**June 2019**



**Immersion Day**

*Getting Started with Elastic Beanstalk*

# Table of Contents

[Overview 3](#_Toc494471427)

[Launch an Elastic Beanstalk Environment 3](#_Toc494471428)

[Deploy the Sample Application 8](#_Toc494471429)

[Update the Application's Configuration File 10](#_Toc494471430)

[Deploy the nodejs-tutorial.zip Source Bundle to Your Environment 11](#_Toc494471431)

[View & Test the Application 11](#_Toc494471432)

[View the Table 12](#_Toc494471433)

[Configure Your Environment for High Availability 12](#_Toc494471434)

[Clean Up 15](#_Toc494471435)

[Conclusion 15](#_Toc494471436)

# Overview

AWS Elastic Beanstalk is an even easier way for you to quickly deploy and manage applications in the AWS cloud. You simply upload your application and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling and application health monitoring.

At the same time, with Elastic Beanstalk, you retain full control over the AWS resources powering your application, and you can access the underlying resources at any time. Elastic Beanstalk leverages AWS services such as Amazon Elastic Cloud Compute (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon Simple Notification Service (Amazon SNS), Elastic Load Balancing, and Auto Scaling to deliver the same highly reliable, scalable, and cost-effective infrastructure that hundreds of thousands of businesses depend on today.

For this lab you will build a simple, scalable web-based customer signup form that is deployed to [AWS Elastic Beanstalk](http://aws.amazon.com/elasticbeanstalk/). The application stores data in [Amazon DynamoDB](http://aws.amazon.com/dynamodb/) and publishes notifications to the Amazon Simple Notification Service (SNS) when a customer fills out the form.

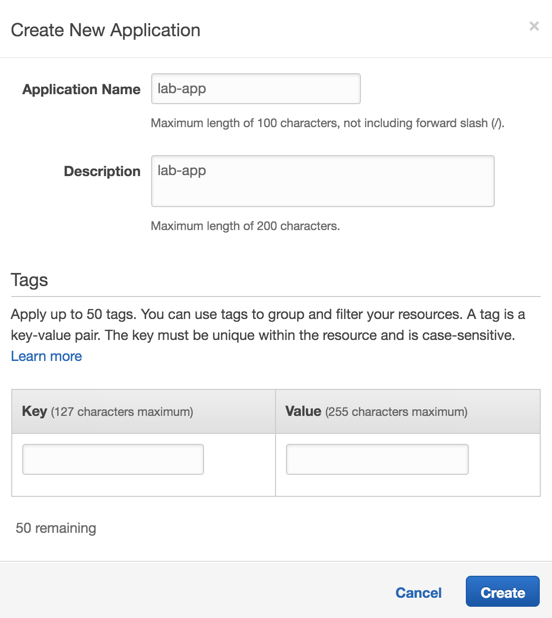
This lab will walk you through the following:

* Launch an Elastic Beanstalk Environment
* Add Permissions to Your Environment's Instances
* Deploy the Sample Application
* Update the Application's Configuration Files
* Configure Your Environment for High Availability
* Clean Up

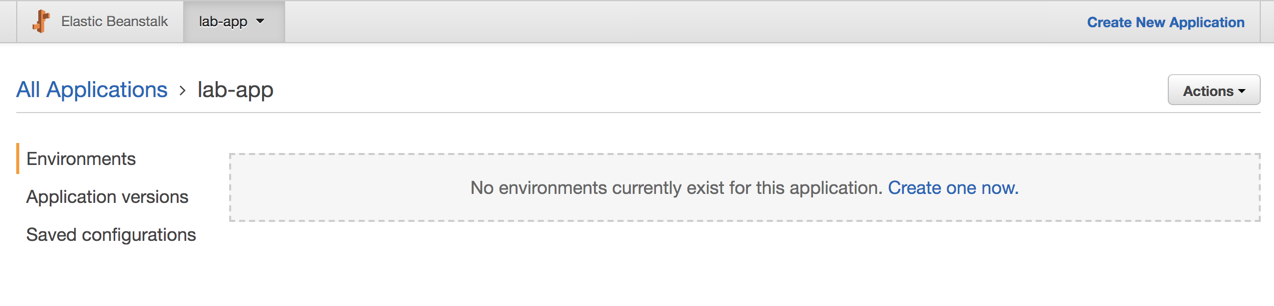
# Launch an Elastic Beanstalk Environment

AWS Elastic Beanstalk makes it easy to create new environments for your application. You can create and manage separate environments for development, testing, and production use, and you can deploy any version of your application to any environment.

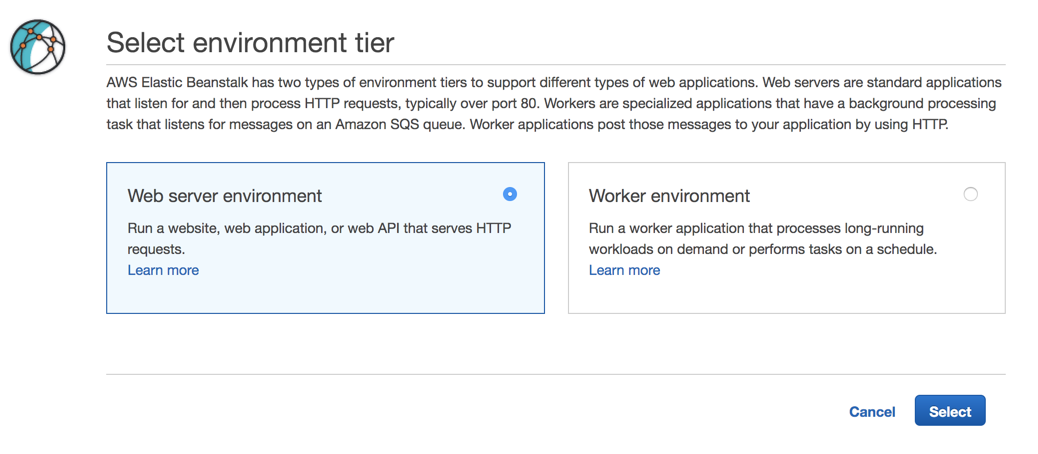
1. Download the sample application source bundle from GitHub: [eb-node-express-sample-v1.1.zip](https://github.com/awslabs/eb-node-express-sample/releases/download/v1.1/eb-node-express-sample-v1.1.zip) (https://bit.ly/2pubq2c)
2. Sign in to the AWS Management Console and open the AWS Elastic Beanstalk console at <https://console.aws.amazon.com/elasticbeanstalk>
3. Click **Create New Application** on the top right**.** The **Create New Application** dialog appears.



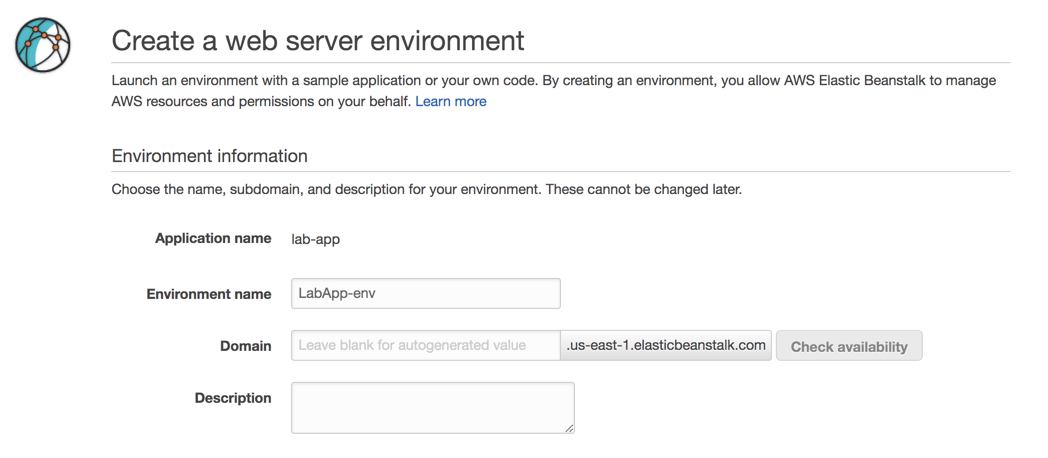
1. In the **Application name** field, type a name (e.g. **lab-app**).
2. In the **Description** field, type a description (e.g. **lab-app**).
3. Leave the Tags fields empty and click **Create**. The console displays the **All Applications** page.



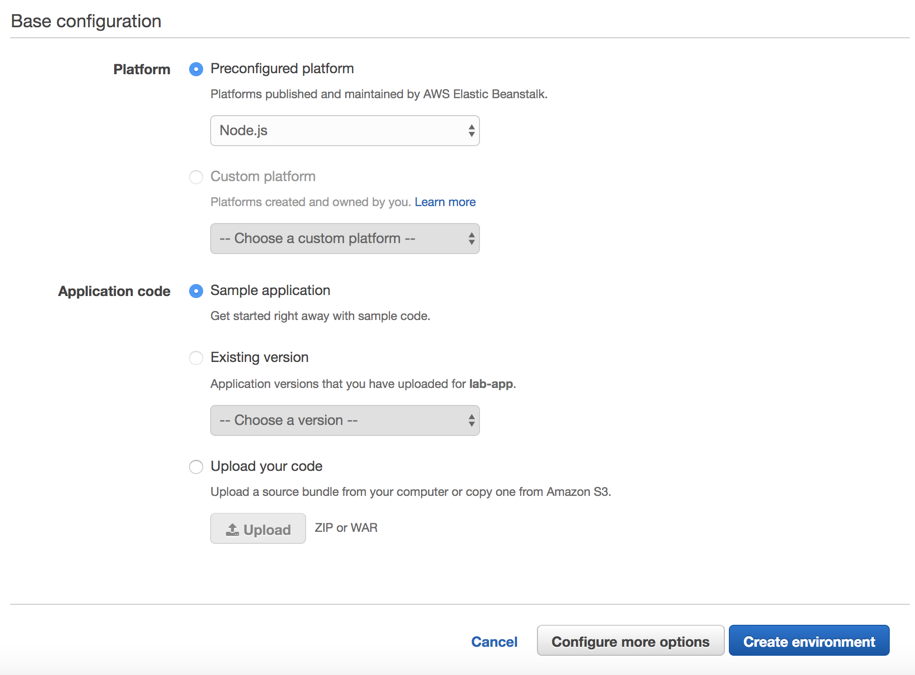
1. Click **Create one now** in the **Environments** view. The **Select an environment tier** dialog appears.



1. Select the **Web server environment** radio button and click **Select**. The **Create a web server environment** page is displayed.



1. Copy the value in the **Environment name** field and paste it into the **Domain** field. Click **Check availability.** If the domain is not available, add characters to the domain and try again or you can leave it blank and let Elastic Beanstalk autogenerate the domain name with your environment’s name. The Elastic Beanstalk domain name must be globally unique.
2. For **Platform**, select the **Preconfigured platform** radio button and select **Node.js** from the dropdown list.

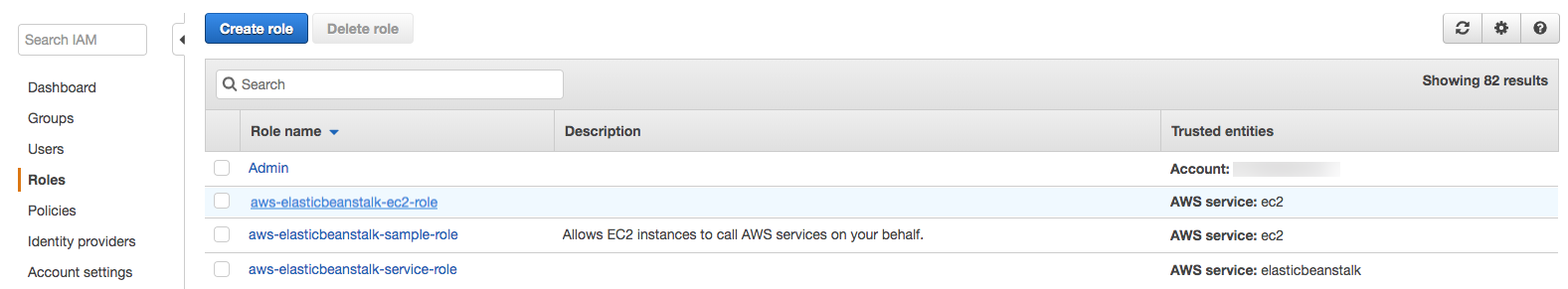


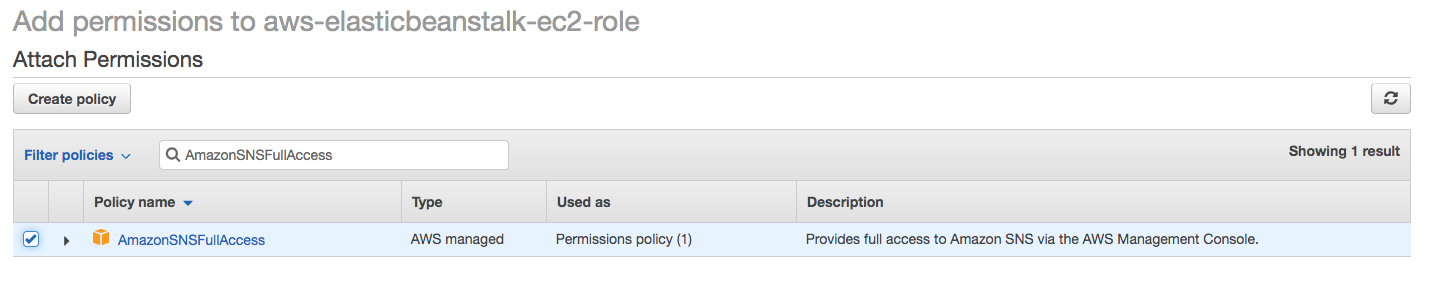
1. For **Application code**, select the **Sample application** radio button.
2. Click **Create environment**. This will take a few minutes to complete. During the environment creation process, the console tracks its progress and displays events.

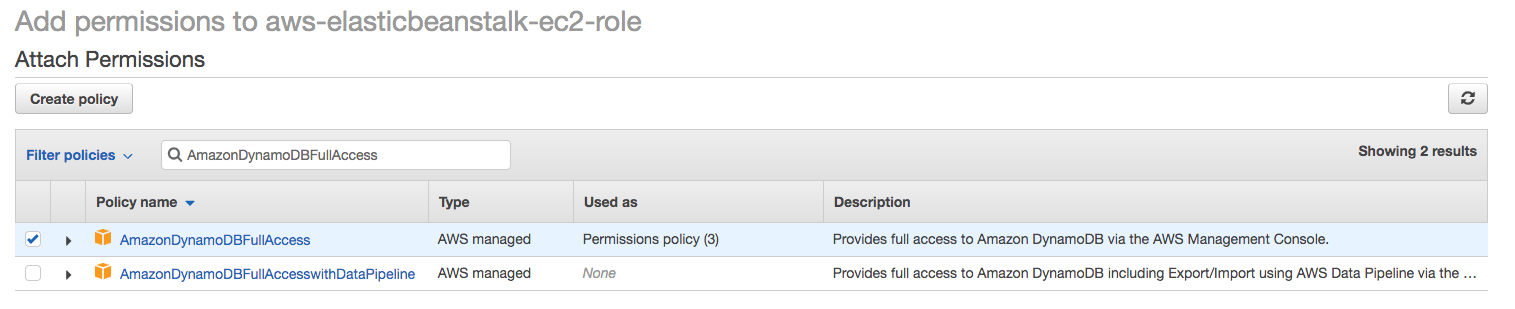
# Add Permissions to Your Environment’s Instances

The sample application uses instance permissions to write data to a DynamoDB table, and to send notifications to an Amazon SNS topic with the SDK for JavaScript in Node.js. Add the following managed policies to the default instance profile to grant the EC2 instances in your environment permission to access DynamoDB and Amazon SNS:

1. In the **AWS Management Console**, on the **Services** menu, under **Security, Identity & Compliance** headline, click **IAM.**
2. Click **Roles** in the left navigation.
3. Click **aws-elasticbeanstalk-ec2-role** in the list



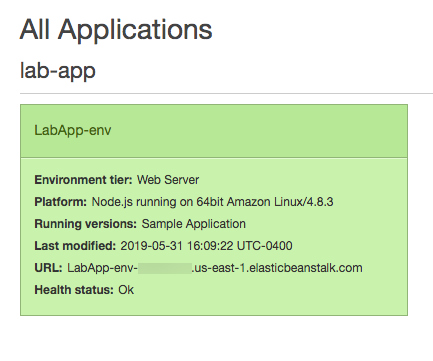
1. On the Permissions tab, click **Attach policies**.
2. Select the managed policies for the additional services that your application uses: **AmazonSNSFullAccess** and **AmazonDynamoDBFullAccess**. You can search for these policies by typing their names into **Search** field. After finding them, click the checkbox.



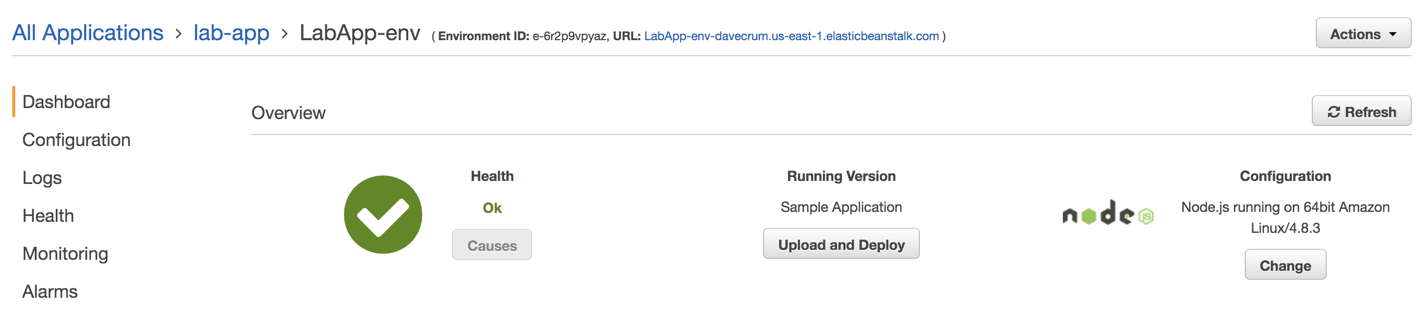
1. Click **Attach policy**.

# Deploy the Sample Application

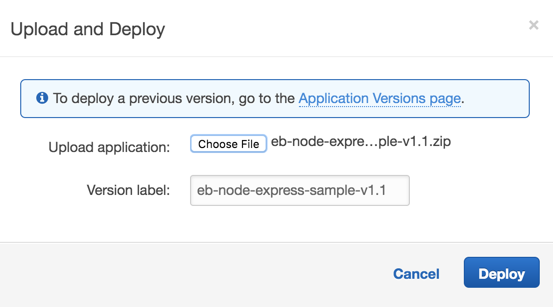
1. On the **Services** menu, under **History** headline, click **Elastic Beanstalk.**
2. Click on the environment (LabApp-env) created for lab-app.



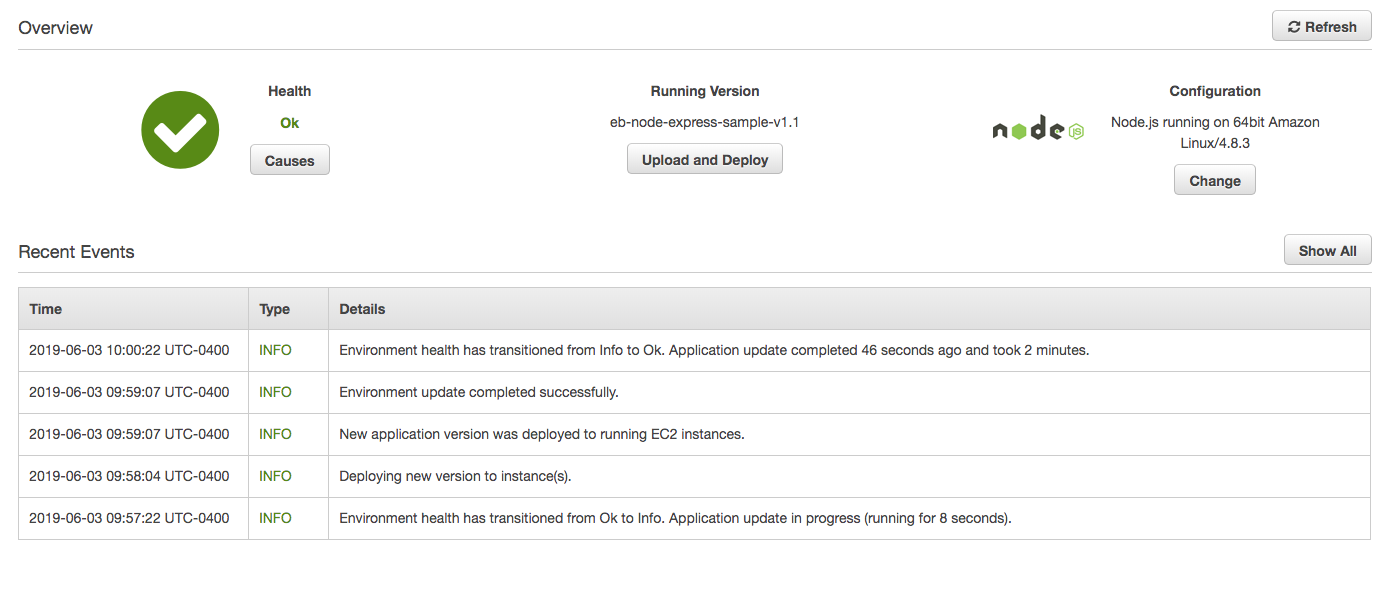
1. Click **Upload and Deploy.**



1. Click **Choose File.**



1. Browse for the downloaded application file and click **Deploy.** The deployment will take a few minutes to complete. You can monitor the progress on the dashboard page.



# Update the Application's Configuration File

In this step, we will update the configuration file in the application source to define our email address.

Linux/Mac:

1. Extract the project files from the source bundle:

~$ **mkdir nodejs-tutorial**

~$ **cd nodejs-tutorial**

~/nodejs-tutorial$ **unzip ~/Downloads/eb-node-express-sample-v1.1.zip**

1. Open **.ebextensions/options.config** and change the value of the following setting:
   * **NewSignupEmail** – Your email address.

This configures the email address that the Amazon SNS topic uses for notifications.

1. Create a source bundle from the modified code.

~/nodejs-tutorial$ **zip nodejs-tutorial.zip -r \* .[^.]\***

Windows PowerShell:

1. Extract the project files from the source bundle:

PS D:\Users\workshop\Downloads> **mkdir nodejs-tutorial**

PS D:\Users\workshop\Downloads> **cd nodejs-tutorial**

PS D:\Users\workshop\Downloads\nodejs-tutorial> **Expand-Archive -Path D:\Users\workshop\Downloads\eb-node-express-sample-v1.1.zip -DestinationPath .**

1. Open **.ebextensions/options.config** and change the value of the following setting:

PS D:\Users\workshop\Downloads\nodejs-tutorial> **powershell\_ise.exe .\.ebextensions\options.config**

* + **NewSignupEmail** – Your email address.

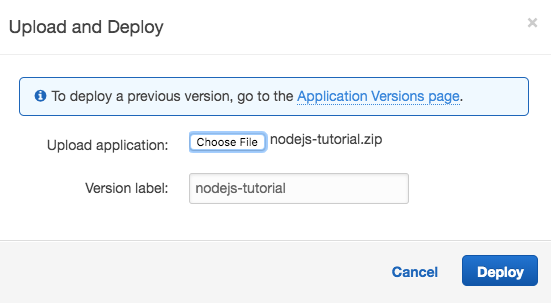
This configures the email address that the Amazon SNS topic uses for notifications.

1. Create a source bundle from the modified code.

PS D:\Users\workshop\Downloads\nodejs-tutorial> **Compress-Archive -Path D:\Users\workshop\Downloads\nodejs-tutorial\\* -DestinationPath D:\Users\workshop\Downloads\nodejs-tutorial\nodejs-tutorial**

# Deploy the nodejs-tutorial.zip Source Bundle to Your Environment

1. On the AWS **Services** menu, under **History** headline, click **Elastic Beanstalk.**
2. Click on the environment (LabApp-env) created for lab-app.
3. Click **Upload and Deploy.**
4. Click **Choose File.**

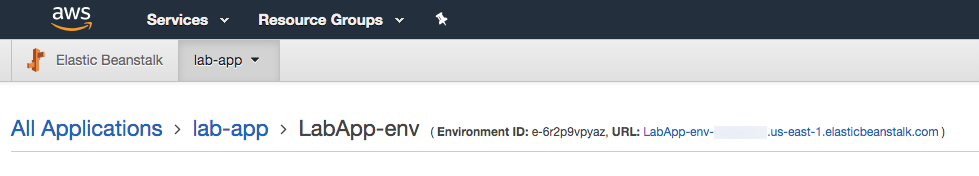


1. Browse for the application zip file you just created (**nodejs-tutorial.zip**) and click **Deploy.** The deployment will take a few minutes to complete. You can monitor the progress on the dashboard page.
2. You will receive an email with subject **AWS Notification - Subscription Confirmation.** Click the **Confirm subscription** link in the email body. When you deploy the application, Elastic Beanstalk updates the configuration of the Amazon SNS topic.



# View & Test the Application

1. On the AWS **Services** menu, under **History** headline, click **Elastic Beanstalk.**
2. Click on the environment (LabApp-env) created for lab-app.
3. You will see the URL of your application at the top. Click on the URL to open the application.



1. Click the **Sign up today** button and fill out the form. You should receive an email message for each entry you submit.

# View the DynamoDB Table

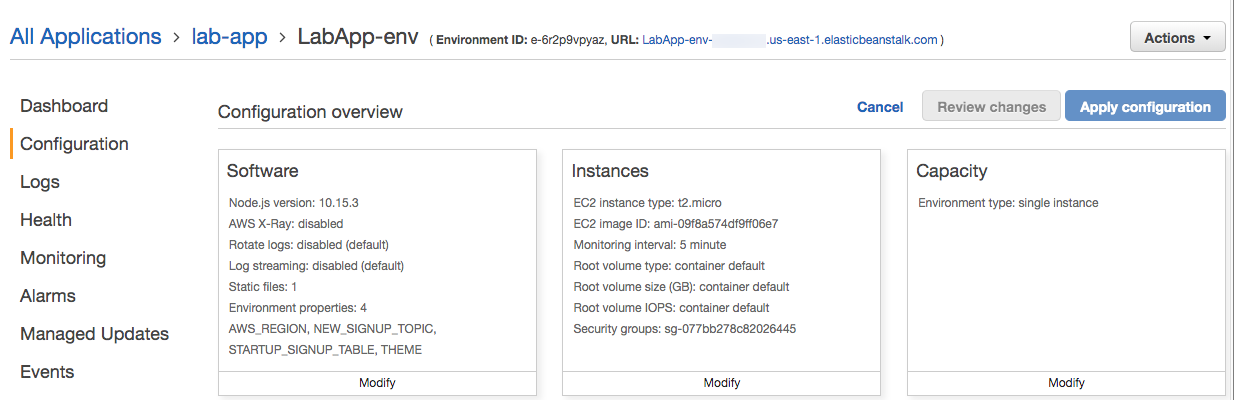
You can add AWS Elastic Beanstalk configuration files (.ebextensions) to your web application's source code to configure your environment and customize the AWS resources that it contains. The sample application includes [configuration files](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ebextensions.html) **(.ebextensions/create-dynamodb-table)** that create the DynamoDB table used by the application.

1. Open the [Tables page](https://console.aws.amazon.com/dynamodb/home?#tables:) in the DynamoDB console.
2. Find the table that contains **StartupSignupsTable** text in its name.
3. Select the table, choose the **Items** tab, and then click **Start search** to view all items in the table.
4. To get more items in the table, go back to the web application and fill out the form again. Then come back to the DynamoDB table and refresh the page.

# Configure Your Environment for High Availability

Finally, configure your environment's Auto Scaling group with a higher minimum instance count. Run at least two instances at all times to prevent the web servers in your environment from being a single point of failure and to allow you to deploy changes without taking your site out of service.

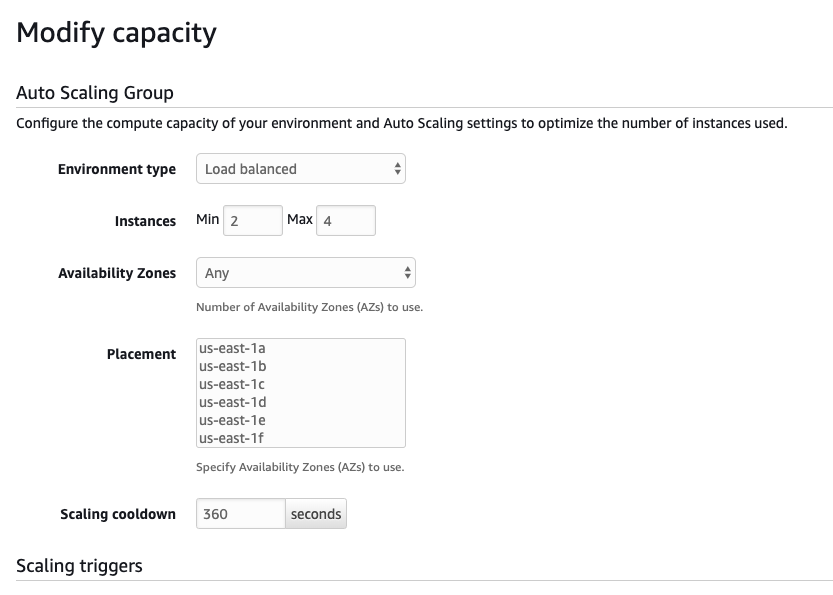
1. On the AWS **Services** menu, under **History** headline, click **Elastic Beanstalk.**
2. Click on the environment (**LabApp-env**) created for lab-app.
3. Choose **Configuration** from the menu on the left.
4. In the **Capacity** section, click the **Modify** link.



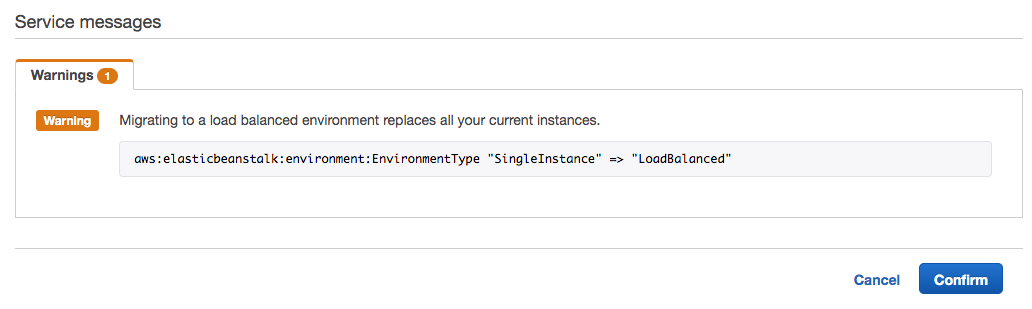
1. In the Auto Scaling Group section, configure the following settings.

**Environment type** – Select **Load balanced**.

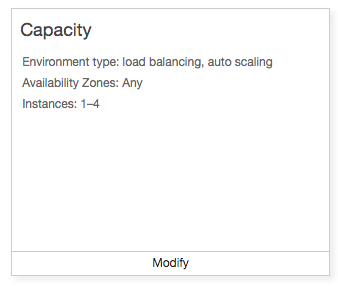
**Min instances** – **2**



1. Scroll to the bottom, click **Apply**, then click **Confirm**.

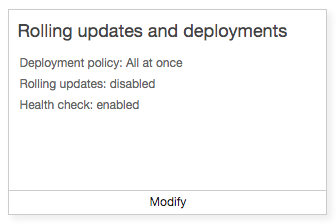


1. Wait until Elastic Beanstalk is done with updating your environment.
2. Choose **Configuration** from the left navigation and note the **Capacity** section has changed.

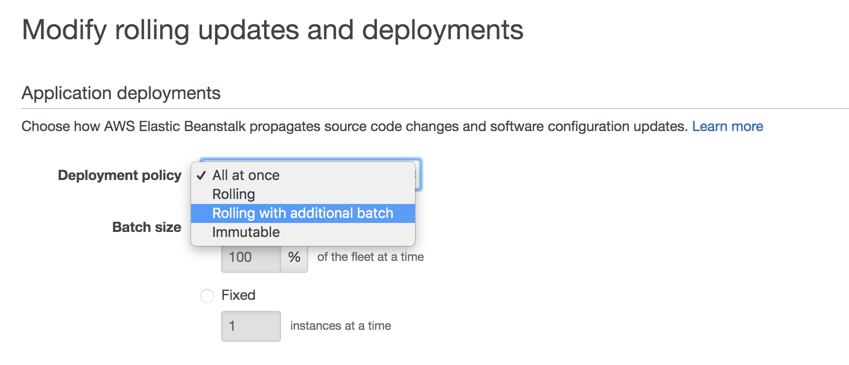


# Enable rolling deployments

1. While on the Configuration page, click **Modify** in the **Rolling updates and deployments** section.



1. For Deployment policy, select **Rolling with additional batch**.
2. Scroll to the bottom and click **Apply**.



# Perform another deployment (Refer to Section: [Deploy the nodejs-tutorial.zip Source Bundle to Your Environment](#_Deploy_the_nodejs-tutorial.zip)) to verify your deployment setup is as expected ([EC2](https://console.aws.amazon.com/ec2#Instances), [load balancer](https://console.aws.amazon.com/ec2#LoadBalancers:), and [auto scaling](https://console.aws.amazon.com/ec2/autoscaling/home#AutoScalingGroups:) dashboards will reflect the deployment approach).

**Note:**

Elastic Beanstalk creates an Amazon S3 bucket named **elasticbeanstalk-***region*-*account-id* for each region in which you create environments. Elastic Beanstalk uses this bucket to store objects, including deployment packages and temporary configuration files, that are required for the proper operation of your application.

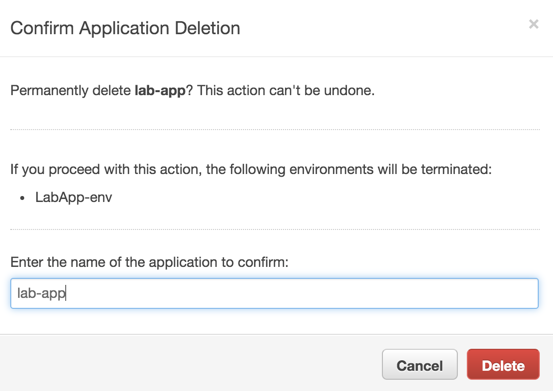
You can see these objects by going to the [S3 console](https://s3.console.aws.amazon.com/s3/) and inspecting the contents of the bucket beginning with the name **elasticbeanstalk-**.

# Clean Up

**To terminate your Elastic Beanstalk environment:**

When you finish working with Elastic Beanstalk, you can terminate your environment. Elastic Beanstalk terminates all AWS resources associated with your environment, such as Amazon EC2 instances, database instances, load balancers, security groups, and alarms.

1. On the AWS **Services** menu, under **History** headline, click **Elastic Beanstalk**.
2. Choose **Actions**, and then choose **Delete application**.
3. In the **Confirm Application Deletion** dialog, enter **lab-app** in the application field and click **Delete**.



1. Click on the environment (LabApp-env) to view progress of the deletion.

**To remove instance profile permissions:**

When you create resources outside of Elastic Beanstalk, it is completely independent of Elastic Beanstalk and your Elastic Beanstalk environments, and will not be terminated by Elastic Beanstalk.

1. Open the [**Roles** page](https://console.aws.amazon.com/iam/home#roles) in the IAM console.
2. Choose **aws-elasticbeanstalk-ec2-role.**
3. Remove the following AWS managed policies you attached earlier (Refer to Section: [**Add Permissions to Your Environment’s Instances**](#_Add_Permissions_to)).

* AmazonSNSFullAccess
* AmazonDynamoDBFullAccess

# Conclusion

In this lab you have walked through the process of deploying a sample Node.js application that uses the AWS SDK for JavaScript in Node.js to interact with Amazon DynamoDB. You learned basic operations to deploy and update an application using AWS Elastic Beanstalk. Finally, you learned how to configure your environment for high availability.