

This Lab covers:

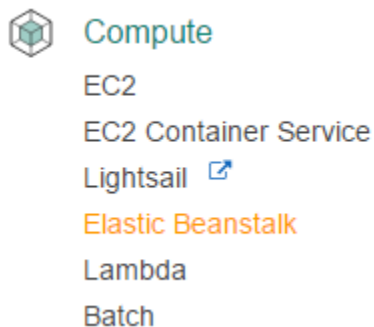
- How to build an Elastic Beanstalk Application container and environments
- The efficient way to perform a rolling deployment on underlying EC2 resources
- Blue-Green deployment with Elastic Beanstalk for zero-downtime launches
- How to mitigate cost by validating cleanup after immutable or blue-green deploys

During this lab, you will walk deployment of a new load balanced, auto scaling Todo application into an AWS VPC, upgrade the code of your app to version 2 using low-friction and rapid rolling deployment, construct a secondary deployment stack, upgrade the secondary for use as the primary, swap the environments with no downtime, and clean up the older versions of code to save money on resources.

## Step 1. Create an Elastic Beanstalk App & Environment

Since we will be performing both a rolling and blue-green deploy in this lab, it makes sense to use the best tool AWS makes available for the job. For controlled deployments and efficient deployment services of code on EC2 instances, Elastic Beanstalks provides a superior interaction model and developer tools experience.

First, we navigate to the AWS Elastic Beanstalk Console, by clicking on the **Services** tab in the top-right of the Console, hovering over the **Compute** Section in the tray, then clicking on the **Elastic Beanstalk** section...



Once inside the Elastic Beanstalk Console, because the account the Lab uses has never used Elastic Beanstalk before, you will be presented with a welcome message. The most obvious button in the center of the screen is actually **NOT** the one we want. That one launches a simplified demo application immediately. Click on the **Create Application** link in the top-right of the Console view.

Elastic Beanstalk
 [Create New Application](#)

## Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

To deploy your **existing web application**, create an [application source bundle](#) and then [create a new application](#). If you're using [Git](#) and would prefer to use it with our command line tool, please see [Getting Started with the EB CLI](#).

To deploy a **sample application** with just one click, select a platform and click **Launch Now**.

By launching the sample application, you allow AWS Elastic Beanstalk to administer AWS resources and necessary permissions on your behalf. [Learn more](#).

[Looking for a different platform? Let us know.](#)

You will be immediately presented with options. The first of which are two text blanks for **Application Name** and **Description**. These are vanity names, and do not really affect anything, so you can pick whatever you like that AWS accepts. Spaces are allowed in both the name and description. Click the blue **Next** button in the bottom right of the view.

Elastic Beanstalk
 [Create New Application](#)

**Application Info**  
 New Environment

### Application Information

To create a new application, enter the details of your application.

Application name: 
Must be less than 100 characters and cannot contain a /

Description: 
Optional.

Next, you are presented with some options for the Environment Type you want to launch. We will be launching a **Web Server Environment**, so click on the large blue button that says **Create Web Server**.

Elastic Beanstalk Create New Environment

Application Info  
New Environment

## New Environment

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

Provides resources for an AWS Elastic Beanstalk web server in either a single instance or load-balancing, auto scaling environment. [Learn more.](#)

Create web server

### Worker Environment\*

Provides resources for an AWS Elastic Beanstalk worker application in either a single instance or load-balancing, auto scaling environment. [Learn more.](#)

Create worker

\* Worker environments require additional permissions to access other AWS services. [Learn more.](#)

Cancel

Done

Then, you will be presented with two drop-downs, the top of which allows you to choose the platform or programming language in which your application will be authored. The code this Lab provides you is all **node.js**, so in the top dropdown, you should select this value. The second dropdown allows you to select either **Load Balancing**, **Auto Scaling** or **Single Instance** for the **Environment Type**. Fortunately for us, once you select **node.js** in the top dropdown, **Load Balancing**, **Auto Scaling** will be automatically selected - this is the value we want. Once you have configured these values and confirmed they match the image below, hit the blue **Next** button in the bottom right of the view.

Elastic Beanstalk Create New Environment

Application Info  
New Environment  
Environment Type

## Environment Type

Choose the platform and type of environment to launch.

Predefined configuration: Node.js Looking for a different platform? [Let us know.](#)

AWS Elastic Beanstalk will create an environment running Node.js on 64bit Amazon Linux 2015.09 v2.0.8. [Change platform version.](#)

Environment type: Load balancing, auto scaling [Learn more](#)

Cancel

Previous

Next

In the next screen, you are presented with a number of options relating to code deployment. The only one we will configure is the **Source** field. Please download [this zip file](#), which contains the

application we will use, then select it for upload using the **Choose File** button associated with the **Upload Your Own** option in the **Source** field under the **Application Version** heading.

We will ignore the **Deployment Preferences** for now - these controls do not matter very much for the lab. These controls configure how quickly and in what size batches the later **Rolling Updates** we'll configure will execute.

Once you have uploaded the **Source** bundle, please click the **Next** button in the bottom-right hand corner of the view to continue.

The screenshot shows the AWS Elastic Beanstalk console interface. The top navigation bar includes 'Services', 'Resource Groups', and user information. The left sidebar lists navigation options: 'Application Info', 'New Environment', 'Environment Type', 'Application Version' (highlighted), 'Environment Info', 'Additional Resources', 'Configuration Details', 'Environment Tags', 'Permissions', and 'Review Information'. The main content area is titled 'Application Version' and contains two sections: 'Application Version' and 'Deployment Preferences'. In the 'Application Version' section, there is a heading 'Select a source for your application version.' and three radio button options: 'Sample application', 'Upload your own (Learn more)' (which is selected), and 'S3 URL'. The 'Upload your own' option has a file input field containing 'Sfogla... v1.zip'. The 'S3 URL' option has a text input field with a placeholder '(e.g. https://s3.amazonaws.com/s3Bucket/s3key)'. The 'Deployment Preferences' section includes a note: 'Elastic Beanstalk will update your application in batches so as to avoid downtime when deploying.' It contains several settings: 'Deployment policy' set to 'Rolling' with a 'Learn more' link; 'Healthy threshold' set to 'Ok'; 'Ignore health check' set to 'False'; and 'Batch size' set to 'Percentage' with a value of '30' and a unit of '% of the fleet at a time'. There is also an option for 'Fixed' batch size set to '1' instance at a time. At the bottom right of the form, there are three buttons: 'Cancel', 'Previous', and 'Next'.

You should now be on a view with the heading **Environment Information**. This setting will play a key role later. Note that AWS attempts to auto-populate your **Environment Name** and **Environment URL** fields for you. This will sometimes not work, because the URLs operate on a subdomain basis - so all users on all AWS regions share from the same pool of URL prefix names. Pick something memorable, as you will be using this value later. Use the **Check Availability** button to make sure the name you settle on is available for your use, then when you are ready, click on the blue **Next** button in the bottom-right of the view.

The screenshot shows the 'Environment Information' step in the Elastic Beanstalk console. The left sidebar contains a list of steps: Application Info, New Environment, Environment Type, Application Version, Environment Info (highlighted), Additional Resources, Configuration Details, Environment Tags, Permissions, and Review Information. The main content area is titled 'Environment Information' and includes the instruction 'Enter your environment information.' Below this, there are three input fields: 'Environment name' with the value 'reallyCoolApp-env', 'Environment URL' with the value 'mightHaveToChange.us-west-2.elasticbeanstalk.com', and 'Description' with the value 'This is the first environment for our app.' A 'Check availability' button is located below the Environment URL field. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Check the box for **Create this environment inside a VPC**, as this Lab generated a VPC for your use, and we need to learn the extra step associated with configuring Elastic Beanstalk for use in a VPC. Do not try to **Create an RDS DB Instance with this environment**, simply click on the blue **Next** button in the bottom-right of the view.

The screenshot shows the 'Additional Resources' step in the Elastic Beanstalk console. The left sidebar contains a list of steps: Application Info, New Environment, Environment Type, Application Version, Environment Info, Additional Resources (highlighted), Configuration Details, Environment Tags, and VPC Configuration. The main content area is titled 'Additional Resources' and includes the instruction 'Select additional resources for this environment.' Below this, there are two checkboxes: 'Create an RDS DB Instance with this environment' (unchecked) and 'Create this environment inside a VPC' (checked). Both checkboxes have a 'Learn more' link next to them. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

You will land on a large configuration page. The only value you should set for this lab is to changing the **Application Health Check URL** value from an empty text field to a single slash, to represent the root of the Elastic Beanstalk domain's structure (/). After making sure to set this value to /, scroll down to the bottom and hit the blue **Next** button in the bottom right of the view.

Elastic Beanstalk

Create New Environment

Application Info

New Environment

Environment Type

Application Version

Environment Info

Additional Resources

**Configuration Details**

Environment Tags

VPC Configuration

Permissions

Review Information

### Configuration Details

Modify the following settings or click Next to accept the default configuration. [Learn more.](#)

Instance type:

t1.micro

Determines the processing power of the servers in your environment.

EC2 key pair:

Select a key pair

[Refresh](#)

Optional: Enables remote login to your instances.

Email address:

Optional: Get notified about any major changes to your environment.

Application health check URL:

/

Enter the relative URL that ELB continually monitors to ensure your application is available.

Enable rolling updates:

☒ Lets you control how changes to the environment's instances are propagated. [Learn more.](#)

Health

Specifies whether to wait to deploy updates and deployments according to a set period of time or instance health.

Cross zone load balancing:

☒ Enables load balancing across multiple Availability Zones. [Learn more.](#)

Connection draining:

☒

Enables the load balancer to maintain connections to an Amazon EC2 instance to complete in-progress requests while also stopping

Connection draining timeout:

20

seconds

Maximum time that the load balancer maintains connections to an Amazon EC2 instance before forcibly closing connections.

### Health Reporting

System type:

Enhanced

Determines the health reporting type.

### Root Volume (Boot Device)

Root volume type:

(Container default)

Determines the type of storage volume to attach to instances.

Root volume size:

☐ Enables you to specify the size of the root volume.

GIB

Number of gibibytes of the root volume attached to each instance. Must be between 10 and 16384 for Provisioned IOPS (SSD) and General Purpose (SSD) root volumes and between 8 and 1024 for other root volumes.

Cancel

Previous

Next

In the Environment Tags section click on the **Next** button, you should now have landed on the **VPC Configuration** view. Here you must select the proper subnet values for the EC2 and ELB portions of the Elastic Beanstalk deployment. The Lab will have auto-generated subnets with the same exact same CIDR blocks, so you should put the ELB in the Subnet with a CIDR block covering the lower range of the two, as this is the public subnet's block. When you're done evaluating your work, please proceed to the next view by clicking the blue **Next** button in the bottom right.

*Note: You might have to select a different VPC in order to match the one showed in the laboratory.*

Application Info
New Environment
Environment Type
Application Version
Environment Info
Additional Resources
Configuration Details
Environment Tags
**VPC Configuration**
Permissions
Review Information

## VPC Configuration

Select the VPC to use when creating your environment. [Learn more.](#)

VPC:  [Refresh](#)

☒ Associate Public IP Address

Select different subnets for ELB and EC2 instances in your Availability Zone.

AZ	Subnet	ELB	EC2
us-west-2a	subnet-e3aab886 (10.0.1.0/24)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	subnet-edaab888 (10.0.0.0/24)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
us-west-2b			
us-west-2c			

VPC security group:  [Refresh](#)

ELB visibility:  Select Internal when load balancing a back-end service that should not be publicly available.

[Cancel](#)
[Previous](#)
[Next](#)

The next screen covers Roles and Identity on AWS. Use the standard suggested Elastic Beanstalk roles for the both the individual instances and the service roll.

Elastic Beanstalk

Create New Environment

Application Info
New Environment
Environment Type
Application Version
Environment Info
Additional Resources
Configuration Details
Environment Tags
VPC Configuration
**Permissions**
Review Information

## Permissions

Select an instance profile and service role for your AWS Elastic Beanstalk environment.

An instance profile is an IAM role configured for use with EC2 instances. The instances in your Elastic Beanstalk use the credentials provided by the instance profile to communicate with AWS.

A service role allows the Elastic Beanstalk service to monitor environment resources on your behalf. See [Roles and Instance Profiles](#) in the Elastic Beanstalk developer guide for details.

Instance profile:

Service role:

[Cancel](#)
[Previous](#)
[Next](#)

Scroll all the way down to the bottom of the view, then hit the blue **Launch** button.

### Permissions

**Service role** aws-elasticbeanstalk-service-role

**Instance profile** aws-elasticbeanstalk-ec2-role

[Cancel](#)
[Previous](#)
[Launch](#)

After hitting **Launch**, you should see something like the image below, substituting your own names for a couple of the fields...

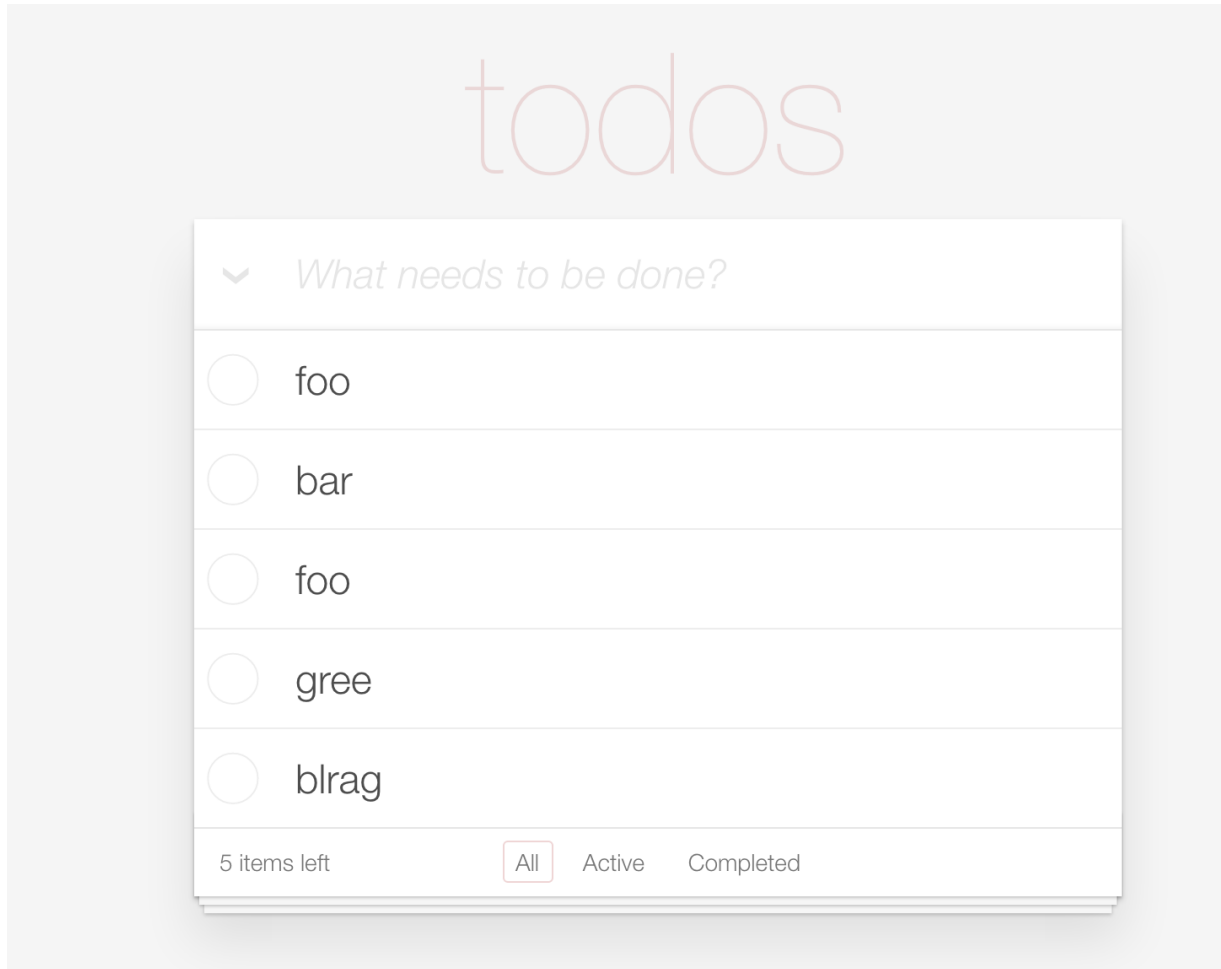
The screenshot shows the Elastic Beanstalk console interface. At the top, there's a header with the Elastic Beanstalk logo and a 'Create New Environment' button. Below the header, an information box states: 'Elastic Beanstalk is now creating your environment. When it has finished it will be running First Release.' The main content area is titled 'Really Cool App > reallyCoolApp-env' with environment details: 'Environment ID: e-ptnrvfxpvys, URL: mightHaveToChange.us-west-2.elasticbeanstalk.com'. A left sidebar contains navigation links: Dashboard, Configuration, Logs, Health (marked as 'NEW'), Monitoring, Alarms, Events, and Tags. The 'Overview' tab is selected, displaying a large blue box with a refresh icon and the text 'Elastic Beanstalk is launching your environment. View Events'. Below this, the 'Health' section shows a circular arrow icon, 'Health Pending', and a 'Causes' button. The 'Running Version' section shows an 'Upload and Deploy' button. The 'Configuration' section shows '64bit Amazon Linux 2015.09 v2.0.8 running Node.js' with a 'Change' button. A 'Refresh' button is located in the top right of the Overview section.

... After waiting for a while, your system will transition into code green, like below.

This screenshot shows the same Elastic Beanstalk console interface as the previous one, but the environment 'reallyCoolApp-env' is now in a 'Healthy' state. The 'Health' section now displays a large green checkmark icon, 'Health Ok', and a 'Causes' button. The 'Running Version' section shows 'v1' and an 'Upload and Deploy' button. The 'Configuration' section remains the same, showing '64bit Amazon Linux 2015.09 v2.0.8 running Node.js' with a 'Change' button. The 'Overview' tab is still selected, and the 'Refresh' button is present in the top right.




Once your system has signaled it's done launching, you should be able to enjoy the app itself. Go check it out!



The application is a simple todo app. This Lab uses the Angular.js [Todomvc.com](http://todomvc.com) mini app.

## Step 2. Run a Rolling Deploy

Once you have had a bit of time to play with the new environment we created, return to the status screen.

 Elastic Beanstalk

Create New Environment

Really Cool App ▸ reallyCoolApp-env ( Environment ID: e-ptnrvfxpvys, URL: [mightHaveToChange.us-west-2.elasticbeanstalk.com](http://mightHaveToChange.us-west-2.elasticbeanstalk.com) )

Actions ▾

Dashboard

Configuration

Logs

Health NEW


Monitoring

Alarms

Events

Tags

Overview



Health


Ok

Causes

Running Version

v1

Upload and Deploy



Configuration

64bit Amazon Linux 2015.09  
v2.0.8 running Node.js

Change

Refresh

When the dialog opens, select [this version two code](#) and click on the **Upload and Deploy** button in the bottom-right corner of the screen.

Upload and Deploy

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

Upload application:

Choose File

v2.zip

Version label:

v2

Deployment Preferences

Elastic Beanstalk will deploy to **30%** of instances in your auto scaling group at a time. Current number of instances: **1**

Batch size:

☒ Percentage

30

%

 of instances at a time

☐ Fixed

1

 instances at a time (max: 4)

Ignore health check:

false

Cancel

Deploy

After hitting deploy, Elastic Beanstalk will begin working on a Mutable Rolling Deploy of your code bundle. The environment will turn Gray, and will have a number of messages which amount to "please wait".

Elastic Beanstalk

Create New Environment

Really Cool App ▶ reallyCoolApp-env (Environment ID: e-ptnrvfxpvy5, URL: [mightHaveToChange.us-west-2.elasticbeanstalk.com](http://mightHaveToChange.us-west-2.elasticbeanstalk.com))

Actions ▾

Dashboard

Configuration

Logs

Health NEW

Monitoring

Alarms

Events

Tags

Elastic Beanstalk is updating your environment.

To cancel this operation select **Abort Current Operation** from the **Actions** dropdown.

[View Events](#)

Overview

Refresh

Health

Ok

Causes

Running Version

v1

Upload and Deploy

nodejs

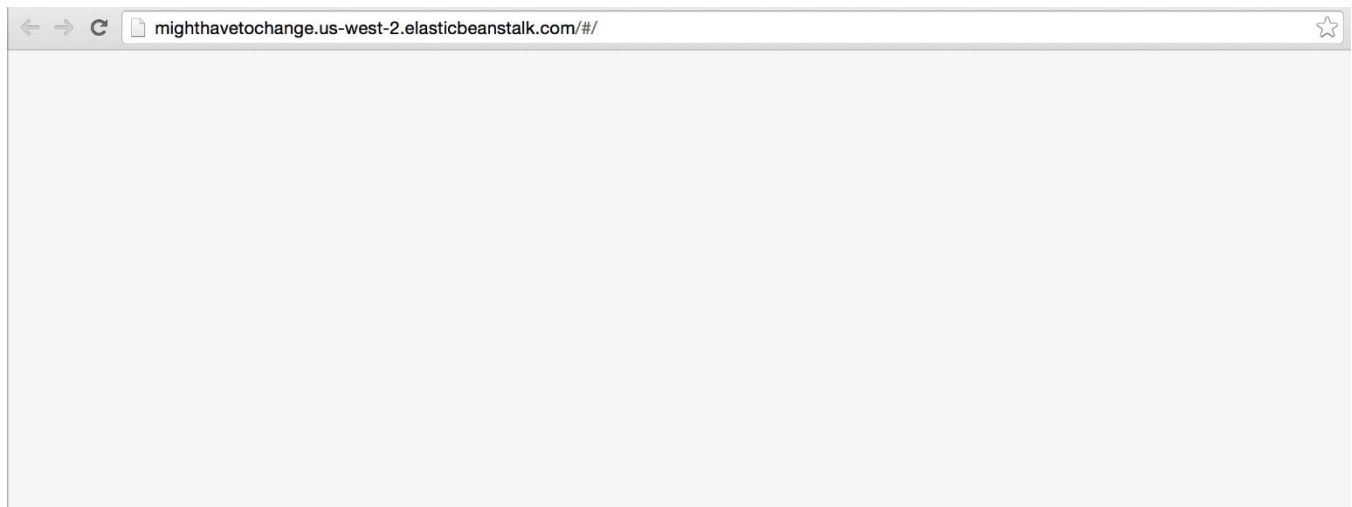
Configuration

64bit Amazon Linux 2015.09

v2.0.8 running Node.js

Change

While you are waiting, think about how a rolling deploy works. If you watched the associated course for Advanced Deployment Techniques on AWS, you should know that a Mutable Rolling Deploy will update the environment by briefly shutting down portions of it during deploy. If you go back to the "todo" front-end, you may be able to refresh during this Rolling Deploy and find that the application is unavailable.



While the Rolling Deploy in Elastic Beanstalk is very simple and the most convenient one, it may not fit your use case.

After a while, your environment should stabilize. The gray spinner on the left of center will stop spinning and turn back into a green check mark.

Elastic Beanstalk

Create New Environment

Really Cool App ▸ reallyCoolApp-env ( Environment ID: e-ptnrvfxpvys, URL: mightHaveToChange.us-west-2.elasticbeanstalk.com )

Actions ▾

Dashboard

Configuration

Logs

Health NEW

Monitoring

Alarms

Events

Tags

Overview

Health

Ok

Causes

Running Version

v2

Upload and Deploy

node

Configuration

64bit Amazon Linux 2015.09  
v2.0.8 running Node.js

Change

Recent Events

Show All

Your environment should report that it is running V2 now. The only change we made in V2 was altering some text in the todo view from **todos** to **Cool V2**. Are you able to see the updated test when you re-visit the application environment URL?

# Cool V2

▼

What needs to be done?

☐

foo

☐

bar

☐

foo

☐

gree

☐

blrag

5 items left

All

Active

Completed

Once you confirm that you can see the changes, proceed to the **next step** in this lab.

## Step 3. Prepare a Blue-Green Deploy

Now that we have run a Rolling Deploy, it's time to try our hand at the other major kind of deploy, the Blue-Green deploy.

If you recall, the first thing we need with this kind of deploy is a complete second environment that mirrors the one currently being used.

Fortunately, Elastic Beanstalk has a **Clone Environment** feature. From within the environment overview area, click on the **Actions** dropdown in the top right, then click on **Clone Environment**.


Elastic Beanstalk Create New Environment

Really Cool App ▸ reallyCoolApp-env (Environment ID: e-ptnrvfxpvy, URL: mightHaveToChange.us-west-2.elasticbeanstalk.com)

Dashboard Overview

Configuration

Logs

Health NEW  **Health** Ok Causes

Monitoring

Alarms

Events

Tags

Running Version v2 Upload and Deploy

64bit Amazon Linux 2015.09 v2.0.8 running Node.js

Change

Recent Events Show All

**Actions**

- Load Configuration
- Save Configuration
- Swap Environment URLs
- Clone Environment**
- Clone with Latest Platform
- Abort Current Operation
- Restart App Server(s)
- Rebuild Environment
- Terminate Environment

You will be presented with a form and a view that describes the existing environment at a high level, then also shows the settings of the to-be-created environment. We are going to use this as the new environment to try blue-green on, so pick a name and Environment URL which is self-explanatory.

#### Original Environment

Environment name: reallyCoolApp-env

Environment URL: mightHaveToChange.us-west-2.elasticbeanstalk.com

Description: This is the first environment for our app.

Platform: 64bit Amazon Linux 2015.09 v2.0.8 running Node.js

#### New Environment

Environment name:

Environment URL:  Check availability

Description:

Platform:  Most recent version

Service role:  Refresh [Learn more.](#)

Cancel Clone

Once you click the blue **Clone** button, the same process and view that we experienced earlier during environment creation appears. Wait it out again.

The image displays two screenshots of the AWS Elastic Beanstalk console, illustrating the process of creating and deploying an environment.

**Top Screenshot:** The console shows the 'Really Cool App' environment 'reallyCoolApp-round-2' (Environment ID: e-q3p9qrwmwm, URL: reallycoolapp-round-2.us-west-2.elasticbeanstalk.com). The environment is in the 'Pending' state. A blue banner at the top states: 'Elastic Beanstalk is now creating your environment. When it has finished it will be running v2.' The 'Overview' section shows a circular arrow icon, 'Health: Pending', and a 'Causes' button. The 'Running Version' section shows 'v2' and an 'Upload and Deploy' button. The 'Configuration' section shows '64bit Amazon Linux 2015.09 v2.0.8 running Node.js' and a 'Change' button. A 'Refresh' button is visible in the top right of the Overview section.

**Bottom Screenshot:** The console shows the same environment 'reallyCoolApp-round-2'. The environment is now in the 'Ok' state. The 'Overview' section shows a green checkmark icon, 'Health: Ok', and a 'Causes' button. The 'Running Version' section shows 'v2' and an 'Upload and Deploy' button. The 'Configuration' section shows '64bit Amazon Linux 2015.09 v2.0.8 running Node.js' and a 'Change' button. A 'Refresh' button is visible in the top right of the Overview section. Below the Overview section, there is a 'Recent Events' section with a 'Show All' button.

Once you have the second environment open, we need to prepare that environment to be the upgraded version, or version number 3. Open the code deployment dialog again, use [this version three code](#), then execute a Rolling Deployment on this environment - because it is unused right now, we can tolerate the availability and performance concerns surrounding Rolling Deployments.



Upload and Deploy

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

Upload application:

Choose File

v3.zip

Version label:

v3

### Deployment Preferences

Elastic Beanstalk will deploy to **30%** of instances in your auto scaling group at a time. Current number of instances: **0**

Batch size:

☒ Percentage

30

%

of instances at a time

☐ Fixed

1

instances at a time (max: 4)

Ignore health check:

false

Cancel

Deploy

Wait again while this deploy completes.

Elastic Beanstalk

Create New Environment

Really Cool App ▸ reallyCoolApp-round-2 (Environment ID: e-q3p9qrwmwm, URL: reallycoolapp-round-2.us-west-2.elasticbeanstalk.com) Actions

Dashboard

Configuration

Logs

Health NEW

Monitoring

Alarms

Events

Tags

Health

Ok

Causes

Running Version

v2

Upload and Deploy

nodejs

Configuration

64bit Amazon Linux 2015.09

v2.0.8 running Node.js

Change

Refresh

Recent Events Show All

Elastic Beanstalk

Create New Environment

Really Cool App ▸ reallyCoolApp-round-2 (Environment ID: e-q3p9qrwmwm, URL: reallycoolapp-round-2.us-west-2.elasticbeanstalk.com) Actions

Dashboard

Configuration

Logs

Health NEW

Monitoring

Alarms

Events

Tags

Health

Ok

Causes

Running Version

v3

Upload and Deploy

nodejs

Configuration

64bit Amazon Linux 2015.09

v2.0.8 running Node.js

Change

Refresh

Recent Events Show All

Now that the application's third version is deployed, manually verify that the code was deployed correctly to this environment by navigating to the URL associated to the environment and view the functioning Todo application. Note that this time, we changed the header text again, to **B-G Deploy**.

# B-G Deploy

▼

What needs to be done?

☐

foo

☐

bar

☐

foo

☐

gree

☐

blrag

5 items left

All

Active

Completed

Once you're satisfied that a third version of the same application has been deployed, proceed to the **next step** in this Lab.

## Step 4. Do the Deploy Swap

When you are ready to run the rest of the Blue-Green deploy, navigate back to the page with the environment dashboard.

Blue-Green deployments rely on having entire systems or subsystems deployed all at once using a DNS or reverse proxy cutover. We currently have two environments, one of which is running V3 and the other of which is running V2. The one running V2 is using the original DNS that we need to continue using, so we should quickly and invisibly trade DNS on the V2 and V3 versions, so V3 can get the original and more valuable URL.

Click on the **Actions** dropdown, then click on **Swap Environment URLs**.

Elastic Beanstalk

Create New Environment

Really Cool App ▶ reallyCoolApp-round-2 (Environment ID: e-q3p9qrwmwm, URL: reallycoolapp-round-2.us-west-2.elasticbeanstalk.com)

Dashboard

Configuration

Logs

Health NEW

Monitoring

Alarms

Events

Tags

Overview

Health

Ok

Causes

Running Version

v3

Upload and Deploy

64bit Amazon Linux 2015.09

v2.0.8 running Node.js

Change

Recent Events

Show All

Actions

Load Configuration

Save Configuration

Swap Environment URLs

Clone Environment

Clone with Latest Platform

Abort Current Operation

Restart App Server(s)

Rebuild Environment

Terminate Environment

You will be taken to a page which allows you to select the environment you would like to swap with. Pick the V2 environment.

Elastic Beanstalk

Create New Environment

Swap Environment URLs

When you swap an environment's URL with another environment's URL, you can deploy versions with no downtime. [Learn more.](#)

Warning

Swapping the environment URL will modify the Route 53 DNS configuration, which may take a few minutes. Your application will continue to run while the changes are propagated.

Environment Details

Environment name:

reallyCoolApp-round-2

Environment URL:

reallycoolapp-round-2.us-west-2.elasticbeanstalk.com

Select an Environment to Swap

Environment name:

reallyCoolApp-env (e-ptnvfxpvy ↕)

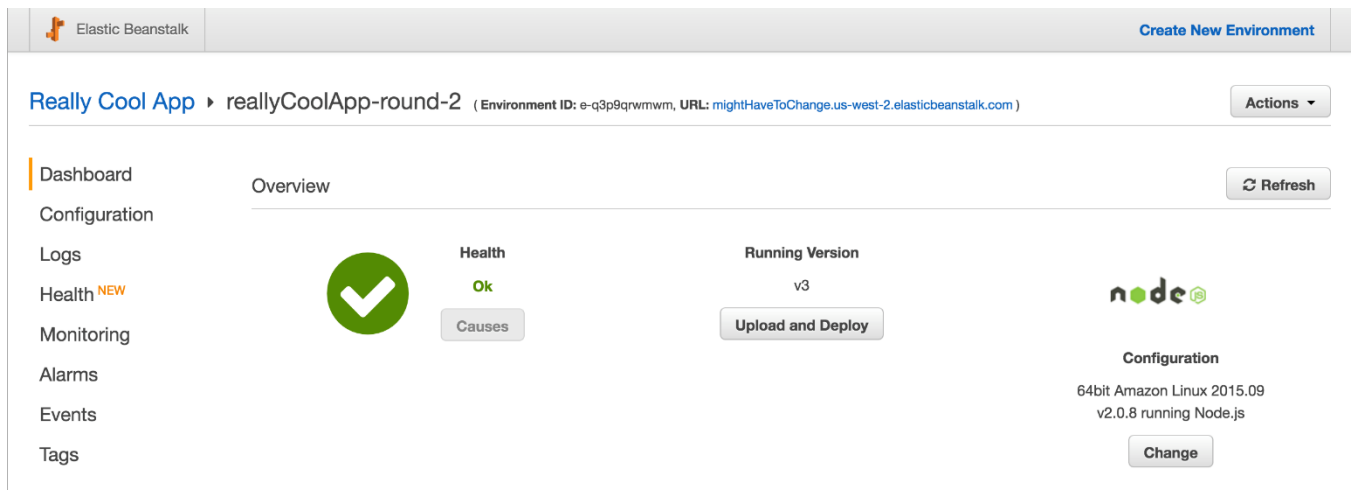
Environment URL:

mightHaveToChange.us-west-2.elasticbeanstalk.com

Cancel

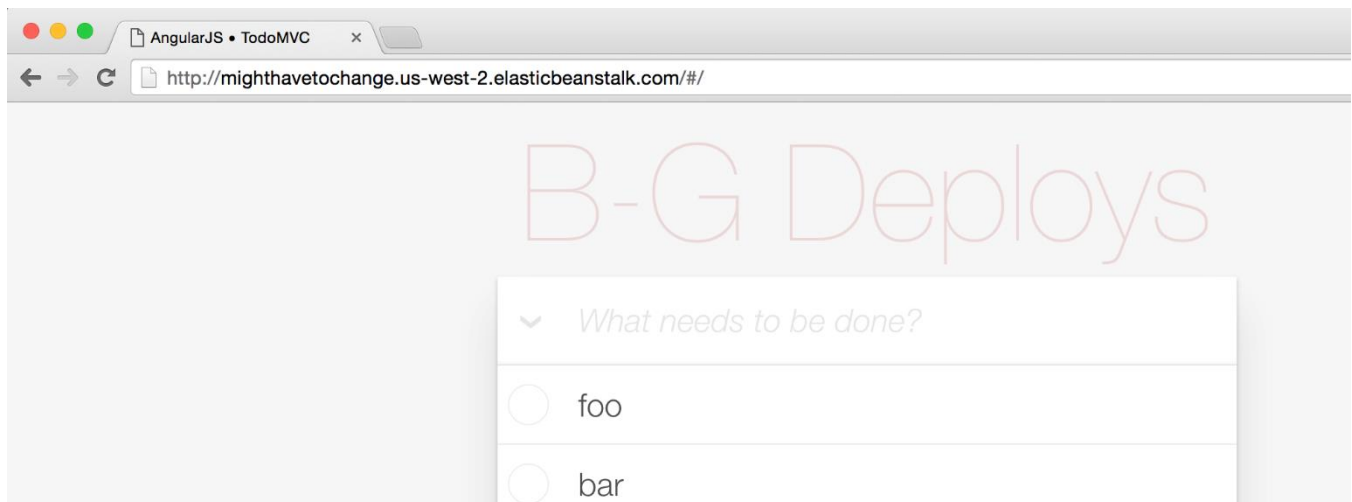
Swap

While the system warns you that the change could take a number of minutes, it's effectively instant in nearly all cases. The environment will likely be finished swapping as soon as you return to the dashboard for either environment.



This swap is simple to perform and fast. There is no downtime during the cutover since the DNS had both environments working correctly. All traffic to the original Elastic Beanstalk Environment is now routed to the new one running V3, as the V3 environment has taken over all the traffic via DNS.

Try it out - visit the original URL you made for the **first** environment, and you should see the V3 version running there now.



Once you are satisfied that the deployment worked as intended, proceed to the **next step**.

## Step 5. Clean Up Old Resources

Now that we have successfully tried two kinds of low-downtime cloud deployment techniques on Elastic Beanstalk, we might be concerned about the costs associated with these techniques. The deploys are inexpensive, but they may get a little more expensive if we simply leave the environment we swapped away from online all the time. We should clean up the environment to save costs.

First, navigate back to the dash for the environment we just swapped to - the one running V3.

The screenshot shows the Elastic Beanstalk console interface. At the top, there's a header with the Elastic Beanstalk logo and a 'Create New Environment' button. Below the header, the breadcrumb 'Really Cool App > reallyCoolApp-round-2' is displayed, along with environment details like ID and URL. A left sidebar contains navigation links: Dashboard, Configuration, Logs, Health (marked as 'NEW'), Monitoring, Alarms, Events, and Tags. The main content area is titled 'Overview' and features a 'Refresh' button. It displays a large green checkmark icon, a 'Health' status of 'Ok' with a 'Causes' button, a 'Running Version' of 'v3' with an 'Upload and Deploy' button, and a 'Configuration' section showing '64bit Amazon Linux 2015.09 v2.0.8 running Node.js' with a 'Change' button.

Click on the Elastic Beanstalk icon in the top-left of this view to navigate to the **All Applications** view. Your application will show up, along with two green boxes, representing two live environments associated with the application.

## All Applications

Filter by Appli

### Really Cool App

<u><a href="#">reallyCoolApp-env</a></u>	reallyCoolApp-round-2
<b>Environment tier:</b> Web Server <b>Running versions:</b> v2	<b>Environment tier:</b> Web Server <b>Running versions:</b> v3

Click the one running **v2**. You will be taken to the dashboard for this now-obsolete environment. Click on the **Actions** drop-down, then click on the **Terminate Environment** option, so we can destroy this environment and save some money.

Elastic Beanstalk

Create New Environment

Really Cool App ▶ reallyCoolApp-env ( Environment ID: e-ptnrvfxpvys, URL: reallycoolapp-round-2.us-west-2.elasticbeanstalk.com )

Dashboard

Configuration

Logs

Health NEW


Monitoring

Alarms

Events

Tags

Overview



Health

Ok

Causes

Running Version

v2

Upload and Deploy

64bit Amazon Linux 2015.09

v2.0.8 running Node.js

Change

Recent Events

Show All

Actions

Load Configuration

Save Configuration

Swap Environment URLs

Clone Environment

Clone with Latest Platform

Abort Current Operation

Restart App Server(s)

Rebuild Environment

Terminate Environment

You should see the environment spin into a gray state, then see the **Health** spin into a death spiral - it might swing between **OK**, **Severe**, and **No Data**. Wild!



# Health

Ok

Causes

Elastic Beanstalk

Create New Environment

Really Cool App ▶ reallyCoolApp-round-2 (Environment ID: e-q3p9qrwmwm, URL: mightHaveToChange.us-west-2.elasticbeanstalk.com) Actions ▾

Dashboard  
Configuration  
Logs  
Health **NEW**  
Monitoring  
Alarms  
Events  
Tags

Overview

Refresh

Health  
Ok  
Causes

Running Version  
v3  
Upload and Deploy

nodejs  
Configuration  
64bit Amazon Linux 2015.09  
v2.0.8 running Node.js  
Change



# Health

No data

Causes

Eventually, this will end, and you will be sent back to the **All Applications** view. This time, there will not be two green environments under your application - there will be one light-gray one, and one green one. The light-gray one is the terminated environment.



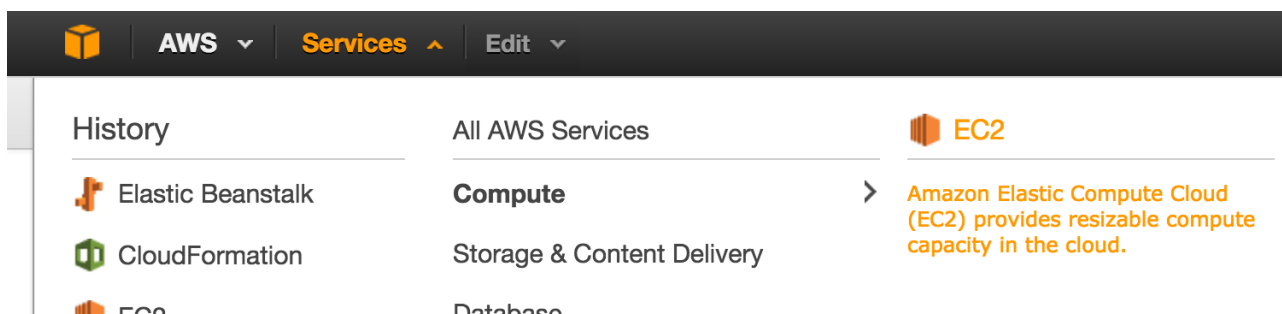
# All Applications

Filter by Applic

## Really Cool App

reallyCoolApp-env (Terminated)	reallyCoolApp-round-2
Environment tier: Web Server	Environment tier: Web Server
Running versions: v2	Running versions: v3

To verify that we reduced the spend of our system, we can enter the EC2 console and make sure that the EC2 instances that Elastic Beanstalk creates on your behalf are terminated. Navigate to the EC2 Console by clicking on the **Services** dropdown in the top-left of the Console, hovering over the **Compute** section, and clicking on the **EC2** section.



After navigating to the top level of the EC2 console, you should be able to see that there is only 1 Running Instance now if you allowed your old/second Elastic Beanstalk Environment to totally terminate.

**EC2 Dashboard**

- Events
- Tags
- Reports
- Limits

## Resources

You are using the following Amazon EC2 resources in the US

- 1 Running Instances
- 0 Dedicated Hosts

For further verification that actual instances were terminated, click the **Running Instances** link and review the table of instances. One should be **Running**, and one or more should display as red and **Terminated**. This is / these are instance(s) which belonged to the now-terminated Elastic Beanstalk Environment.

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

**Instances**

Spot Requests

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

1 to 2 of 2

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
<input type="checkbox"/>	reallyCoolAp...	i-5e4f6799	t1.micro	us-west-2a	<span>●</span> running	<span>●</span> 2/2 checks ...	None	ec2-52-37-198-89.u
<input type="checkbox"/>	reallyCoolAp...	i-7d7c54ba	t1.micro	us-west-2a	<span>●</span> terminated		None	

You've learned how to run two types of low-interruption, high-automation deploys on AWS Elastic Beanstalk.