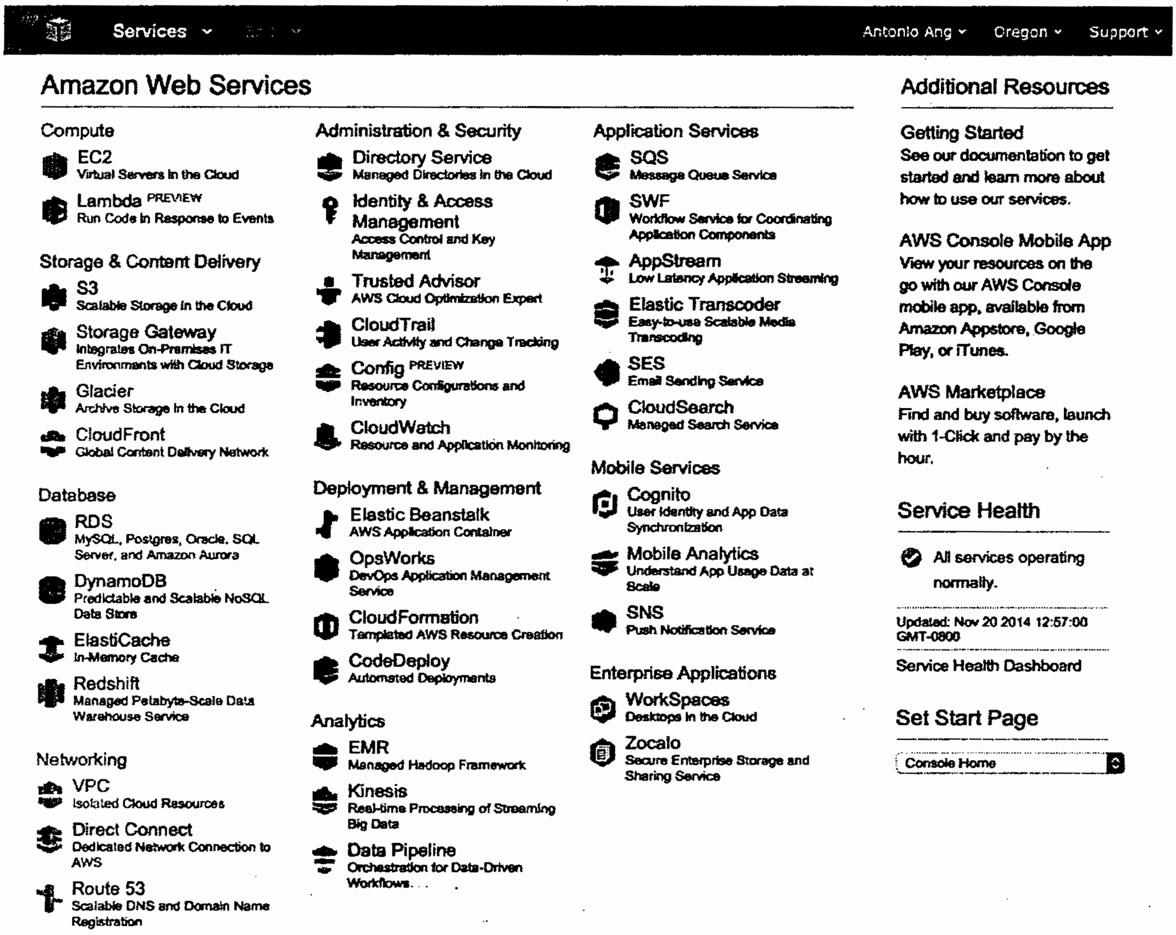
Lab 17

DEPLOY A PHP APPLICATION USING ELASTIC BEANSTALK

STEP 1: Log In to the Amazon Web Service Console

This laboratory experience is about Amazon Web Services and you will use the AWS Management Console in order to complete all the lab steps.



Amazon Web Services

Compute
- EC2 Virtual Servers in the Cloud
- Lambda PREVIEW Run Code in Response to Events

Storage & Content Delivery
- S3 Scalable Storage in the Cloud
- Storage Gateway Integrates On-Premises IT Environments with Cloud Storage
- Glacier Archive Storage in the Cloud
- CloudFront Global Content Delivery Network

Database
- RDS MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
- DynamoDB Predictable and Scalable NoSQL Data Store
- ElastiCache In-Memory Cache
- Redshift Managed Petabyte-Scale Data Warehouse Service

Networking
- VPC Isolated Cloud Resources
- Direct Connect Dedicated Network Connection to AWS
- Route 53 Scalable DNS and Domain Name Registration

Administration & Security
- Directory Service Managed Directories in the Cloud
- Identity & Access Management Access Control and Key Management
- Trusted Advisor AWS Cloud Optimization Expert
- CloudTrail User Activity and Change Tracking
- Config PREVIEW Resource Configurations and Inventory
- CloudWatch Resource and Application Monitoring

Deployment & Management
- Elastic Beanstalk AWS Application Container
- OpsWorks DevOps Application Management Service
- CloudFormation Templated AWS Resource Creation
- CodeDeploy Automated Deployments

Analytics
- EMR Managed Hadoop Framework
- Kinesis Real-time Processing of Streaming Big Data
- Data Pipeline Orchestration for Data-Driven Workflows

Application Services
- SQS Message Queue Service
- SWF Workflow Service for Coordinating Application Components
- AppStream Low Latency Application Streaming
- Elastic Transcoder Easy-to-use Scalable Media Transcoding
- SES Email Sending Service
- CloudSearch Managed Search Service

Mobile Services
- Cognito User Identity and App Data Synchronization
- Mobile Analytics Understand App Usage Data at Scale
- SNS Push Notification Service

Enterprise Applications
- WorkSpaces Desktops in the Cloud
- Zocalo Secure Enterprise Storage and Sharing Service

Additional Resources

Getting Started
See our documentation to get started and learn more about how to use our services.

AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.

Service Health

✓ All services operating normally.

Updated: Nov 20 2014 12:57:00 GMT-0800

Service Health Dashboard

Set Start Page

Console Home

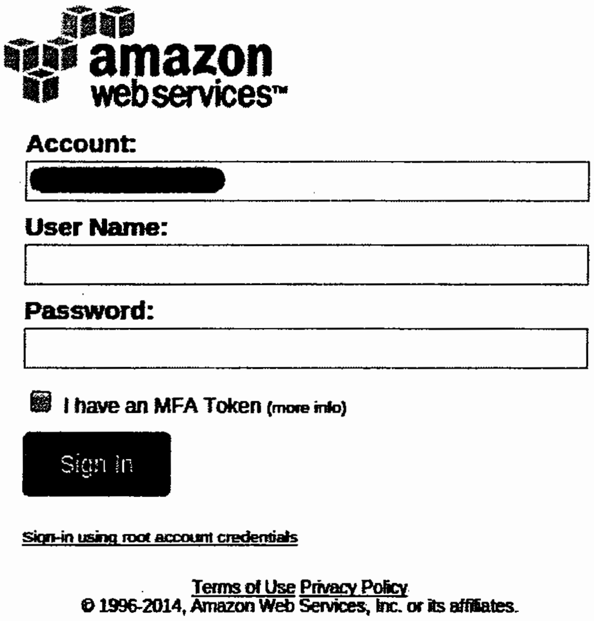
The AWS Management Console is a web control panel for managing all your AWS resources, from EC2 instances to SNS topics. The console enables cloud management for all aspects of the AWS account, including managing security credentials, or even setting up new IAM Users.

Log in to the AWS Management Console

In order to start the laboratory experience, open the Amazon Console by clicking this button:

Open AWS Console

Log in with the username **xxxx** and the password **xxxx**



Account:

[redacted]

User Name:

Password:

☒ I have an MFA Token (more info)

Sign In

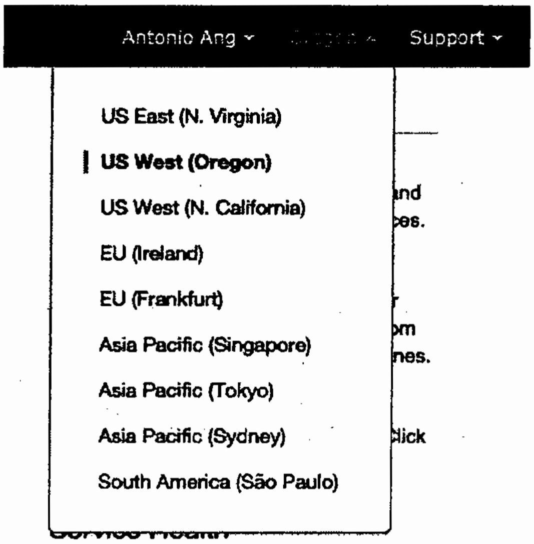
Sign-in using root account credentials

Terms of Use Privacy Policy
© 1996-2014, Amazon Web Services, Inc. or its affiliates.

Select the right AWS Region

Amazon Web Services is available in different regions all over the world, and the console lets you provision resources across multiple regions. You usually choose a region that best suits your business needs to optimize your customer's experience, but you must use the region **US West (Oregon)** for this laboratory.

You can select the **US West (Oregon)** region using the upper right dropdown menu on the AWS Console page.



US East (N. Virginia)

US West (Oregon)

US West (N. California)

EU (Ireland)

EU (Frankfurt)

Asia Pacific (Singapore)

Asia Pacific (Tokyo)

Asia Pacific (Sydney)

South America (São Paulo)

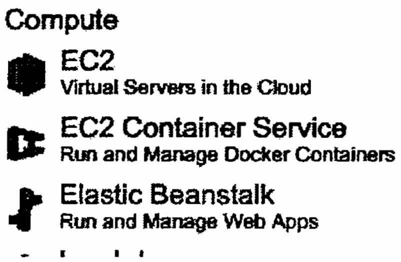
STEP 2: Deploy a sample Beanstalk application

Amazon's Elastic Beanstalk has been around for a while -- since 2011 -- but it's still a very viable Platform-as-a-Service. Beanstalk itself costs nothing, as AWS only charges for the underlying resources. In the case of a simple application, that's just the load balancers, EC2 instances, and network bandwidth to host your app.

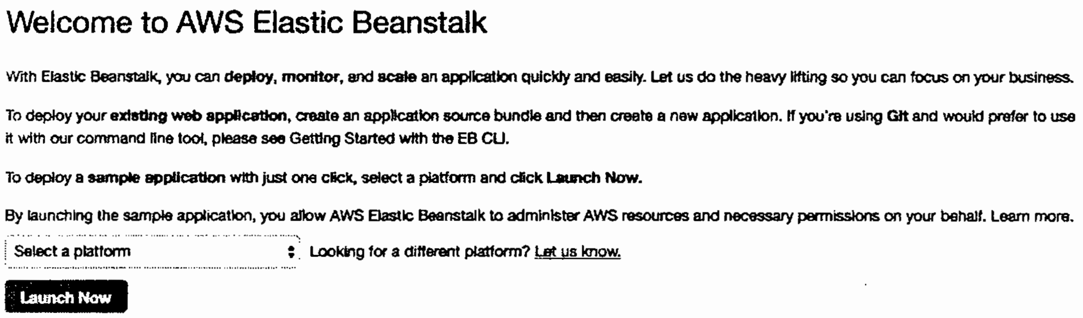
In this lab, we'll start by deploying an example app, written in PHP, then update the code and redeploy to see Beanstalk's automation in action.

Running a web application in AWS and building automation yourself isn't necessary with Elastic Beanstalk, but it ties together EC2, RDS, Elastic Load Balancer, and more. It also comes with a deploy system, and the ability to roll back versions.

To get started, click on the Beanstalk icon from the AWS console.



In Elastic Beanstalk, an **application** is a web application supplied in a specific format. An application can be deployed to multiple environments. In this lab we'll use a single environment but you may want to have production, staging, and testing environments.



Welcome to AWS Elastic Beanstalk

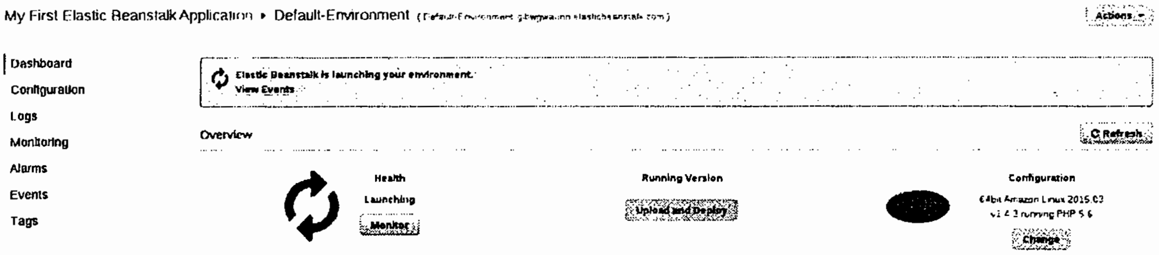
With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

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.

Hit "Launch Now" after selecting **PHP** from the dropdown menu.



My First Elastic Beanstalk Application ▸ Default-Environment (Default-Environment [illegible].elasticbeanstalk.com)

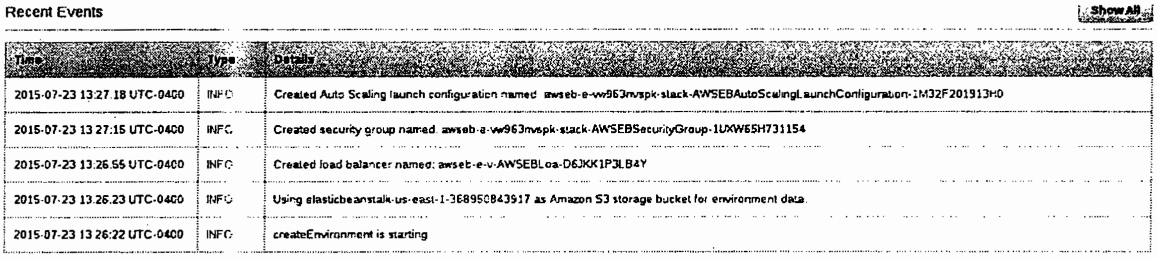
Elastic Beanstalk is launching your environment.
View Events

Health: Launching

Running Version

Configuration: 64bit Amazon Linux 2015.03 v1.4.3 running PHP 5.6

The new application will start launching immediately, and soon you should see progress on the dashboard.



Recent Events

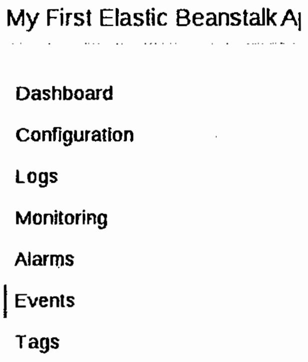
Time	Type	Details
2015-07-23 13:27:18 UTC-0400	INFO	Created Auto Scaling launch configuration named: awseb-e-vw963mvspk-stack-AWSEBAutoScalingLaunchConfiguration-1M32F201913H0
2015-07-23 13:27:15 UTC-0400	INFO	Created security group named: awseb-e-vw963mvspk-stack-AWSEBSecurityGroup-1UXW65H731154
2015-07-23 13:26:56 UTC-0400	INFO	Created load balancer named: awseb-e-v-AWSEBLoa-D6JKK1P3LB4Y
2015-07-23 13:26:23 UTC-0400	INFO	Using elasticbeanstalk-us-east-1-368950843917 as Amazon S3 storage bucket for environment data.
2015-07-23 13:26:22 UTC-0400	INFO	createEnvironment is starting

You can watch in the logs as each needed resource is created. In addition to an EC2 instance, Beanstalk must create a load balancer, autoscaling group, and network security groups. The S3 bucket is created to hold deploy artifacts (versions of the application).

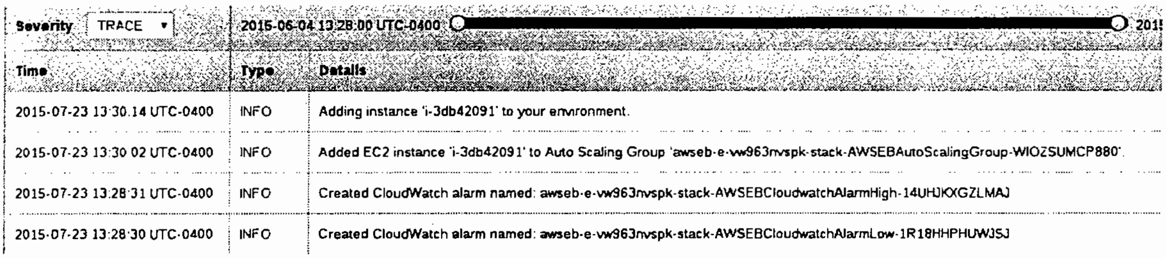
While we wait for Elastic Beanstalk to create the infrastructure, we can take a tour of the different tabs in the next step.

STEP 3: Tour of the Beanstalk Console

On the left side, head to the "Events" tab.

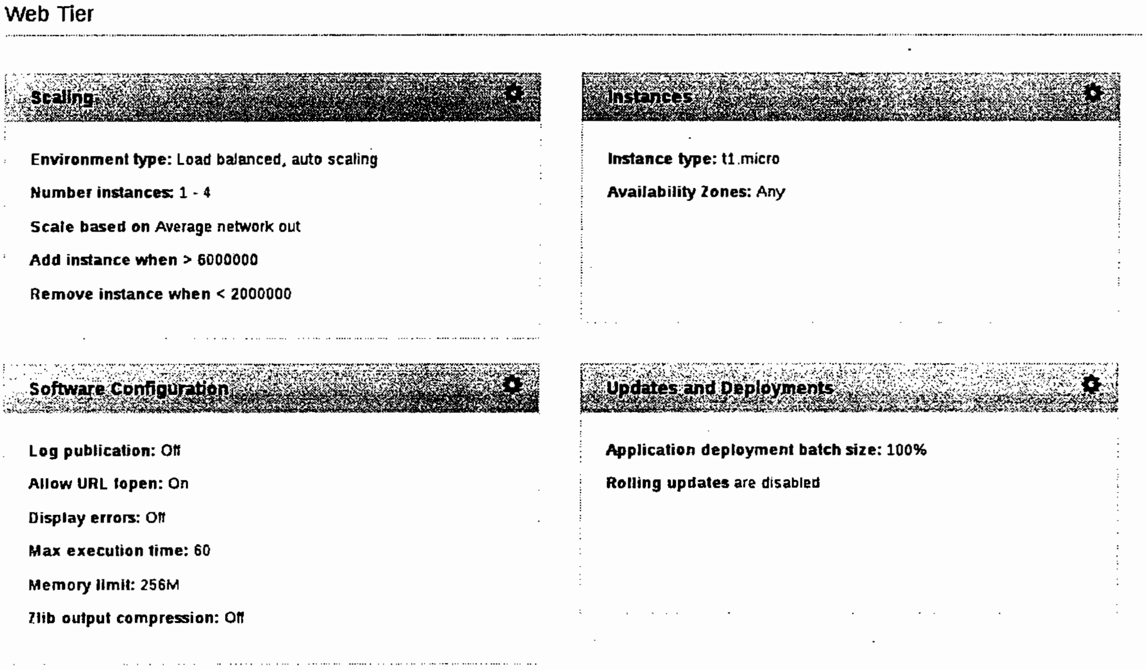


When you redeploy, or want to see what's happening in the infrastructure, this is the first place to check. You'll see that the infrastructure for our demo app is still in progress, but that the instance it will run on has started.



Time	Type	Details
2015-07-23 13:30.14 UTC-0400	INFO	Adding instance 'i-3db42091' to your environment.
2015-07-23 13:30 02 UTC-0400	INFO	Added EC2 instance 'i-3db42091' to Auto Scaling Group 'awseb-e-vw963nvspk-stack-AWSEBAutoScalingGroup-WIOZSUMCP880'.
2015-07-23 13:28:31 UTC-0400	INFO	Created CloudWatch alarm named: awseb-e-vw963nvspk-stack-AWSEBCloudwatchAlarmHigh-14UHJKXGZLMAJ
2015-07-23 13:28:30 UTC-0400	INFO	Created CloudWatch alarm named: awseb-e-vw963nvspk-stack-AWSEBCloudwatchAlarmLow-1R18HHPHUWJSJ

Now head over to the configuration tab, and we can see the available options for your deployment.



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

Instances

Instance type: t1.micro
Availability Zones: Any

Software Configuration

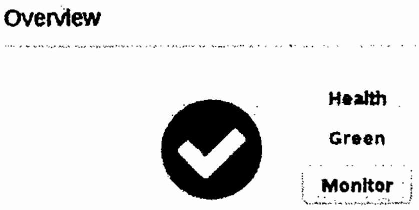
Log publication: Off
Allow URL fopen: On
Display errors: Off
Max execution time: 60
Memory limit: 256M
Zlib output compression: Off

Updates and Deployments

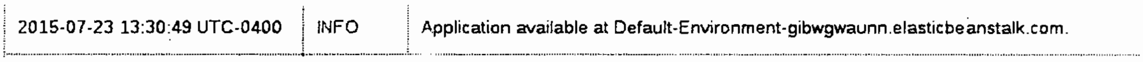
Application deployment batch size: 100%
Rolling updates are disabled

We can see that there are options to change the resources allocated to PHP, change when the app should scale up or down, or make each instance larger. Beanstalk has no shortage of features to tweak. If there's a setting you'd like to modify that Beanstalk doesn't have, you can accomplish this in the specific console for that service. Everything in Elastic Beanstalk is a standard AWS resource, but is managed for you.

Switch back to the main environment page. By now the deployment should be finished and you should see a big green checkmark for the application health.

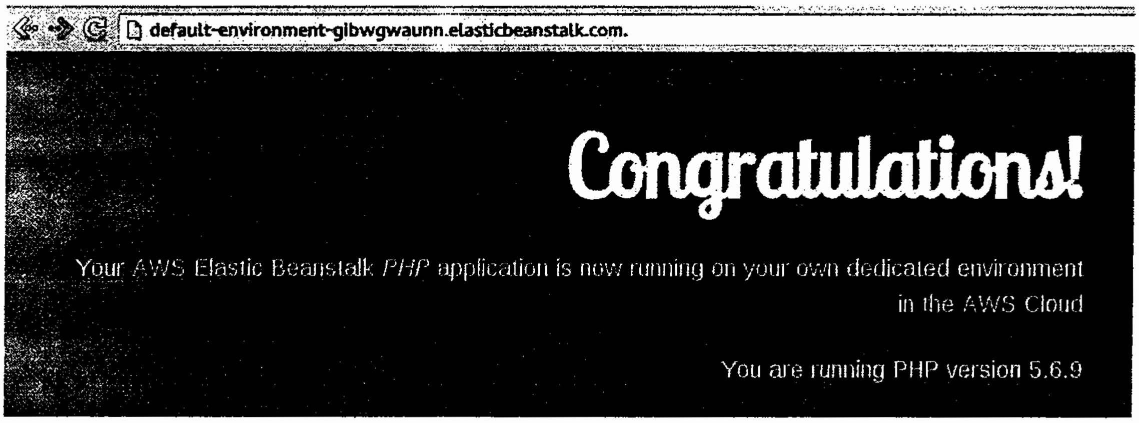


Find the URL for your application by checking the event log. Look for a line like this:



```
2015-07-23 13:30:49 UTC-0400 | INFO | Application available at Default-Environment-gibwgwaunn.elasticbeanstalk.com.
```

Copy that URL into your browser, and you should get the default application page.



default-environment-gibwgwaunn.elasticbeanstalk.com.

Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 5.6.9

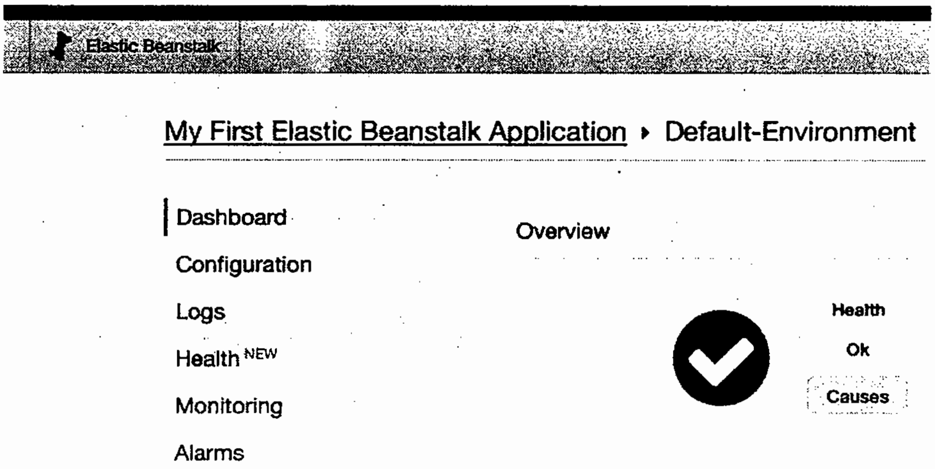
Now that your deployment is working, let's make a code change and deploy a new version.

STEP 4: Deploy a New Version

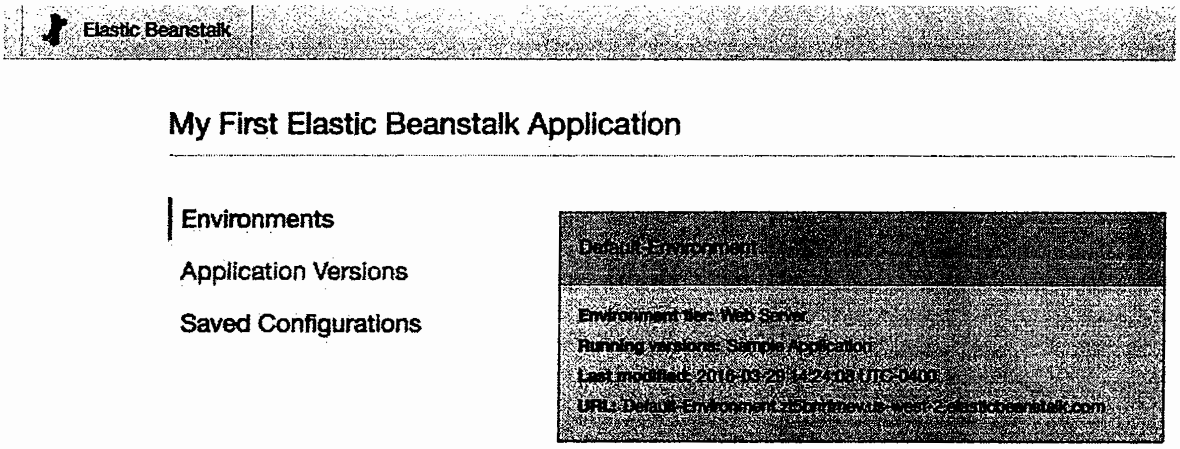
The code for the next version (really just a small change) of the sample application is available on Vepsun's Github, including a zipfile of the new code. Elastic Beanstalk stores each version as its own artifact -- which is a zip file including the code and some Elastic Beanstalk configurations.

In the new version, you can see we've changed "Congratulations" to "New version is deployed!" Download the zip file and we'll see how to deploy the new code.

Click on **My First Elastic Beanstalk Application**



Then select the "Application Versions" option from the application menu.

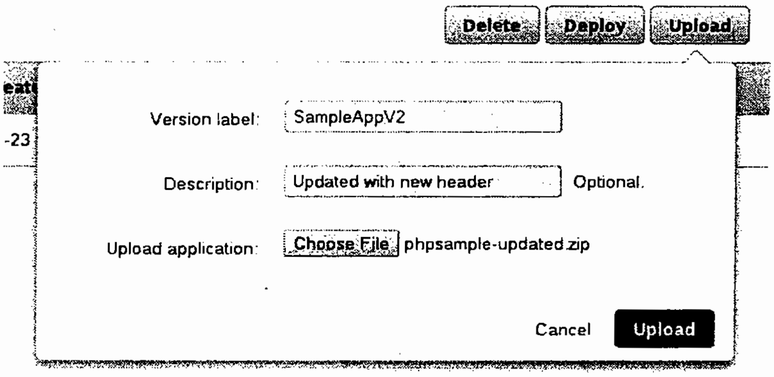


My First Elastic Beanstalk Application

Default-Environment

Environment tier: Web Server
Running versions: Sample Application
Last modified: 2016-03-29 14:24:08 UTC-0400
URL: Default-Environment.[illegible].us-west-2.elasticbeanstalk.com

In the upper right, there's an **Upload** button where you can add new versions of your code. Click it and put in a version name for the changes.

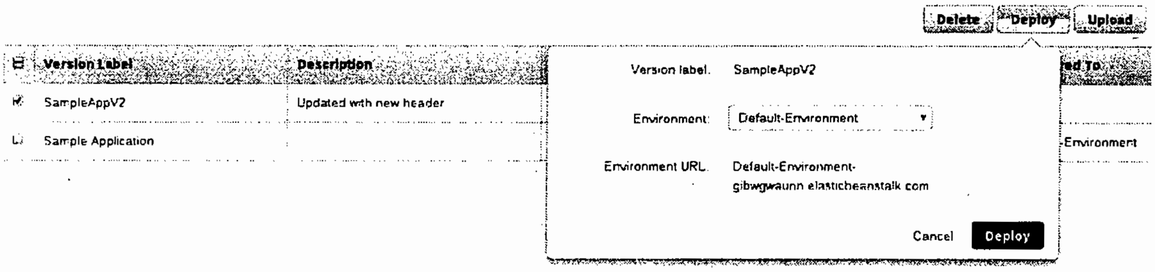


Version label: SampleAppV2

Description: Updated with new header Optional.

Upload application: Choose File phpsample-updated.zip

Once the upload is successful, you're ready to deploy. Select the V2 we uploaded, and clickDeploy



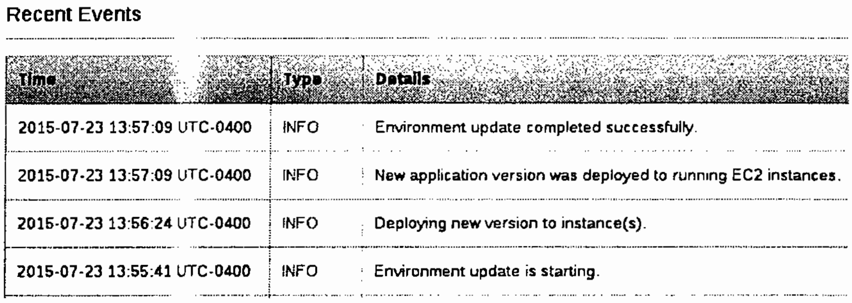
☐	Version Label	Description
☒	SampleAppV2	Updated with new header
☐	Sample Application	

Version label. SampleAppV2

Environment: Default-Environment

Environment URL. Default-Environment-gibwgwaunn elasticbeanstalk com

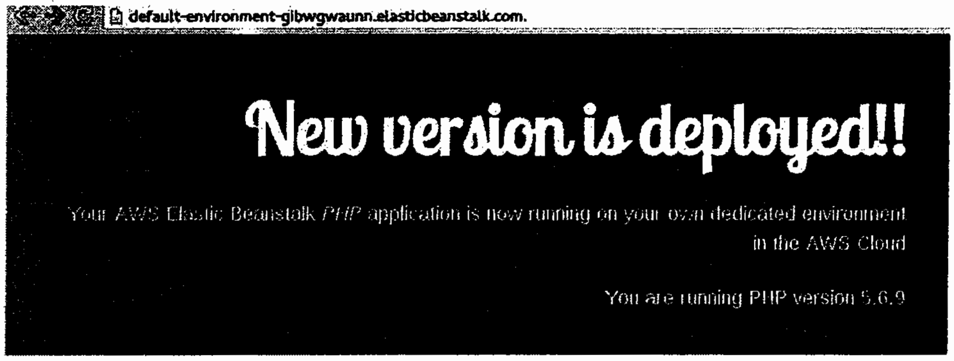
You can go back to the Events page or the Environment Dashboard to watch as Beanstalk deploys the new code. You should see an "update successful" message within a few minutes.



Recent Events

Time	Type	Details
2015-07-23 13:57:09 UTC-0400	INFO	Environment update completed successfully.
2015-07-23 13:57:09 UTC-0400	INFO	New application version was deployed to running EC2 instances.
2015-07-23 13:56:24 UTC-0400	INFO	Deploying new version to instance(s).
2015-07-23 13:55:41 UTC-0400	INFO	Environment update is starting.

Now that your deploy is complete, go back to the application URL you opened earlier to see the new version in action.



default-environment-gibwgwaunn.elasticbeanstalk.com.

New version is deployed!!

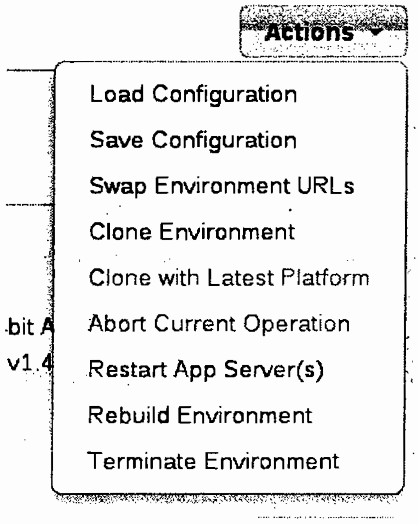
Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 5.6.9

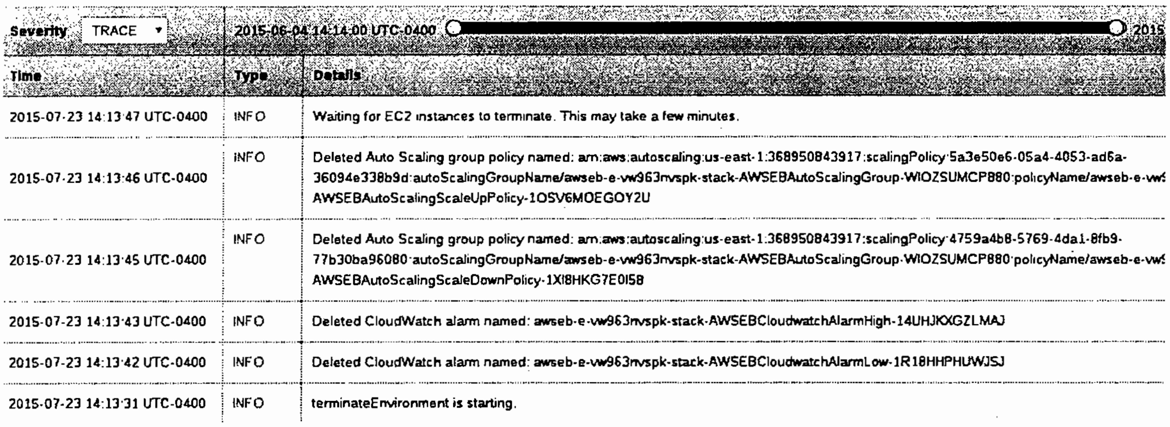
STEP 5: Delete the Application

Now that we've created and updated our application, it's time to clean up the AWS resources we used.

Go to the "Actions" menu to terminate the environment.

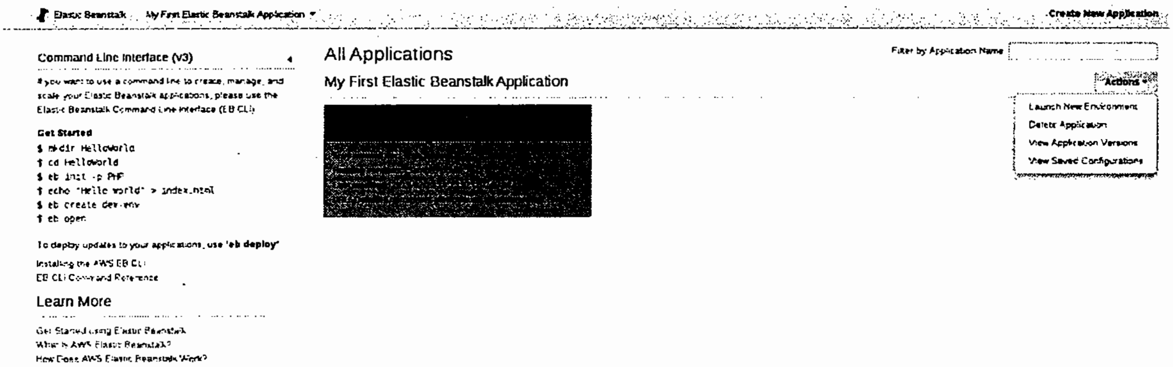


It can take up to 10 minutes to delete all the resources (including the EC2 instance). You can watch the progress in the Events tab.



Time	Type	Details
2015-07-23 14:13:47 UTC-0400	INFO	Waiting for EC2 instances to terminate. This may take a few minutes.
2015-07-23 14:13:46 UTC-0400	INFO	Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:368950843917:scalingPolicy:5a3e50e6-05a4-4053-ad6a-36094e338b9d:autoScalingGroupName/awseb-e-vw963nvspk-stack-AWSEBAutoScalingGroup-WIOZSUMCP880:policyName/awseb-e-vw AWSEBAutoScalingScaleUpPolicy-1OSV6MOEGOY2U
2015-07-23 14:13:45 UTC-0400	INFO	Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:368950843917:scalingPolicy:4759a4b8-5769-4da1-8fb9-77b30ba96080:autoScalingGroupName/awseb-e-vw963nvspk-stack-AWSEBAutoScalingGroup-WIOZSUMCP880:policyName/awseb-e-vw AWSEBAutoScalingScaleDownPolicy-1XI8HKG7E0I58
2015-07-23 14:13:43 UTC-0400	INFO	Deleted CloudWatch alarm named: awseb-e-vw963nvspk-stack-AWSEBCloudwatchAlarmHigh-14UHJKXGZLMAJ
2015-07-23 14:13:42 UTC-0400	INFO	Deleted CloudWatch alarm named: awseb-e-vw963nvspk-stack-AWSEBCloudwatchAlarmLow-1R18HHPHUWJSJ
2015-07-23 14:13:31 UTC-0400	INFO	terminateEnvironment is starting.

After the environment is deleted, do the same for the application. An application can have many environments, such as production, staging, and testing.


Elastic Beanstalk
My First Elastic Beanstalk Application
Create New Application
Filter by Application Name
Actions
Launch New Environment
Delete Application
View Application Versions
View Saved Configurations

Command Line Interface (V3)

If you want to use a command line to create, manage, and scale your Elastic Beanstalk applications, please use the Elastic Beanstalk Command Line Interface (EB CLI)

Get Started

```
$ mkdir HelloWorld
$ cd HelloWorld
$ eb init -p PHP
$ echo "Hello World" > index.html
$ eb create dev-env
$ eb open
```

To deploy updates to your applications, use 'eb deploy'

Installing the AWS EB CLI
EB CLI Command Reference

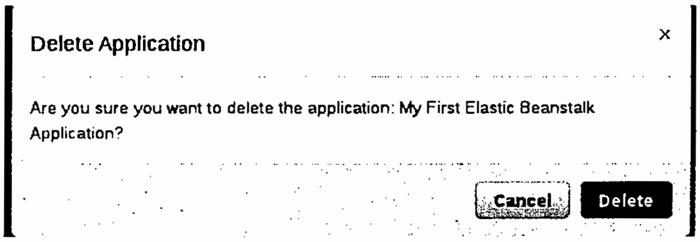
Learn More

Get Started using Elastic Beanstalk
What is AWS Elastic Beanstalk?
How Does AWS Elastic Beanstalk Work?

All Applications

My First Elastic Beanstalk Application

You'll need to confirm this because Elastic Beanstalk will be deleting the code we uploaded and all prior versions.



Delete Application

Are you sure you want to delete the application: My First Elastic Beanstalk Application?

Congratulations! You've finished the full lab.

In review, we've learned that Elastic Beanstalk lets you focus on building your application by handling infrastructure for you. It manages scaling, provisioning, and deployments without requiring your intervention (well, except writing the code).