Immersion Day

Getting Started with Elastic Beanstalk

July 2021

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 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. The application stores data in Amazon 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:

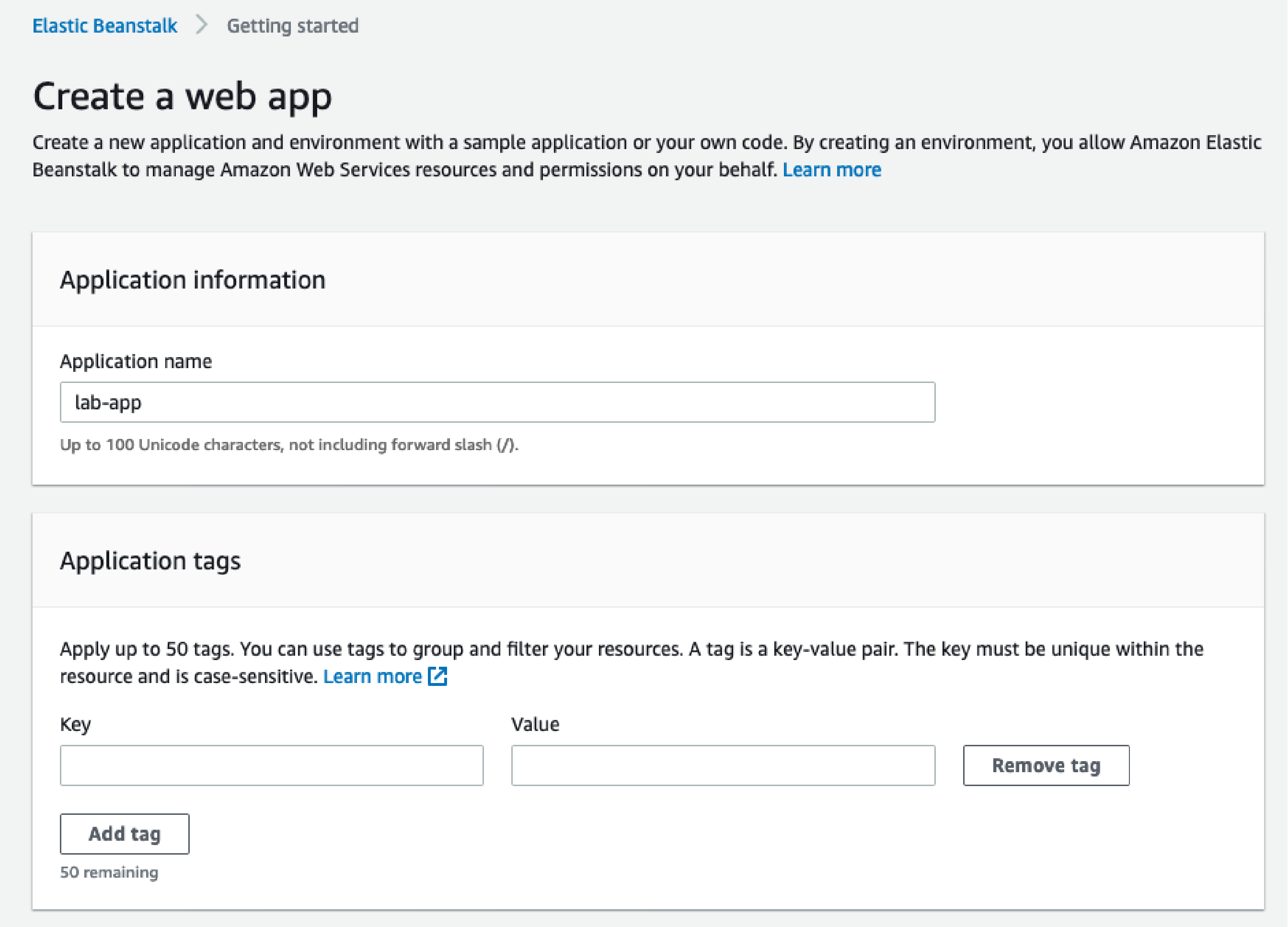
- Launch an Elastic Beanstalk Environment
- Add Permissions to Your Environment's Instances
- Deploy the Sample Application
- Update the Application's Configuration Files
- Configure Your Environment for High Availability
- Clean Up

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://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 Application** on the **top right.** The **Create a Web App** dialog appears.

4. In the **Application name** field, type a name (e.g. **lab-app**).
5. Leave the Tags fields empty.



Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. Learn more

Application information

Application name: lab-app

Up to 100 Unicode characters, not including forward slash (/).

Application tags

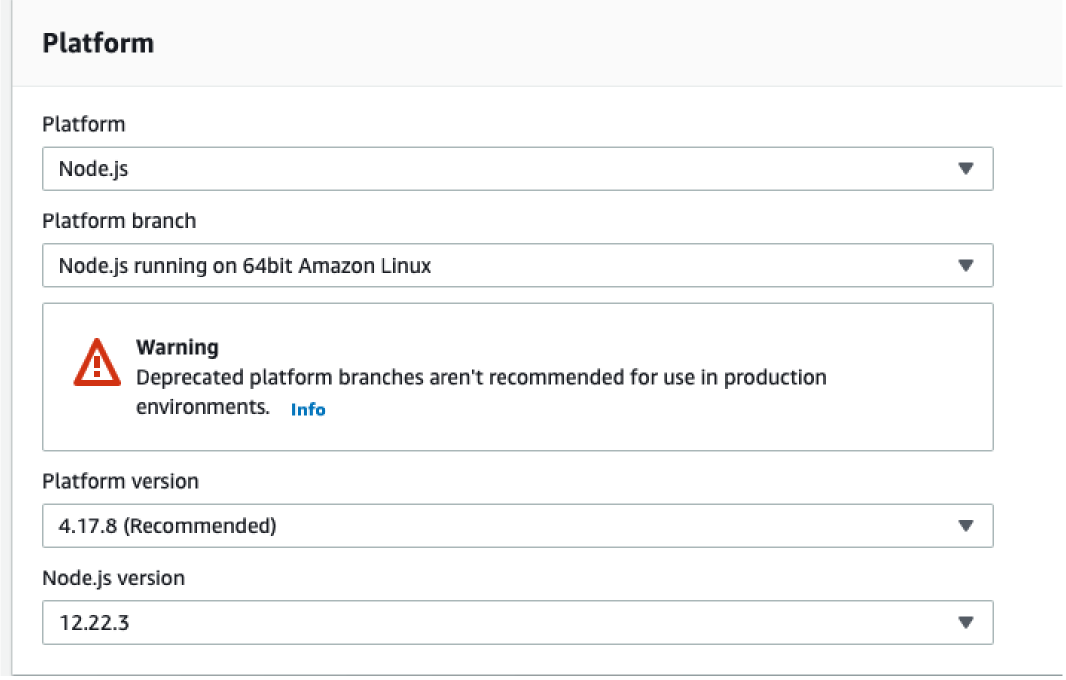
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

Key:

Value:

6. For **Platform**, Choose Node.js
7. For **Platform Branch**, select Node.js running on 64bit Amazon Linux, under **Depreciated** options.

   (**Note**: Do not select Amazon Linux 2. Application example used in this Lab was written using older Node.js branch)



Platform

Platform: Node.js

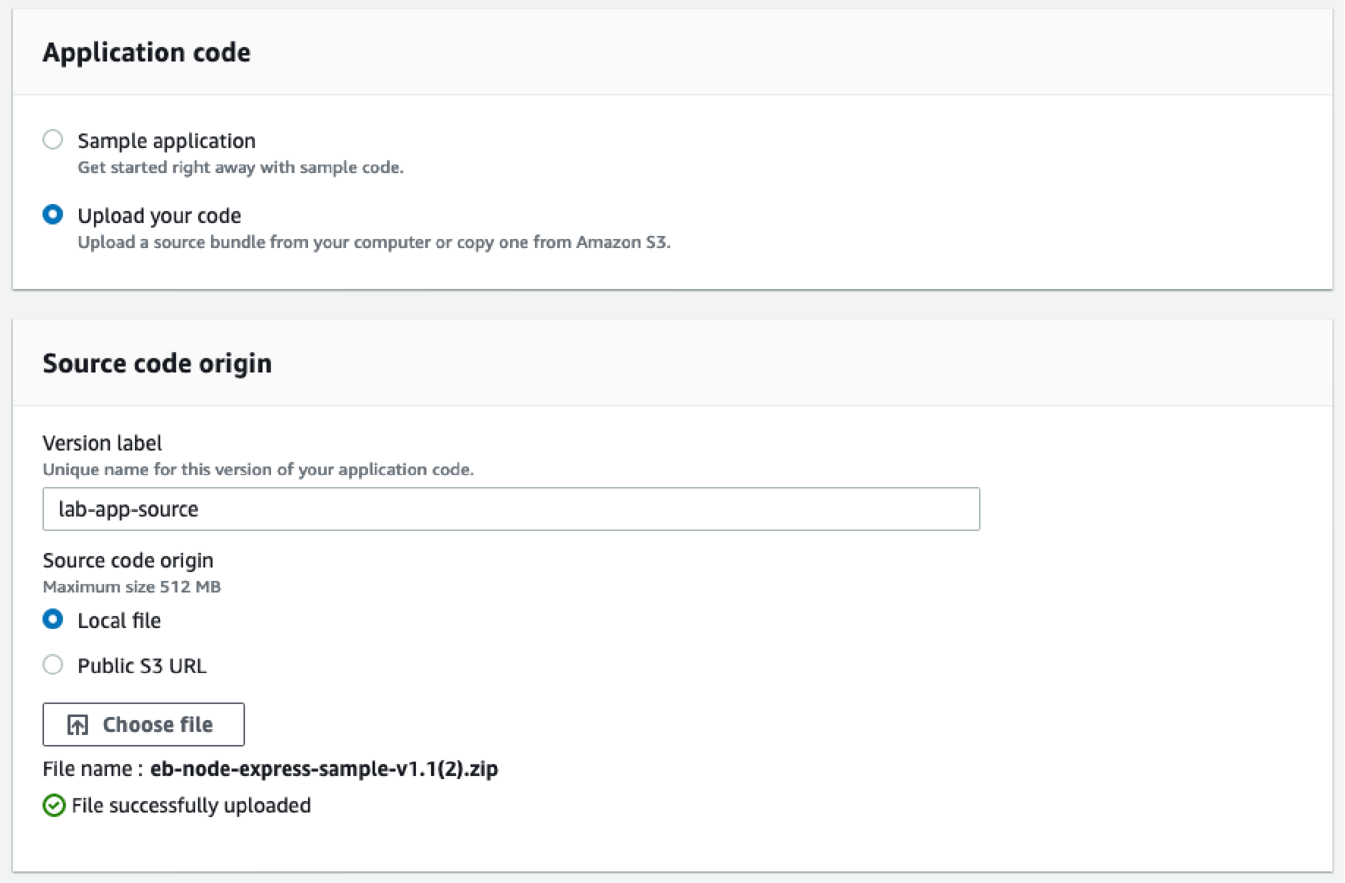
Platform branch: Node.js running on 64bit Amazon Linux

Warning
Deprecated platform branches aren't recommended for use in production environments. Info

Platform version: 4.17.8 (Recommended)

Node.js version: 12.22.3

8. For **Application Code**, choose **Upload** your code and select the .zip file you downloaded in step 1 above



Application code

○ Sample application
Get started right away with sample code.

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

Source code origin

Version label
Unique name for this version of your application code.
lab-app-source

Source code origin
Maximum size 512 MB

● Local file

○ Public S3 URL

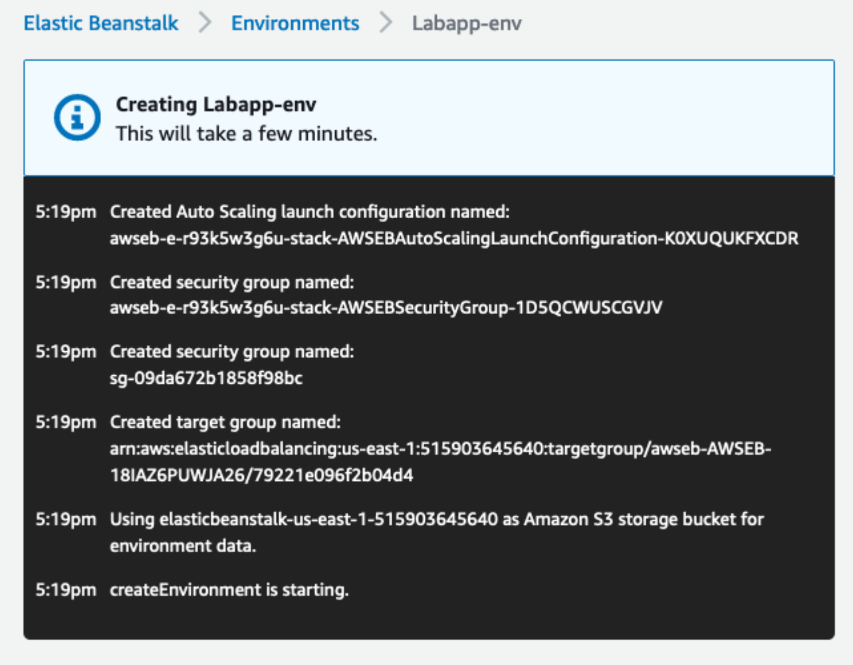
File name : **eb-node-express-sample-v1.1(2).zip**

✓ File successfully uploaded

9. Click the orange **Create application** button on bottom right of page.

10. It will take a few minutes for the application to be created. You can monitor progress as the environment is being launched.

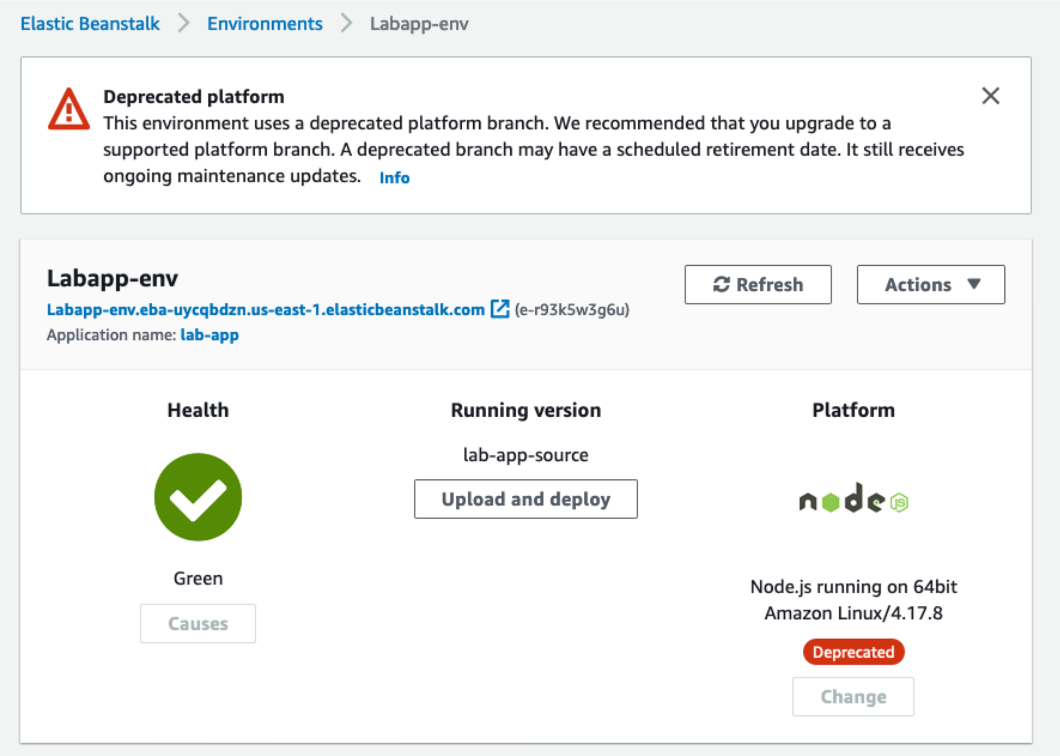
Examine the outputs. Elastic Beanstalk launches and configures the environment for your application to run. This includes an EC2 instance, Auto Scaling group, Elastic Load Balancer, and CloudWatch for logging.



Creating Labapp-env
This will take a few minutes.

```
5:19pm  Created Auto Scaling launch configuration named:
        awseb-e-r93k5w3g6u-stack-AWSEBAutoScalingLaunchConfiguration-K0XUQUKFXCDR

5:19pm  Created security group named:
        awseb-e-r93k5w3g6u-stack-AWSEBSecurityGroup-1D5QCWUSCGVJV

5:19pm  Created security group named:
        sg-09da672b1858f98bc

5:19pm  Created target group named:
        arn:aws:elasticloadbalancing:us-east-1:515903645640:targetgroup/awseb-AWSEB-
        18IAZ6PUWJA26/79221e096f2b04d4

5:19pm  Using elasticbeanstalk-us-east-1-515903645640 as Amazon S3 storage bucket for
        environment data.

5:19pm  createEnvironment is starting.
```

11. Once the deployment has completed, the AWS Console will refresh and show a Green Health check status as shown below:



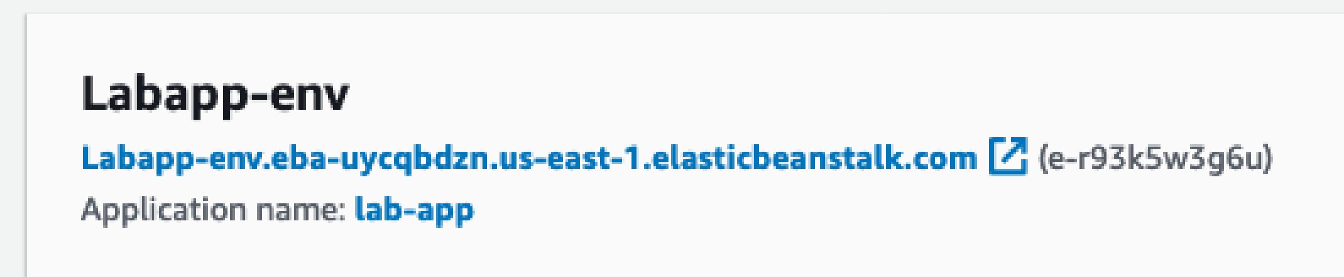
Elastic Beanstalk > Environments > Labapp-env

Deprecated platform
This environment uses a deprecated platform branch. We recommended that you upgrade to a supported platform branch. A deprecated branch may have a scheduled retirement date. It still receives ongoing maintenance updates. Info

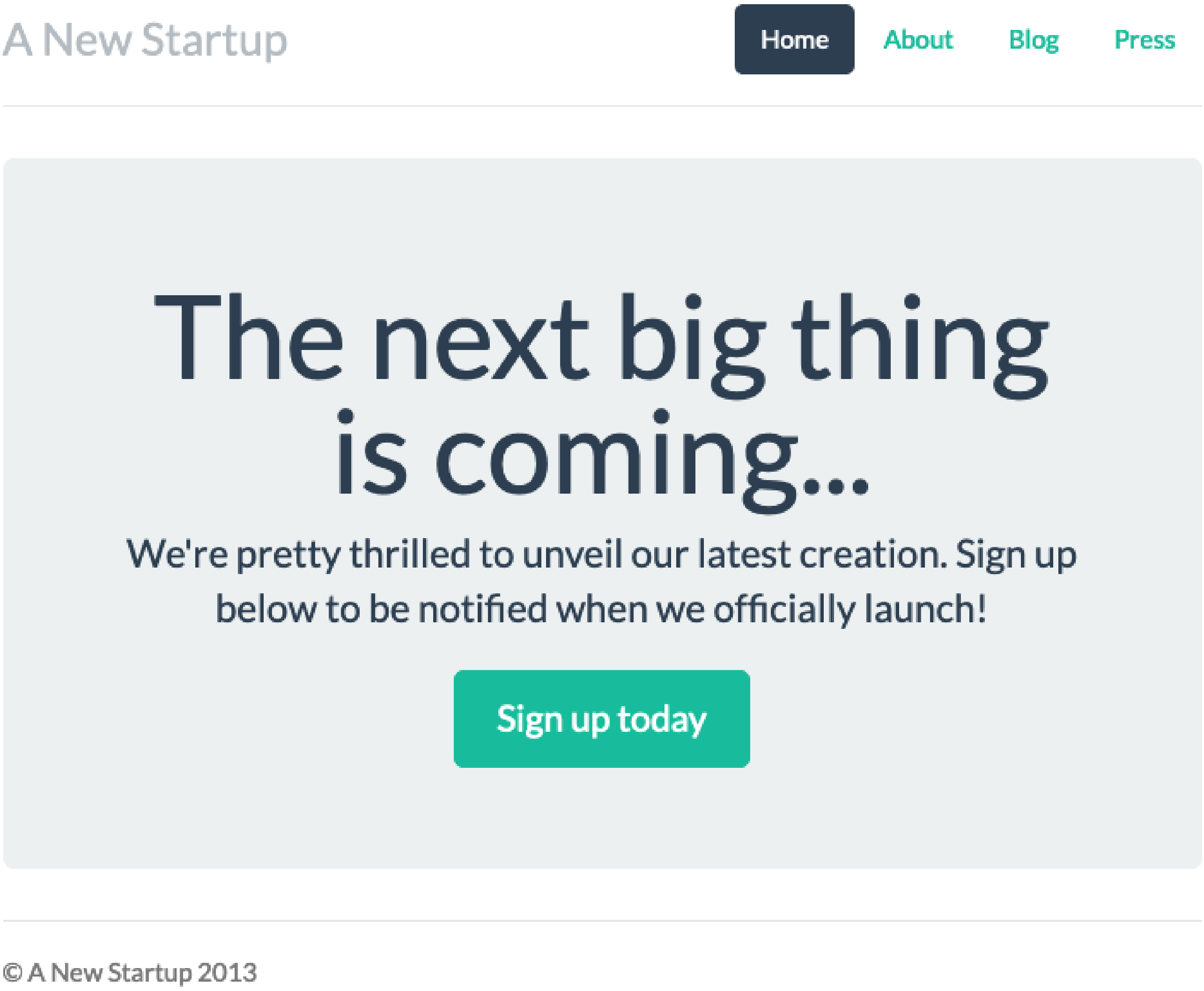
Labapp-env
Labapp-env.eba-uycqbdzn.us-east-1.elasticbeanstalk.com (e-r93k5w3g6u)
Application name: lab-app

Health	Running version	Platform
Green	lab-app-source	Node.js running on 64bit Amazon Linux/4.17.8
Causes	Upload and deploy	Deprecated Change

12. You can visit your deployed application by clicking on the URL displayed on the page.



Labapp-env
Labapp-env.eba-uycqbdzn.us-east-1.elasticbeanstalk.com (e-r93k5w3g6u)
Application name: lab-app



The next big thing is coming...

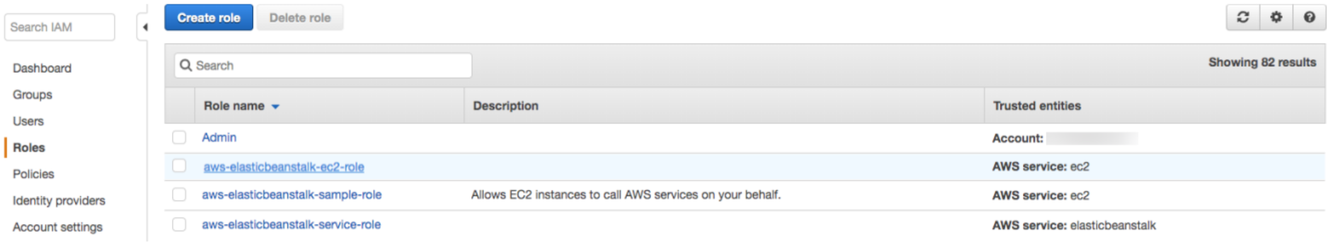
We're pretty thrilled to unveil our latest creation. Sign up below to be notified when we officially launch!

© A New Startup 2013

Add Permissions to Your Environment’s Instances

The sample application uses EC2 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:

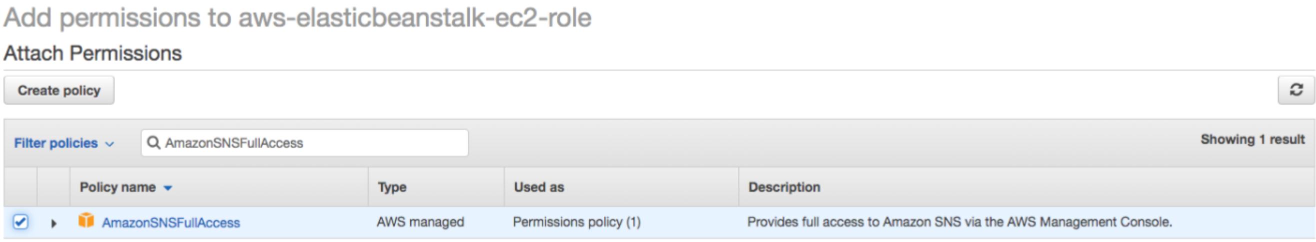
1. In the **AWS Management Console**, on the **Services** menu, under **Security, Identity & Compliance** headline, click **IAM.**
2. Click **Roles** in the left navigation pane.
3. Click **aws-elasticbeanstalk-ec2-role** in the list



Showing 82 results

	Role name	Description	Trusted entities
□	Admin		**Account:** [redacted]
□	aws-elasticbeanstalk-ec2-role		**AWS service:** ec2
□	aws-elasticbeanstalk-sample-role	Allows EC2 instances to call AWS services on your behalf.	**AWS service:** ec2
□	aws-elasticbeanstalk-service-role		**AWS service:** elasticbeanstalk

4. On the Permissions tab, click **Attach policies**.
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** field. After finding them, click the checkbox.



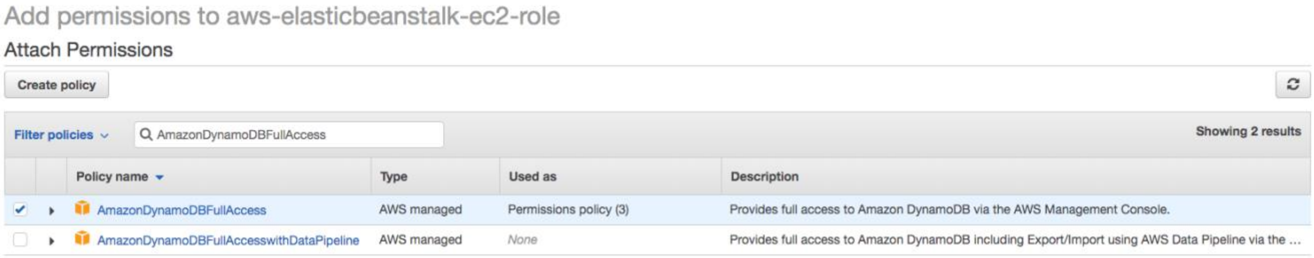
Add permissions to aws-elasticbeanstalk-ec2-role

Attach Permissions

AmazonSNSFullAccess

Showing 1 result

	Policy name	Type	Used as	Description
☑	AmazonSNSFullAccess	AWS managed	Permissions policy (1)	Provides full access to Amazon SNS via the AWS Management Console.



Add permissions to aws-elasticbeanstalk-ec2-role

Attach Permissions

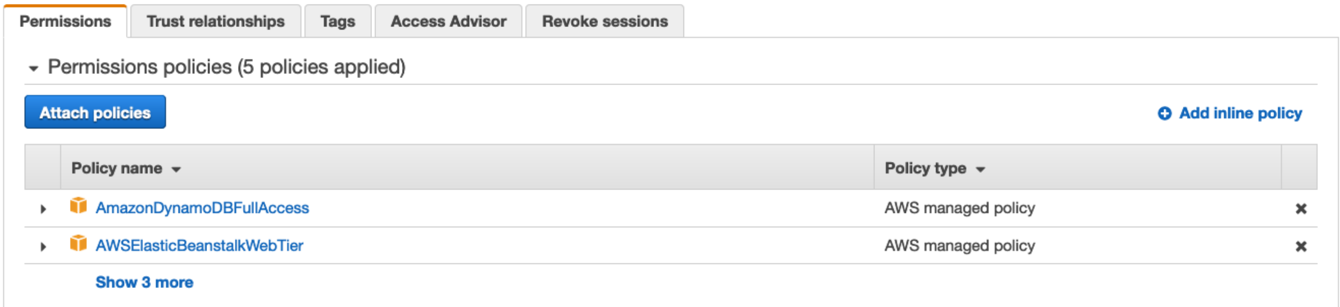
AmazonDynamoDBFullAccess

Showing 2 results

	Policy name	Type	Used as	Description
☑	AmazonDynamoDBFullAccess	AWS managed	Permissions policy (3)	Provides full access to Amazon DynamoDB via the AWS Management Console.
□	AmazonDynamoDBFullAccesswithDataPipeline	AWS managed	None	Provides full access to Amazon DynamoDB including Export/Import using AWS Data Pipeline via the ...

6. Click **Attach policy**.

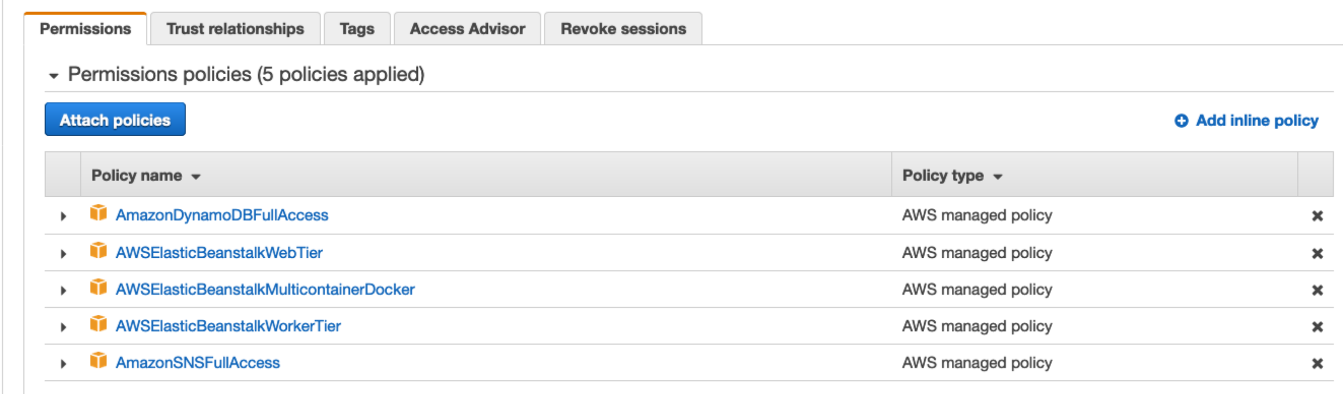
7. In the Permissions tab, validate both policies have been applied to IAM role. Click **Show more** if needed to get an expanded view of all policies applied.



Permissions policies (5 policies applied)

Policy name	Policy type	
AmazonDynamoDBFullAccess	AWS managed policy	✖
AWSElasticBeanstalkWebTier	AWS managed policy	✖

Show 3 more



Permissions policies (5 policies applied)

Policy name	Policy type	
AmazonDynamoDBFullAccess	AWS managed policy	✖
AWSElasticBeanstalkWebTier	AWS managed policy	✖
AWSElasticBeanstalkMulticontainerDocker	AWS managed policy	✖
AWSElasticBeanstalkWorkerTier	AWS managed policy	✖
AmazonSNSFullAccess	AWS managed policy	✖

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:

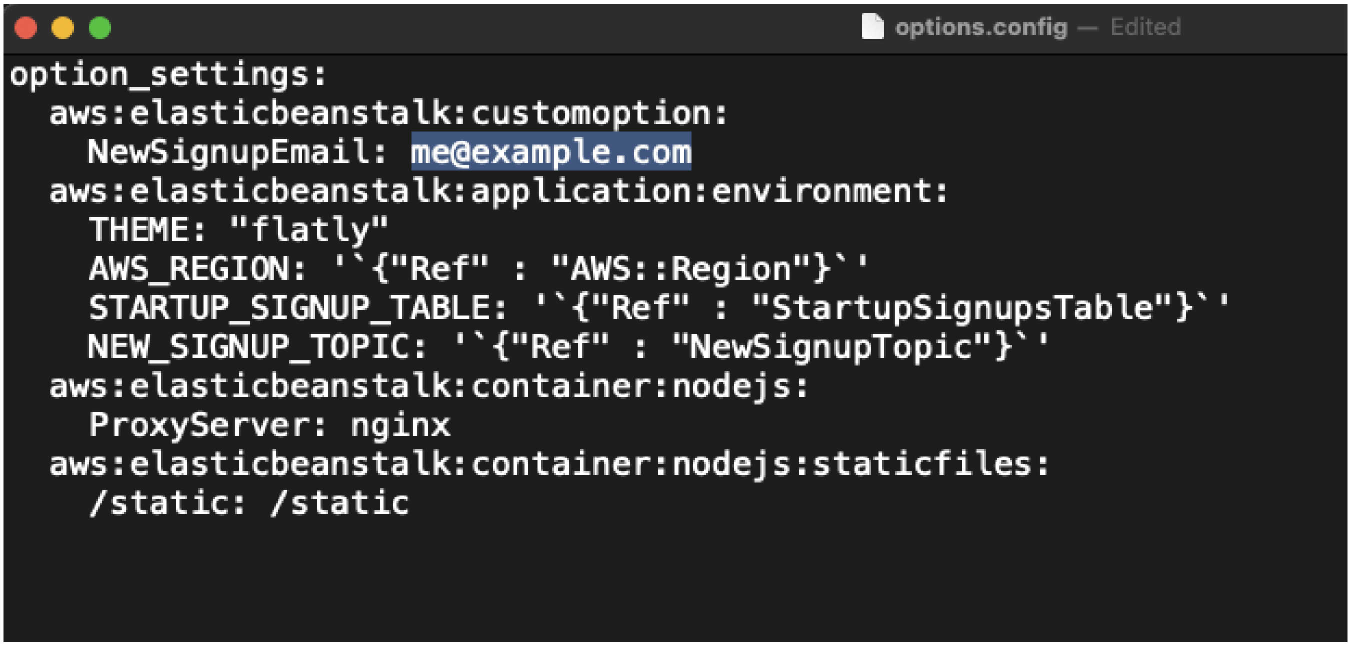
1. Extract the project files from the source bundle:

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

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.



```
option_settings:
  aws:elasticbeanstalk:customoption:
    NewSignupEmail: me@example.com
  aws:elasticbeanstalk:application:environment:
    THEME: "flatly"
    AWS_REGION: '`{"Ref" : "AWS::Region"}`'
    STARTUP_SIGNUP_TABLE: '`{"Ref" : "StartupSignupsTable"}`'
    NEW_SIGNUP_TOPIC: '`{"Ref" : "NewSignupTopic"}`'
  aws:elasticbeanstalk:container:nodejs:
    ProxyServer: nginx
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /static: /static
```

3. 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:

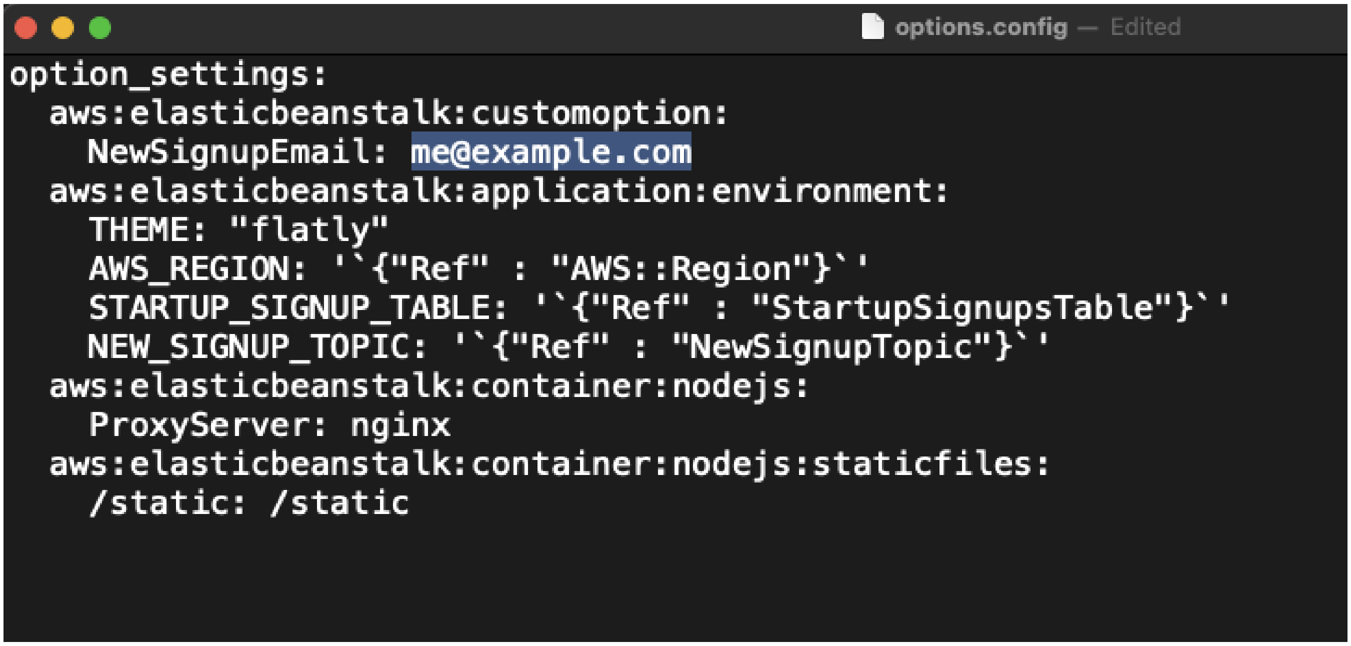
```
PS D:\Users\workshop\Downloads> mkdir nodejs-tutorial
PS D:\Users\workshop\Downloads> cd nodejs-tutorial
PS D:\Users\workshop\Downloads\nodejs-tutorial> Expand-Archive -Path
D:\Users\workshop\Downloads\eb-node-express-sample-v1.1.zip -DestinationPath .
```

2. 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.



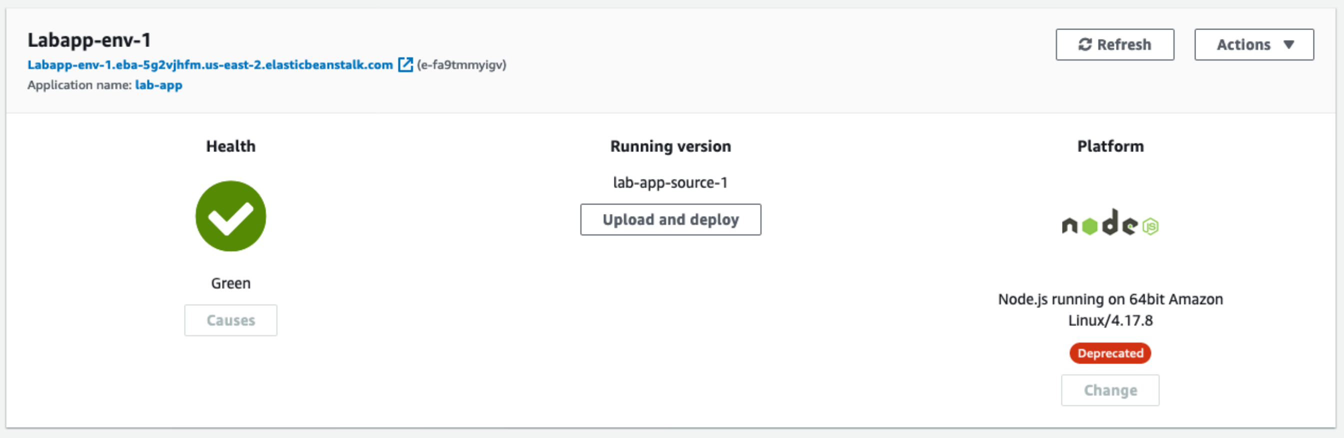
```
option_settings:
  aws:elasticbeanstalk:customoption:
    NewSignupEmail: me@example.com
  aws:elasticbeanstalk:application:environment:
    THEME: "flatly"
    AWS_REGION: '`{"Ref" : "AWS::Region"}`'
    STARTUP_SIGNUP_TABLE: '`{"Ref" : "StartupSignupsTable"}`'
    NEW_SIGNUP_TOPIC: '`{"Ref" : "NewSignupTopic"}`'
  aws:elasticbeanstalk:container:nodejs:
    ProxyServer: nginx
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /static: /static
```

3. 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
```

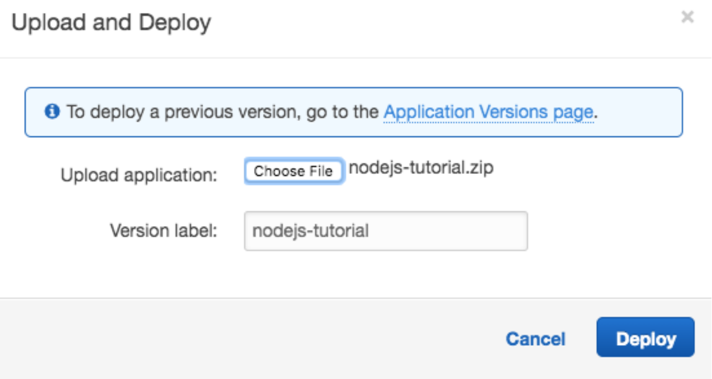
Deploy the nodejs-tutorial.zip Source Bundle to Your Environment

1. Open the AWS Elastic Beanstalk console at https://console.aws.amazon.com/elasticbeanstalk
2. Click on the environment (Labapp-env) created for lab-app.
3. Click **Upload and Deploy.**



Labapp-env-1
Labapp-env-1.eba-5g2vjhfm.us-east-2.elasticbeanstalk.com (e-fa9tmmyigv)
Application name: lab-app

4. Click **Choose File.**



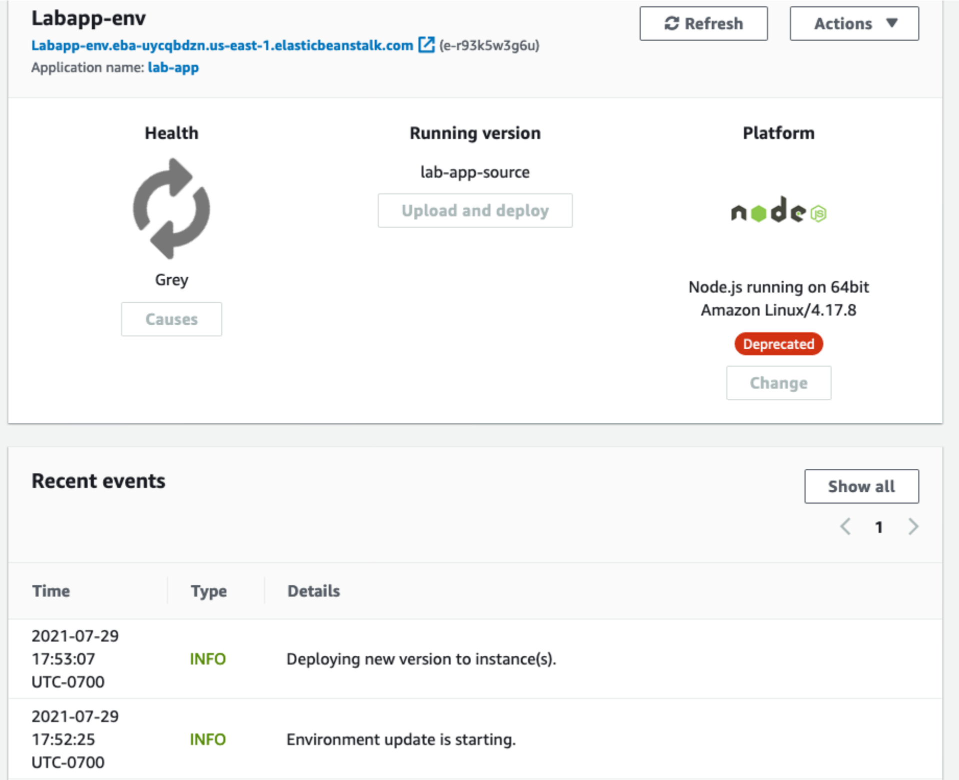
Upload and Deploy

To deploy a previous version, go to the Application Versions page.

Upload application: Choose File nodejs-tutorial.zip

Version label: nodejs-tutorial

5. 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.



Labapp-env

Labapp-env.eba-uycqbdzn.us-east-1.elasticbeanstalk.com (e-r93k5w3g6u)

Application name: lab-app

Health: Grey

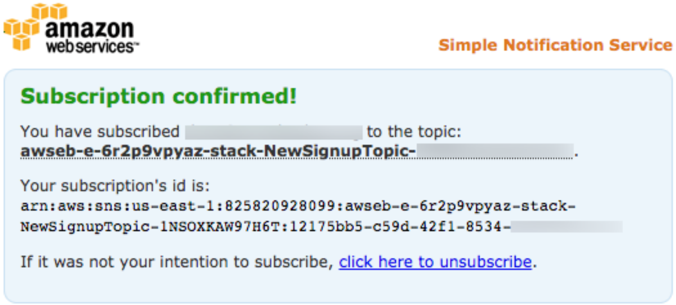
Running version: lab-app-source

Platform: Node.js running on 64bit Amazon Linux/4.17.8

Recent events

Time	Type	Details
2021-07-29 17:53:07 UTC-0700	INFO	Deploying new version to instance(s).
2021-07-29 17:52:25 UTC-0700	INFO	Environment update is starting.

6. 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.



Subscription confirmed!

You have subscribed [redacted] to the topic:
awseb-e-6r2p9vpyaz-stack-NewSignupTopic-[redacted].

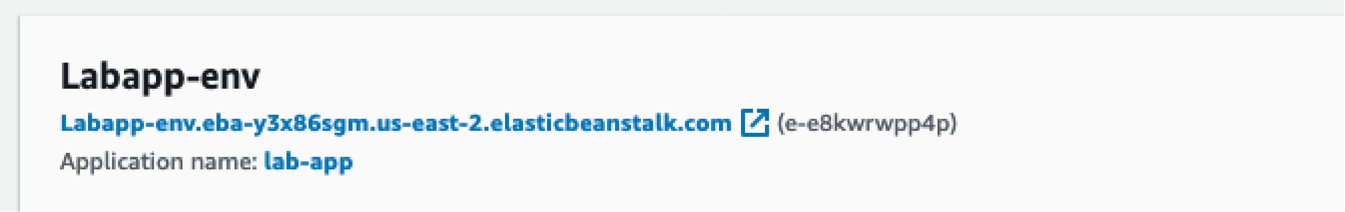
Your subscription's id is:
`arn:aws:sns:us-east-1:825820928099:awseb-e-6r2p9vpyaz-stack-NewSignupTopic-1NSOXKAW97H6T:12175bb5-c59d-42f1-8534-[redacted]`

If it was not your intention to subscribe, click here to unsubscribe.

View & Test the Application

1. Open the AWS Elastic Beanstalk console at https://console.aws.amazon.com/elasticbeanstalk
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.



Labapp-env

Labapp-env.eba-y3x86sgm.us-east-2.elasticbeanstalk.com (e-e8kwrwpp4p)

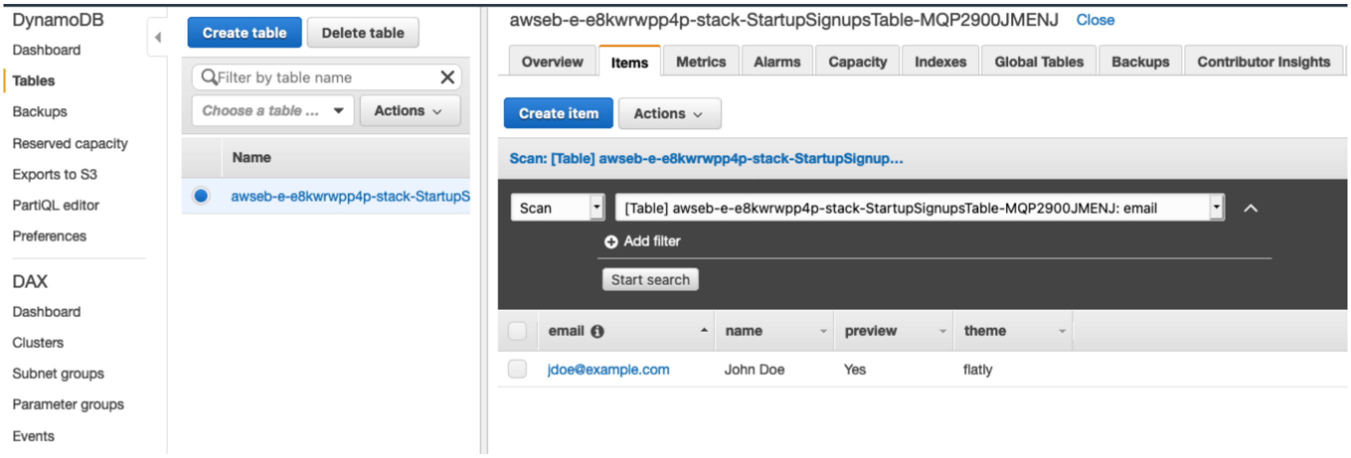
Application name: lab-app

4. 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 **(.ebextensions/create-dynamodb-table)** that create the DynamoDB table used by the application.

1. Open the Tables page 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.



awseb-e-e8kwrwpp4p-stack-StartupSignupsTable-MQP2900JMENJ

Name
● awseb-e-e8kwrwpp4p-stack-StartupS

Scan: [Table] awseb-e-e8kwrwpp4p-stack-StartupSignup...

Scan | [Table] awseb-e-e8kwrwpp4p-stack-StartupSignupsTable-MQP2900JMENJ: email

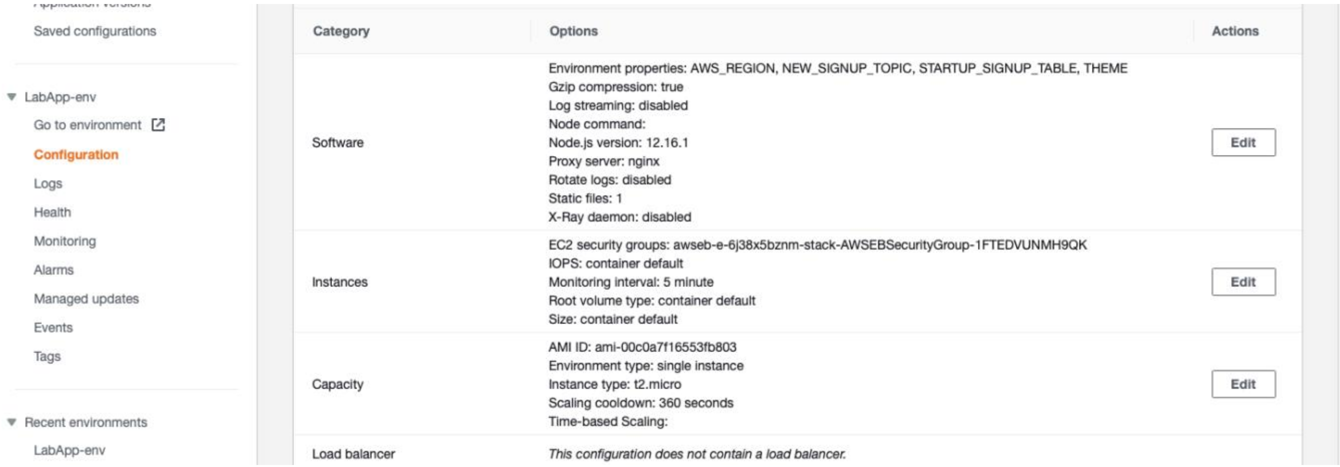
email	name	preview	theme
jdoe@example.com	John Doe	Yes	flatly

4. To get more items in the table, go back to the web application and fill out the form again. You can use a random name/email for this. 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.

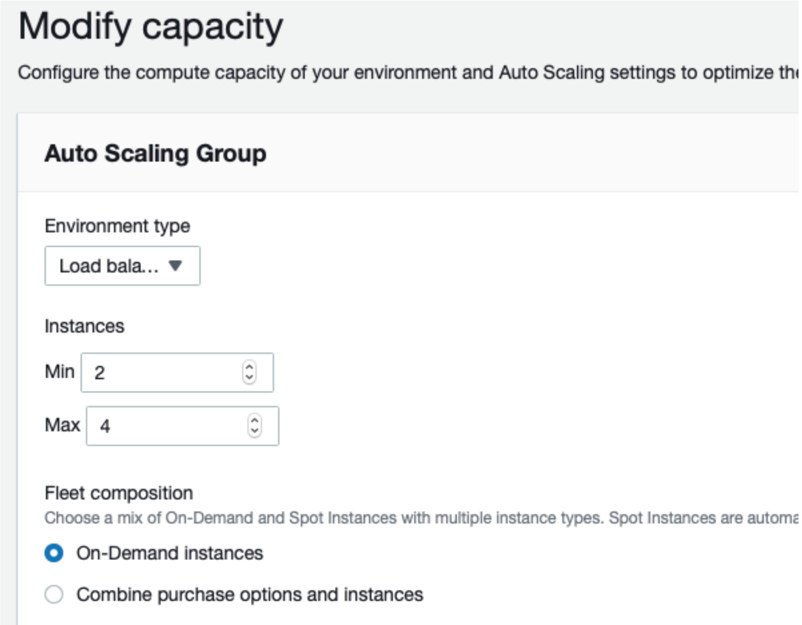
1. Open the AWS Elastic Beanstalk console at https://console.aws.amazon.com/elasticbeanstalk
2. Click on the environment (**LabApp-env**) created for lab-app.
3. Choose **Configuration** from the menu on the left.
4. In the **Capacity** section, click the **Edit** link.



Category	Options	Actions
Software	Environment properties: AWS_REGION, NEW_SIGNUP_TOPIC, STARTUP_SIGNUP_TABLE, THEME Gzip compression: true Log streaming: disabled Node command: Node.js version: 12.16.1 Proxy server: nginx Rotate logs: disabled Static files: 1 X-Ray daemon: disabled	Edit
Instances	EC2 security groups: awseb-e-6j38x5bznm-stack-AWSEBSecurityGroup-1FTEDVUNMH9QK IOPS: container default Monitoring interval: 5 minute Root volume type: container default Size: container default	Edit
Capacity	AMI ID: ami-00c0a7f16553fb803 Environment type: single instance Instance type: t2.micro Scaling cooldown: 360 seconds Time-based Scaling:	Edit
Load balancer	*This configuration does not contain a load balancer.*	

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

   Environment type – Select **Load balanced**.
   Min instances – **2**



Modify capacity

Configure the compute capacity of your environment and Auto Scaling settings to optimize th

Auto Scaling Group

Environment type: Load bala...

Instances

Min: 2

Max: 4

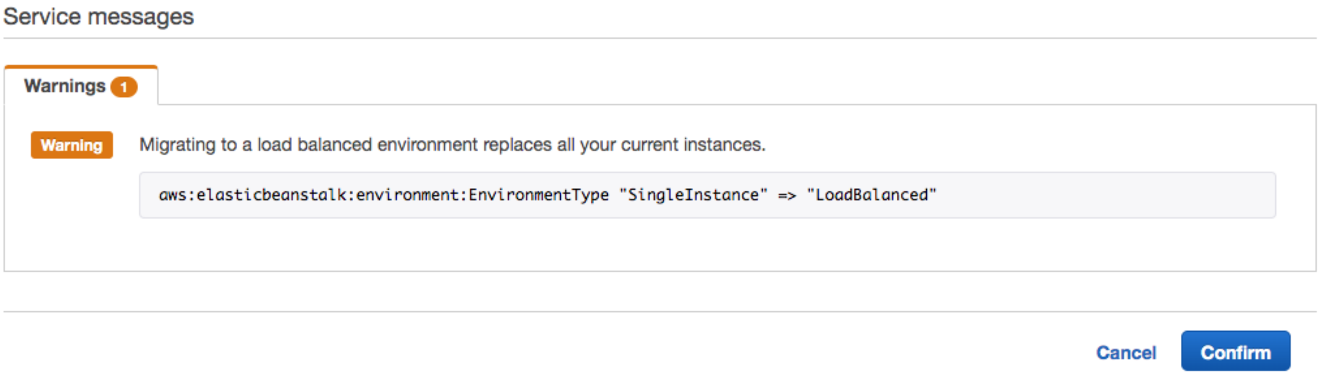
Fleet composition

Choose a mix of On-Demand and Spot Instances with multiple instance types. Spot Instances are automa

● On-Demand instances

○ Combine purchase options and instances

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

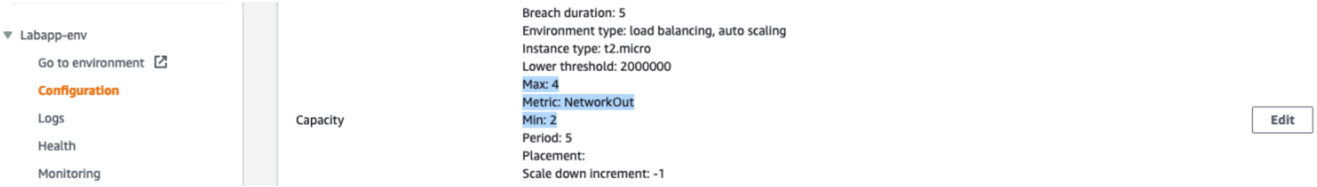


Service messages

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

```
aws:elasticbeanstalk:environment:EnvironmentType "SingleInstance" => "LoadBalanced"
```

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

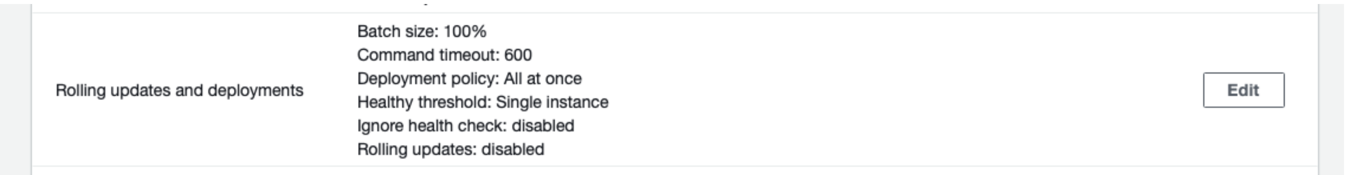


Capacity

Breach duration: 5
Environment type: load balancing, auto scaling
Instance type: t2.micro
Lower threshold: 2000000
Max: 4
Metric: NetworkOut
Min: 2
Period: 5
Placement:
Scale down increment: -1

Enable rolling deployments

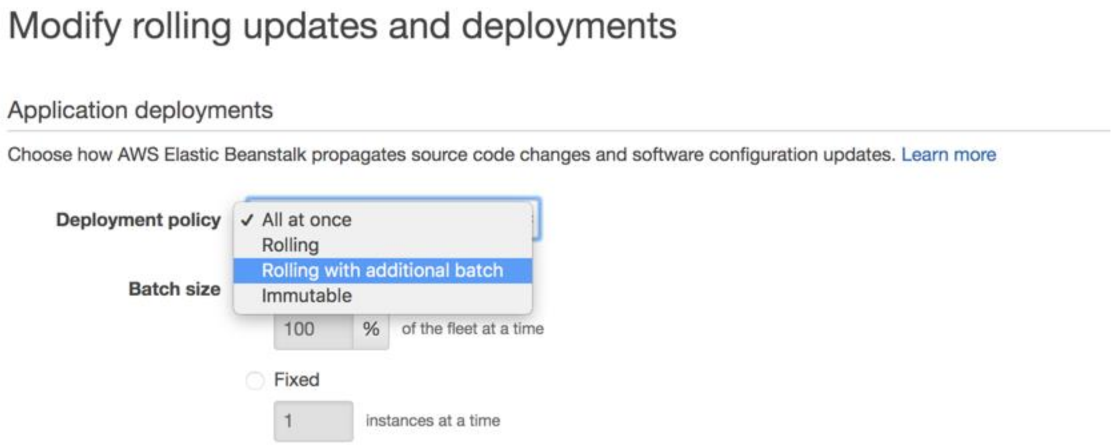
1. While on the Configuration page, click **Edit** in the **Rolling updates and deployments** section.



Rolling updates and deployments

Batch size: 100%
Command timeout: 600
Deployment policy: All at once
Healthy threshold: Single instance
Ignore health check: disabled
Rolling updates: disabled

2. At top of page, under **Application Deployments > Deployment policy**, select **Rolling with additional batch**.
3. Scroll to the bottom 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

Deployment policy: ✓ All at once / Rolling / Rolling with additional batch / Immutable

Batch size: 100 % of the fleet at a time

○ Fixed: 1 instances at a time

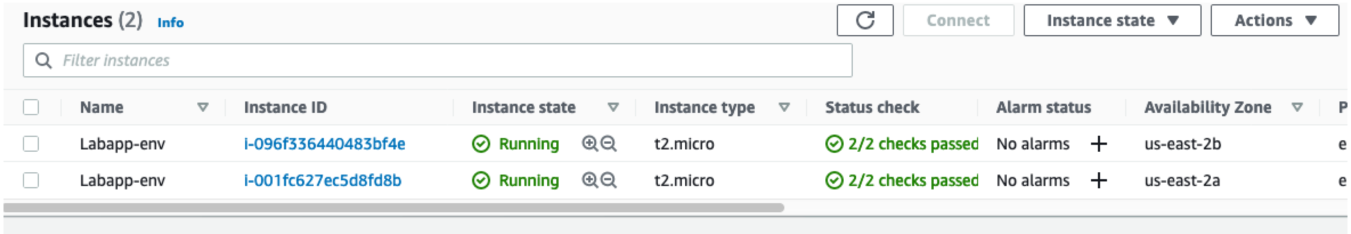
4. In the next section, you will perform a deployment to test the rolling update with batch configuration.

Perform another deployment

1. Open the AWS EC2 Console at: https://console.aws.amazon.com/ec2

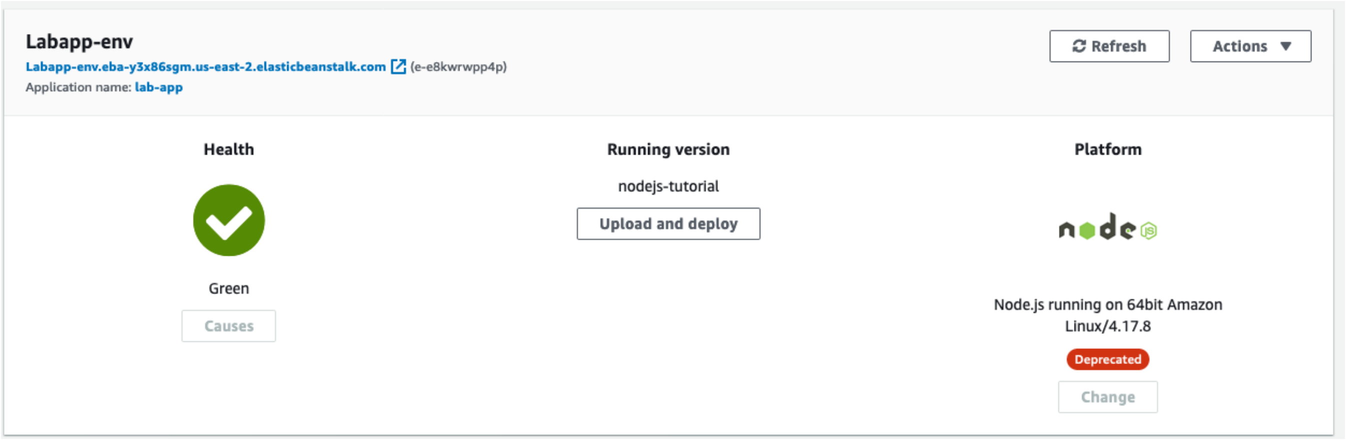
2. Click on Instances in the EC2 Dashboard to view currently running EC2 instances in your environment.

   You will see two EC2 instances that have been deployed by Elastic Beanstalk which are named (**Labapp-env**)



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	P
□	Labapp-env	i-096f336440483bf4e	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2b	e
□	Labapp-env	i-001fc627ec5d8fd8b	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2a	e

3. Keep the EC2 Dashboard open and launch a new browser to navigate to the Elastic Beanstalk dashboard: https://console.aws.amazon.com/elasticbeanstalk
4. Click on the **Labapp-env.** In the next page, click on **Upload and deploy**.



Labapp-env

Labapp-env.eba-y3x86sgm.us-east-2.elasticbeanstalk.com (e-e8kwrwpp4p)

Application name: lab-app

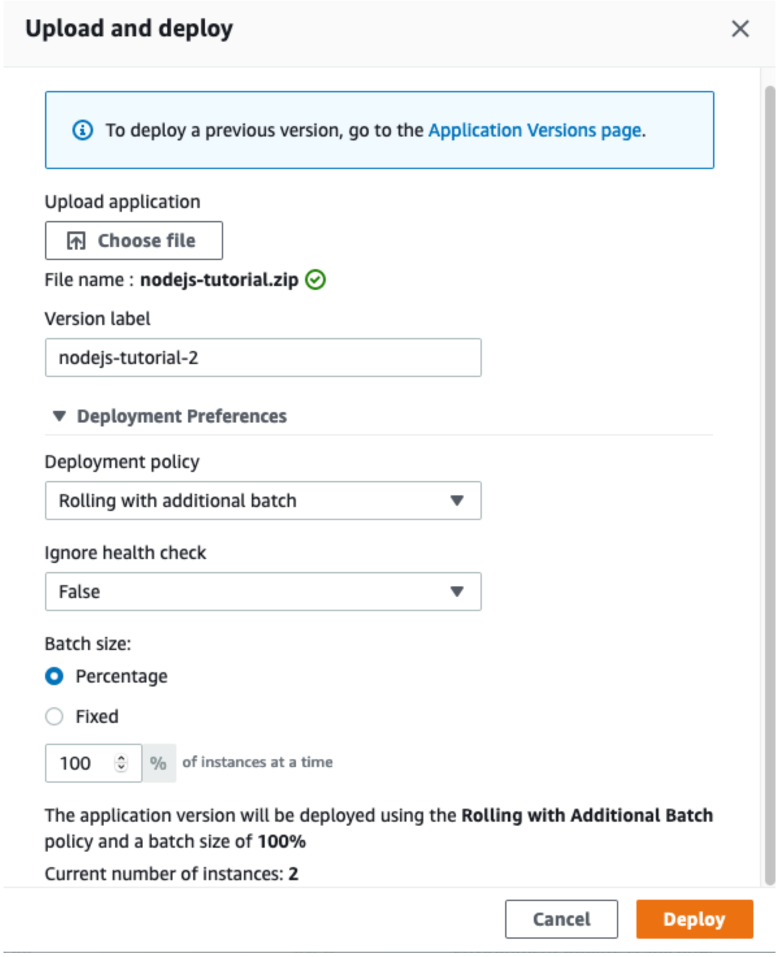
Health: Green

Running version: nodejs-tutorial

Platform: Node.js running on 64bit Amazon Linux/4.17.8 Deprecated

5. To test the rolling updates, you are going to simulate a new deployment by redeploying the Source Bundle that was created earlier in the lab.

   Expand the Deployment Preferences details. The default deployment policy selected will show **Rolling with additional batch**, which you have configured in the previous step.



To deploy a previous version, go to the Application Versions page.

Upload application

File name : **nodejs-tutorial.zip** ✓

Version label: nodejs-tutorial-2

▼ Deployment Preferences

Deployment policy: Rolling with additional batch

Ignore health check: False

Batch size:

● Percentage

○ Fixed

100 % of instances at a time

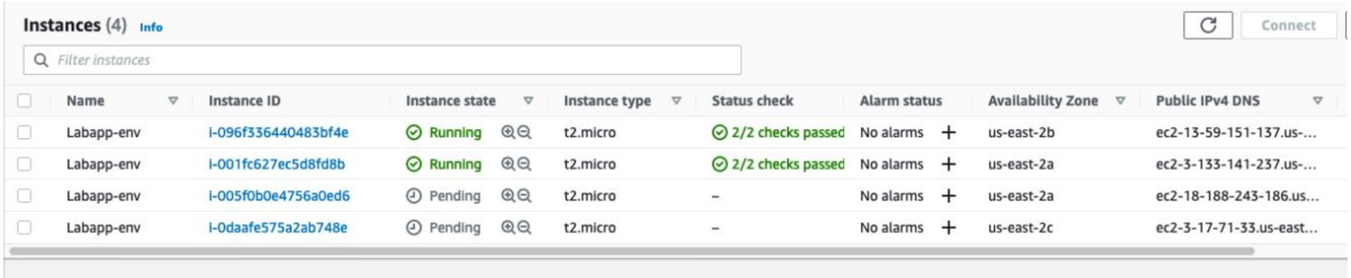
The application version will be deployed using the **Rolling with Additional Batch** policy and a batch size of **100%**

Current number of instances: **2**

6. **Deploy** your application
7. You can observe and validate the Deployment Policy behavior by navigating back to the EC2 instances page.

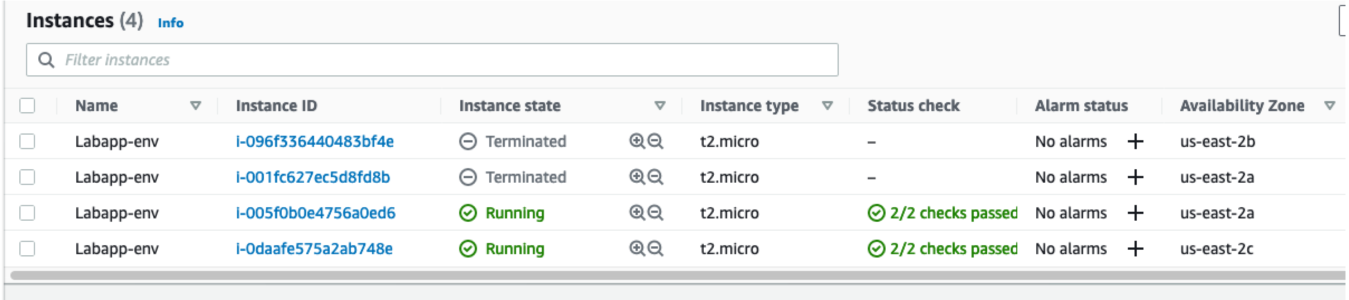
   During this deployment, your application will remain online and continue accepting new registrations.

   Two new EC2 instances will launch to support the Rolling deployment. The two new instances will take a few minutes to initialize and run.



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Labapp-env	i-096f336440483bf4e	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2b	ec2-13-59-151-137.us-...
□	Labapp-env	i-001fc627ec5d8fd8b	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2a	ec2-3-133-141-237.us-...
□	Labapp-env	i-005f0b0e4756a0ed6	Pending	t2.micro	–	No alarms +	us-east-2a	ec2-18-188-243-186.us...
□	Labapp-env	i-0daafe575a2ab748e	Pending	t2.micro	–	No alarms +	us-east-2c	ec2-3-17-71-33.us-east...

Once the two new EC2 instances are running with the new Application version deployed and passed health checks, Elastic Beanstalk will terminate the initial instances.



Instances (4) Info

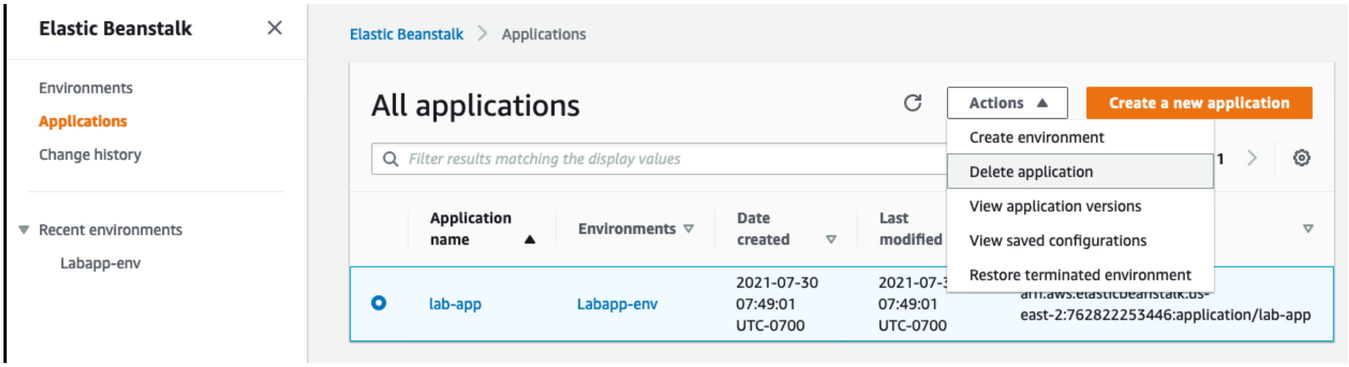
	Name	Instance ID	Instance state		Instance type	Status check	Alarm status		Availability Zone
□	Labapp-env	i-096f336440483bf4e	⊖ Terminated	🔍	t2.micro	–	No alarms	+	us-east-2b
□	Labapp-env	i-001fc627ec5d8fd8b	⊖ Terminated	🔍	t2.micro	–	No alarms	+	us-east-2a
□	Labapp-env	i-005f0b0e4756a0ed6	✓ Running	🔍	t2.micro	✓ 2/2 checks passed	No alarms	+	us-east-2a
□	Labapp-env	i-0daafe575a2ab748e	✓ Running	🔍	t2.micro	✓ 2/2 checks passed	No alarms	+	us-east-2c

Clean Up

To terminate your Elastic Beanstalk environment:

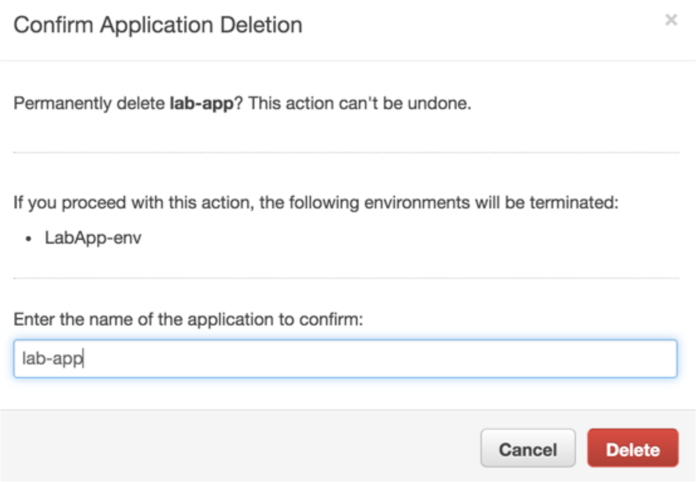
When you finish working with Elastic Beanstalk, you can terminate your environment. Elastic Beanstalk terminates all AWS resources associated with your environment, such as Amazon EC2 instances, database instances, load balancers, security groups, and alarms.

1. Open the AWS Elastic Beanstalk console at https://console.aws.amazon.com/elasticbeanstalk
2. Navigate to the **Applications** page and select the **lab-app**
3. Choose **Actions**, and then choose **Delete application**.



	Application name	Environments	Date created	Last modified	
●	lab-app	Labapp-env	2021-07-30 07:49:01 UTC-0700	2021-07-3 07:49:01 UTC-0700	arn:aws:elasticbeanstalk:us-east-2:762822253446:application/lab-app

4. In the **Confirm Application Deletion** dialog, enter the name of the application in the application field and click **Delete**.



Confirm Application Deletion

Permanently delete **lab-app**? This action can't be undone.

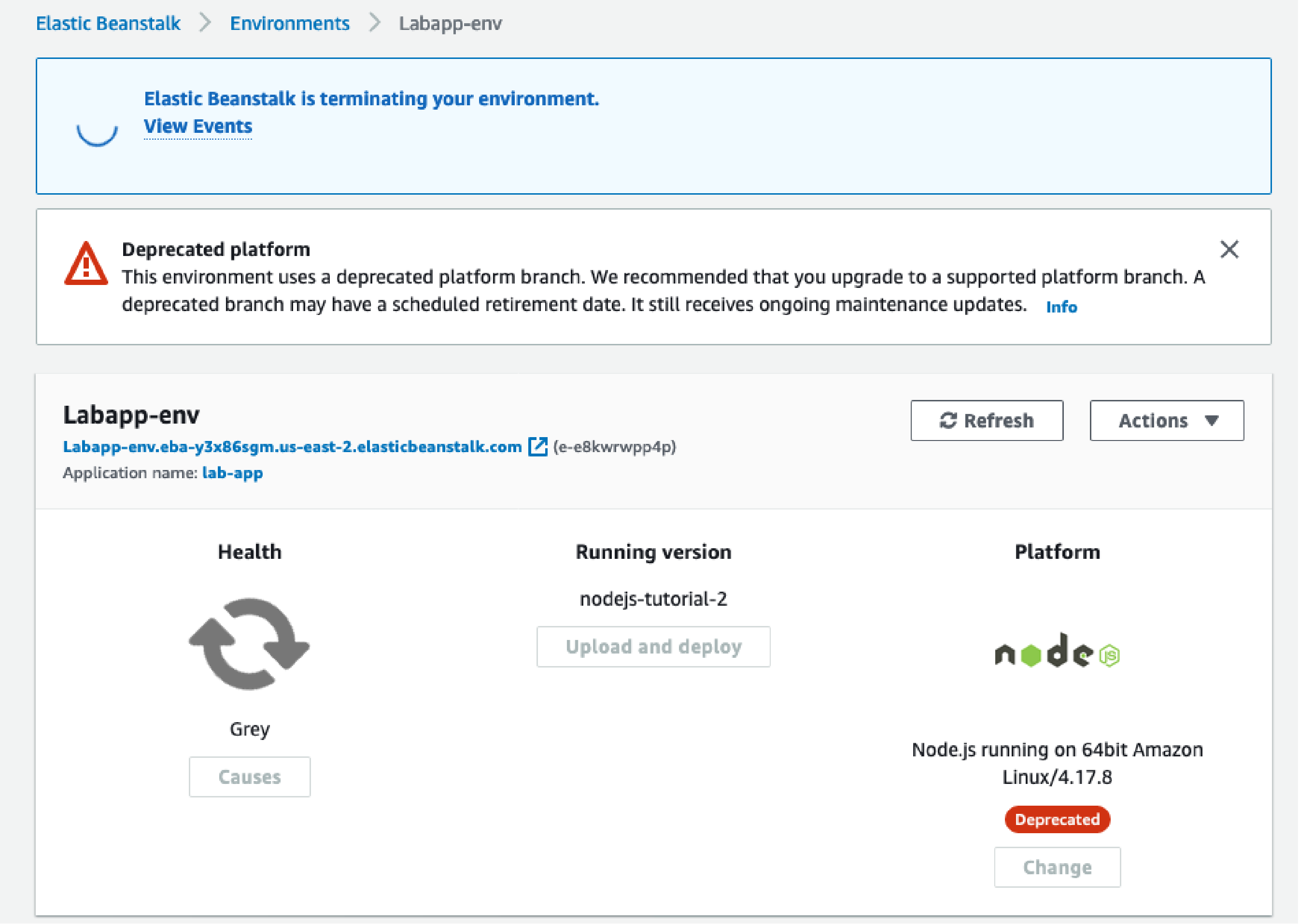
If you proceed with this action, the following environments will be terminated:

- LabApp-env

Enter the name of the application to confirm:

lab-app

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



Elastic Beanstalk > Environments > Labapp-env

Elastic Beanstalk is terminating your environment.
View Events

Deprecated platform
This environment uses a deprecated platform branch. We recommended that you upgrade to a supported platform branch. A deprecated branch may have a scheduled retirement date. It still receives ongoing maintenance updates. Info

Labapp-env
Labapp-env.eba-y3x86sgm.us-east-2.elasticbeanstalk.com (e-e8kwrwpp4p)
Application name: lab-app

Health: Grey

Running version: nodejs-tutorial-2

Platform: Node.js running on 64bit Amazon Linux/4.17.8 Deprecated

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.