



## **Immersion Day**

*Getting Started with Elastic Beanstalk*

---

**September 2018**

## Table of Contents

Overview.....	3
Launch an Elastic Beanstalk Environment.....	3
Deploy the Sample Application.....	7
Update the Application's Configuration File .....	9
Deploy the nodejs-tutorial.zip Source Bundle to Your Environment .....	10
View & Test the Application .....	10
View the Table.....	11
Configure Your Environment for High Availability .....	11
Clean Up.....	14
Conclusion.....	15

## 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 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.

This lab will walk you through the following:

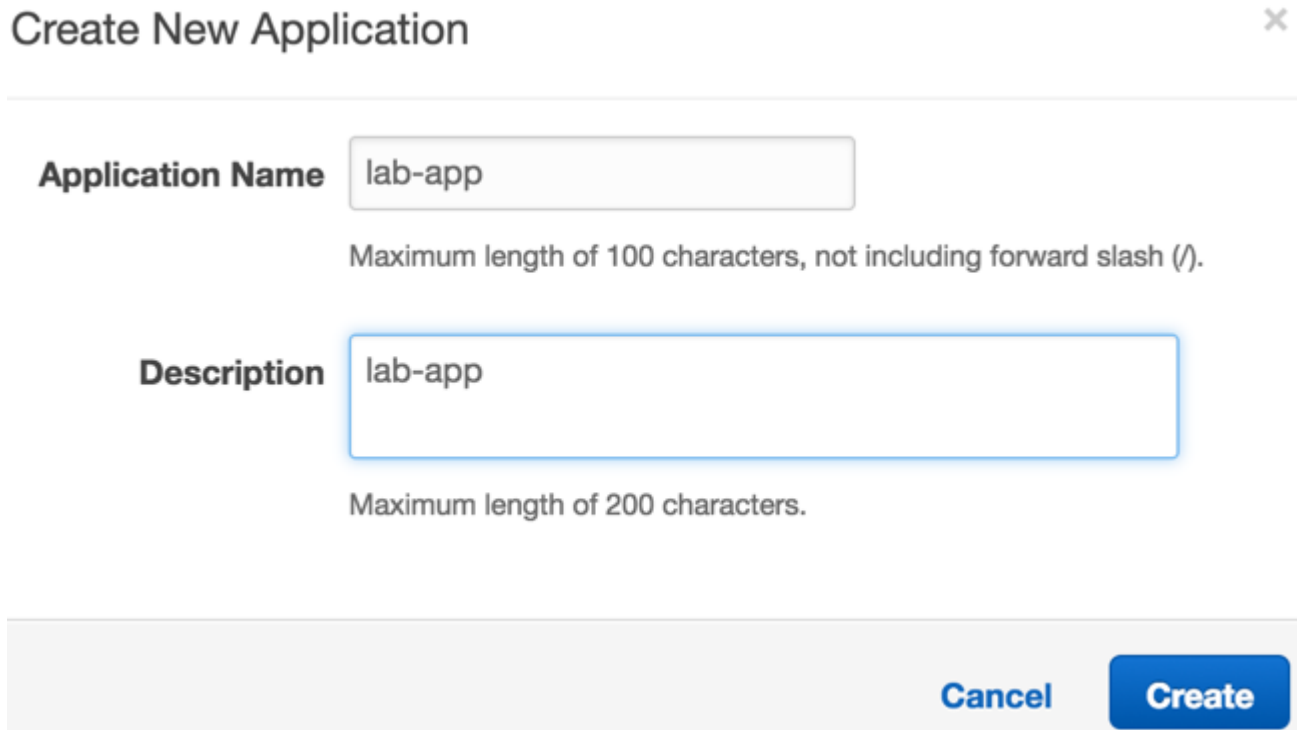
- Launch an Elastic Beanstalk Environment
- Add Permissions to Your Environment's Instances
- Deploy the Sample Application
- Create a DynamoDB Table
- Update the Application's Configuration Files
- Configure Your Environment for High Availability
- Clean Up

## 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://bit.ly/2pubq2c)  
<https://bit.ly/2pubq2c>
2. Sign into 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 box appears.



The dialog box is titled "Create New Application" with a close button (X) in the top right corner. It contains two input fields: "Application Name" and "Description". The "Application Name" field has the text "lab-app" and a note below it stating "Maximum length of 100 characters, not including forward slash (/)". The "Description" field also has the text "lab-app" and a note below it stating "Maximum length of 200 characters." At the bottom right, there are two buttons: "Cancel" and "Create".

4. In the **Application name** box, type a name (ie: **lab-app**)
5. In the **Description** box, type a description (ie: **lab-app**) and click **Create**. The console displays **All Applications** page.



The console page shows the "All Applications" view for the application "lab-app". The breadcrumb is "All Applications > lab-app". On the left, there are three tabs: "Environments", "Application versions", and "Saved configurations". The "Environments" tab is selected. The main content area shows a message: "No environments currently exist for this application. [Create one now.](#)". There is an "Actions" dropdown menu in the top right corner.

6. Click **Create one now** in **Environments** tab. The **Choose an environment tier** dialog box appears.

## 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](#)

7. Select **Web server environment** radio button and click **Select** button. The **Create a new environment** page is displayed

Elastic Beanstalk lab-app

## Create a new 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 lab-app

Environment name LabApp-env

Domain Leave blank for autogenerated value .us-east-1.elasticbeanstalk.com [Check availability](#)

Description

### Base configuration

Tier Web Server (Choose tier)

Platform ☒ Preconfigured platform

8. Copy the value in **Environment name** box and paste it into Domain box. Click **Check availability**. If domain is not available add random string to domain and try again.

9. In the **Predefined configuration** drop-down list, in the **Preconfigured** section, click **Node.js**.

### Base configuration

---

**Tier** Web Server [\(Choose tier\)](#)

**Platform** ☒ Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Node.js

☐ Custom platform **NEW**

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

-- Choose a custom platform --

**Application code** ☒ Sample application

Get started right away with sample code.

☐ Existing version

Application versions that you have uploaded for **lab-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](#)

10. Select **Sample application** radio button

11. Click **Create Environment**

## Add Permissions to Your Environment's Instances

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



4. On the Permissions tab, click **Attach Policy**.
5. 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** box. After finding them, click the checkbox.

### Attach policy

Filter: Policy type		Q AmazonDynamoDBFullAccess		Showing 2 results	
	Policy name	Type	Attachments	Description	
<input checked="" type="checkbox"/>	AmazonDynamoDBFullAccess	AWS managed	0	Provides full access to Amazon DynamoDB via the AWS Management Console.	
<input type="checkbox"/>	AmazonDynamoDBFullAccesswith...	AWS managed	0	Provides full access to Amazon DynamoDB including Export/Import using AWS ...	

### Attach policy

Filter: Policy type		Q AmazonSNSFullAccess		Showing 1 result	
	Policy name	Type	Attachments	Description	
<input checked="" type="checkbox"/>	AmazonSNSFullAccess	AWS managed	0	Provides full access to Amazon SNS via the AWS Management Console.	

6. 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


## All Applications

### lab-app

**LabApp-env**

**Environment tier:** Web Server  
**Platform:** 64bit Amazon Linux 2017.03 v4.3.0  
running Node.js  
**Running versions:** Sample Application  
**Last modified:** 2017-09-26 18:38:35 UTC+0300  
**URL:** LabApp-env.wgmzn7sgn7.us-east-1.elastic...


3. Click **Upload and Deploy**

 Elastic Beanstalk **lab-app** ▼

[All Applications](#) > [lab-app](#) > [LabApp-env](#) (Environment ID: e-2hywmxrsaz, URL: LabApp-env.wgmzn7sgn7.us-east-1.ela

[Dashboard](#)  
[Configuration](#)  
[Logs](#)  
[Health](#)  
[Monitoring](#)  
[Alarms](#)

Overview



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

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

4. Click **Choose File**



## Upload and Deploy



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

Upload application:

Choose File

eb-node-expre...ple-v1.1.zip

Version label:

eb-node-express-sample-v1.1

Cancel

Deploy

5. Browse downloaded application file and click **Deploy**

## Update the Application's Configuration File

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

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
```

2. 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.

3. Create a source bundle from the modified code.

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

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

1. On the **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**
5. Browse created application file(**nodejs-tutorial.zip**) and click **Deploy**

### Upload and Deploy



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

Upload application:

**Choose File**

nodejs-tutorial.zip

Version label:

nodejs-tutorial

**Cancel**

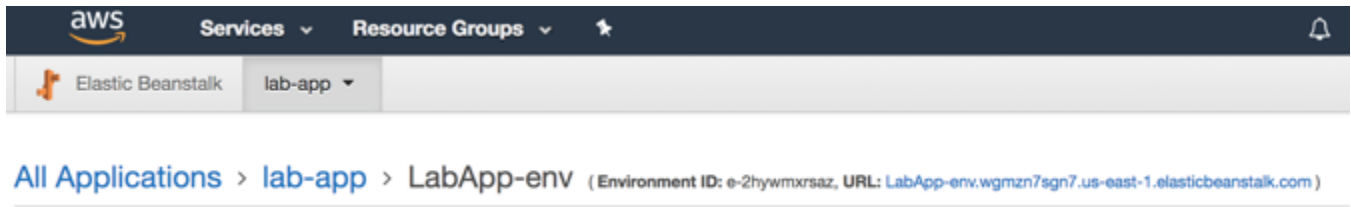
**Deploy**

6. You'll receive an email with subject **AWS Notification - Subscription Confirmation**. Click **Confirm Subscription** in the email body. When you deploy, Elastic Beanstalk updates the configuration of the Amazon SNS topic.

## View & Test the Application

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

3. You'll see URL of your application at the top. Copy the URL into your browser and test the application



## View the Table

1. Open the [Tables page](#) in the DynamoDB console.
2. Find the table contains **StartupSignupsTable** text in it's name.
3. Select the table, choose **Items**, and then choose **Start search** to view all items in the table.

## 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 **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

## Immersion Day

### Getting Started with AWS Elastic Beanstalk

Dashboard

Configuration

Logs

Health

Monitoring

Alarms

Managed Updates

Events

Tags

Configuration overview

Cancel Review changes Apply configuration

**Software**

Node.js version: 6.14.3  
AWS X-Ray: disabled  
Rotate logs: disabled (default)  
Log streaming: disabled (default)  
Static files: 1  
Environment properties: 4  
AWS\_REGION, NEW\_SIGNUP\_TOPIC,  
STARTUP\_SIGNUP\_TABLE, THEME

Modify

**Instances**

EC2 instance type: t1.micro  
EC2 image ID: ami-02bf1e347b5276f7b  
Monitoring interval: 5 minute  
Root volume type: container default  
Root volume size (GB): container default  
Root volume IOPS: container default  
Security groups: sg-0f6753d6994e03d26

Modify

**Capacity**

Environment type: single instance

Modify

#### 5. Select **Load balanced** as the **Environment Type**

## Modify capacity

### Auto Scaling Group

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

**Environment type** Load balanced

**Instances** Min 1 Max 4

**Availability Zones** Any

Number of Availability Zones (AZs) to use.

**Placement** ap-southeast-2a  
ap-southeast-2b  
ap-southeast-2c

Specify Availability Zones (AZs) to use.

**Scaling cooldown** 360 seconds

#### 6. Click **Apply** and Click **Confirm**

## Immersion Day

### Getting Started with AWS Elastic Beanstalk

#### Service messages

**Warnings** 1

**Warning** Migrating to a load balanced environment replaces all your current instances.  
`aws:elasticbeanstalk:environment:EnvironmentType "SingleInstance" => "LoadBalanced"`

Cancel **Confirm**

7. Wait until Elastic Beanstalk is done with updating your environment.
8. Choose **Configuration** and note **Capacity** section has changed

## Capacity

Environment type: load balancing, auto scaling  
Availability Zones: Any  
Instances: 1–4

Modify

9. Click the settings **modify** link and note that **Auto Scaling** configuration is added to the page.

## Modify capacity

### Auto Scaling Group

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

Environment type	<input type="text" value="Load balanced"/>
Instances	Min <input type="text" value="1"/> Max <input type="text" value="4"/>
Availability Zones	<input type="text" value="Any"/> Number of Availability Zones (AZs) to use.
Placement	<input type="text" value="ap-southeast-2a"/> <input type="text" value="ap-southeast-2b"/> <input type="text" value="ap-southeast-2c"/> Specify Availability Zones (AZs) to use.
Scaling cooldown	<input type="text" value="360"/> <input type="text" value="seconds"/>

## Enable rolling deployments

Click on Configuration then in the **Rolling updates and deployments** section click on **Modify**.

### Rolling updates and deployments

Deployment policy: All at once  
Rolling updates: disabled  
Health check: enabled

Modify

Select **Rolling with additional batch** in the deployment policy dropdown and click **Apply**.

## Modify rolling updates and deployments

### Application deployments

---

Choose how AWS Elastic Beanstalk propagates source code changes and software configuration updates. [Learn more](#)

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, there are labels for 'Deployment policy' and 'Batch size'. The 'Deployment policy' dropdown menu is open, showing four options: 'All at once' (checked), 'Rolling', 'Rolling with additional batch' (highlighted in blue), and 'Immutable'. Below the dropdown, the 'Batch size' section shows a value of '100' followed by a '%' symbol and the text 'of the fleet at a time'. Below that, there is a radio button labeled 'Fixed' and a value of '1' followed by the text 'instances at a time'.

Do another deployment to verify your deployment setup is as expected (EC2 & ELB, Autoscaling dashboards will reflect the deployment approach)

Worth also looking at the S3 bucket to see the versions of the app stored there.

## Clean Up

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 **Services** menu, under **History** headline, click **Elastic Beanstalk**
2. Choose **Actions**, and then choose **Delete application**.
3. In the **Confirm Delete Application** dialog box, click **Delete**

## 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 learn how to configure your environment for high availability.