aws

September 2017

Immersion Day

Getting Started with Elastic Beanstalk

Table of Contents

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.
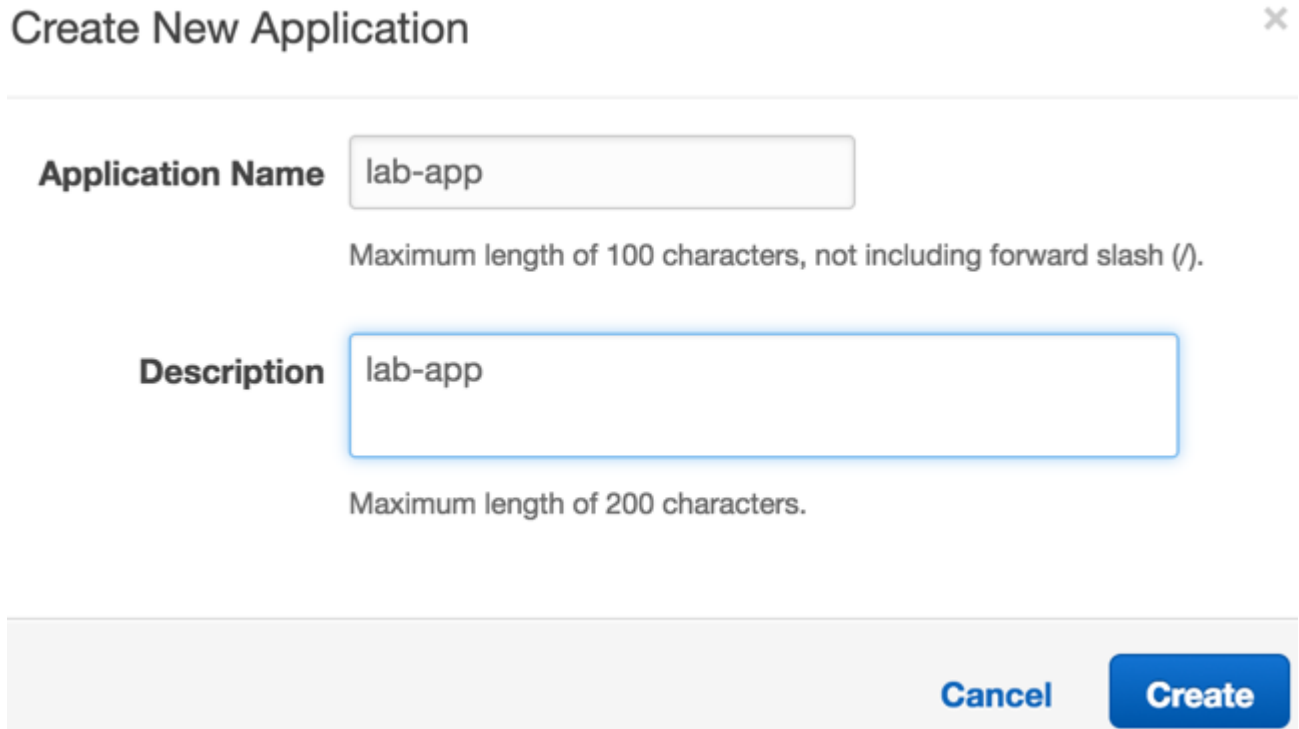
This lab will walk you through the following:

- 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

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.** The **Create New Application** dialog box appears.

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.

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

Choose an environment tier   ✕

● **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. In a worker environment, each instance runs a daemon process that reads messages from an Amazon SQS queue and relays the contents of the messages in a POST request to a path that you configure. Learn more.

Cancel   **Select**

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.

6. Click **Attach policy**

Deploy the Sample Application

Getting Started with AWS Elastic Beanstalk

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**

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 ✕

   ℹ 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
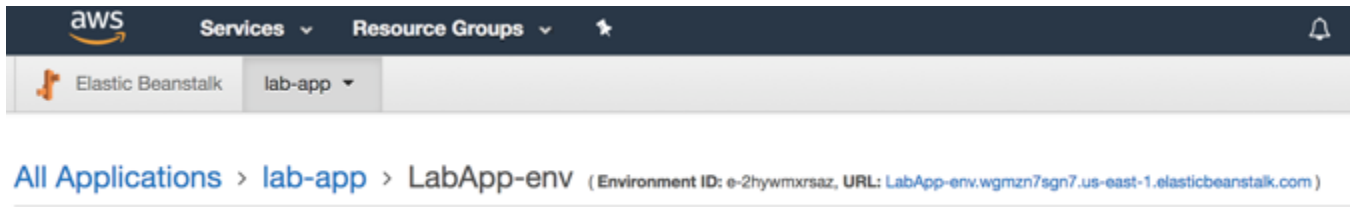
View the Table

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 **Scaling** section, click the settings icon (⚙)

All Applications > lab-app > LabApp-env (**Environment ID:** e-2hywmxrsaz,

Dashboard

Configuration

Logs

Health

Monitoring

Web Tier

Scaling ⚙

Environment type: Single instance

Custom Availability Zones: *blank*

5. Select **Load balancing, auto scaling** as the **Environment Type**

Environment Type

The following settings configure the availability settings of your environment to help reduce the costs for development activities.

Environment type: Load balancing, auto scaling ⇕ Learn more

Current status: 1 instance(s) in service, Min: **1**, Max: **1**

Auto scaling and trigger settings will be available after updating your environment type.

▶ Time-based Scaling

Cancel **Apply**

6. Click **Apply** and Click **Save**

⚠ EnvironmentType: Migrating to a load balanced environment replaces all your current instances.

Cancel **Save**

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

Web Tier

Scaling ⚙

Environment type: Load balanced, auto scaling

Number instances: 1 - 4

Scale based on Average network out

Add instance when > 6000000

Remove instance when < 2000000

9. Click the settings icon (⚙) and note that **Auto Scaling** configuration is added to the page.

Environment Type

The following settings configure the availability settings of your environment to help reduce the costs for development activities.

Environment type: Load balancing, auto scaling   ⇕ Learn more

Current status: 1 instance(s) in service, Min: 1, Max: 4

▾ Auto Scaling

Use the following settings to control auto scaling behavior. Learn more.

Minimum instance count: `1`   Minimum number of instances to run.

Maximum instance count: `4`   Maximum number of instances to run.

Availability Zones: Any ⇕ Number of Availability Zones to run in.

Custom Availability Zones: us-east-1a / us-east-1b / us-east-1c / us-east-1d   Specific Availability Zones to launch instances in.

Scaling cooldown (seconds):   `360`   The amount of time after a scaling activity before any further trigger-related scaling :

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.