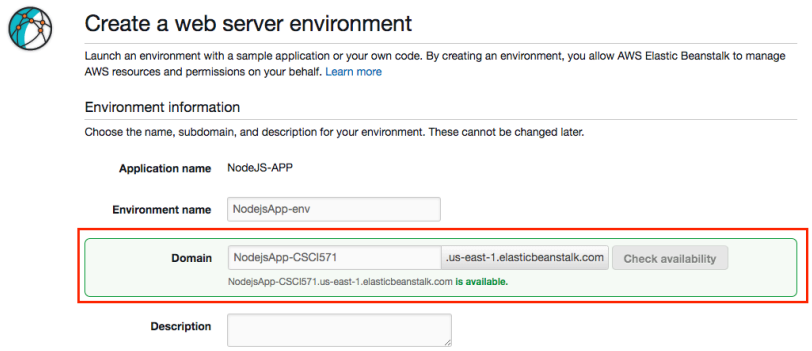


- In the **Choose an environment tier** dialog select **Web server environment** and click on **Select** button.

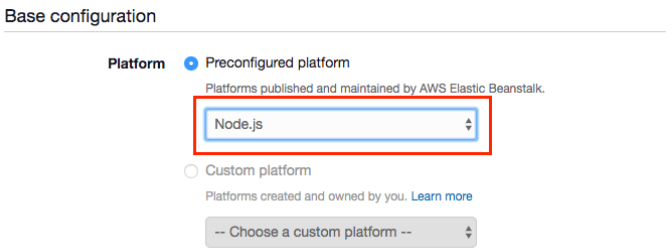


- In the **Environment Information** section, select a **Domain** (use the default by leaving the field blank, or check availability of your own subdomain of

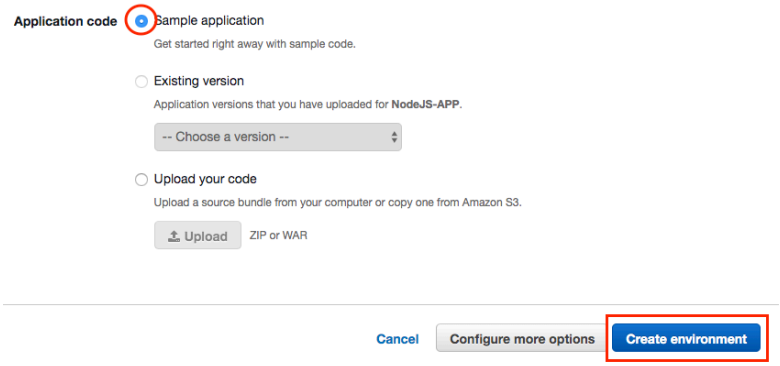
elasticbeanstalk.com). Click on "**Check availability**" button. Your URL should be green. Otherwise you should change the environment URL.



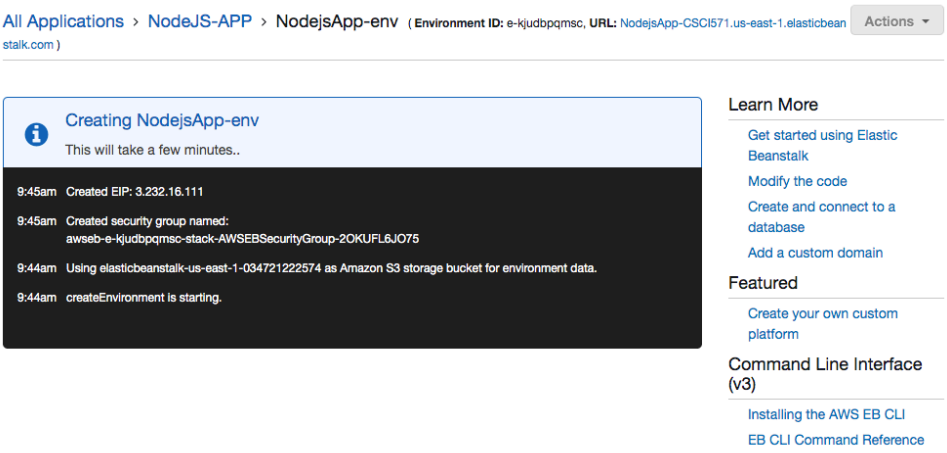
- In the **Base configuration** section, choose the **Preconfigured platform**, and select the platform from the drop-down list:
    - Platform: **Node.js**



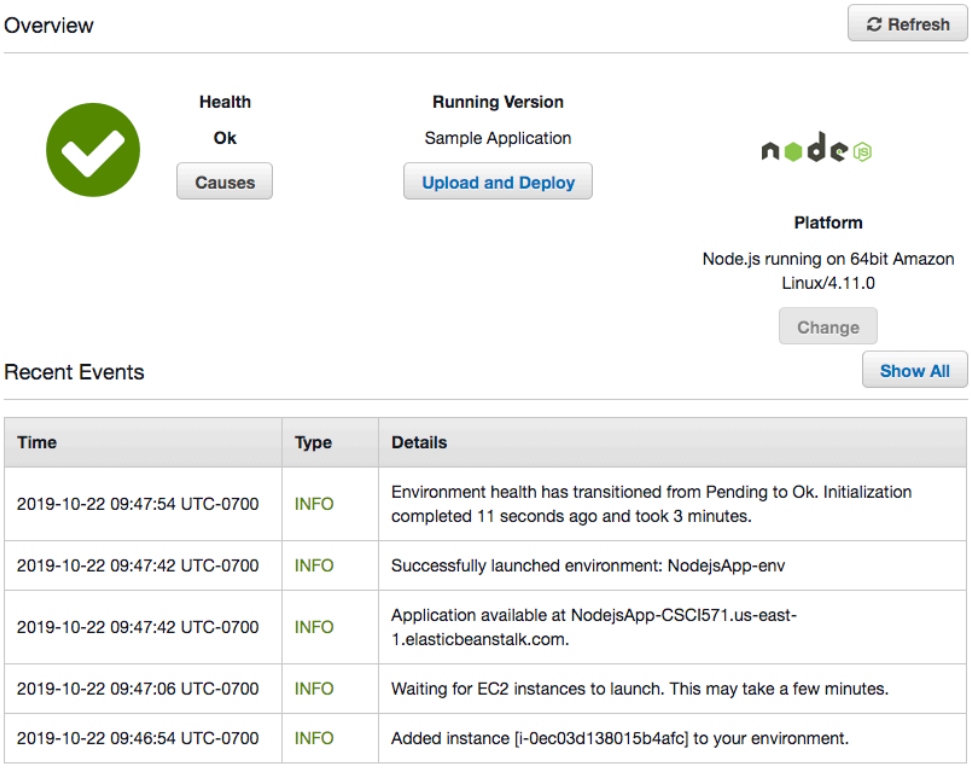
- In the **Application Code** section, select **Sample application.**



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



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.

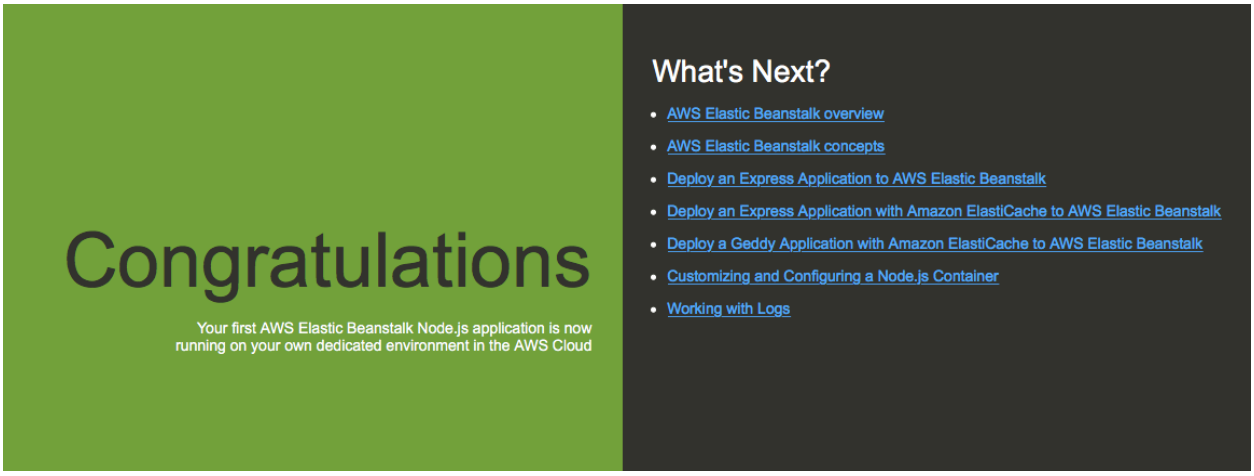


Node.js Instance Dashboard

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:

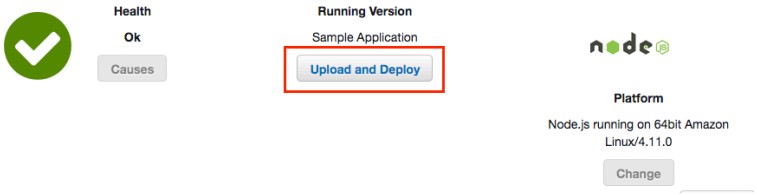
```
{
  "name": "my-app",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

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

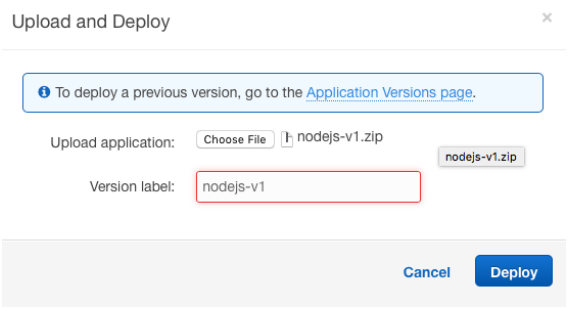
The Node.js platform on AWS includes a proxy server to serve static assets, forward traffic to your application, and compress responses. The default proxy server is **Nginx**.

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.



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

and here (new docs look):

https://docs.aws.amazon.com/en_pv/elasticbeanstalk/latest/dg/create_deploy_nodejs.container.html