**Blue/Green Deployment Pattern with AWS Elastic Beanstalk | Qwiklabs**

**SPL-47 - Version 1.4.8**

© 2020 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.

Errors, corrections or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab Overview**

The purpose of this lab is to show you how to deploy a simple web application continuously using The Elastic Beanstalk Command Line Interface (EB CLI).

AWS Elastic Beanstalk provides a quick and easy way to deploy your web applications to the AWS cloud without requiring knowledge of the individual pieces that make up the infrastructure required to deploy your app to the cloud.

EB CLI is a command line interface for Elastic Beanstalk that provides interactive commands that simplify creating, updating and monitoring environments from a local repository. You can use it to automate deployment tasks and common administrative tasks in AWS.

**Lab Description**

This lab will demonstrate:

* Deploying web application versions on an Elastic Beanstalk environment
* Managing environments with the EB CLI
* Deploying a new version of your application in Rolling update (In-Place Deployment)
* Deploying a new version of your application in Blue/Green Deployment (Red/Black Deployment)

**Lab Pre-requisites**

To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line.

**Other AWS Services**

Other AWS Services than the ones needed for this lab are disabled by IAM policy during your access time in this lab. In addition, the capabilities of the services used in this lab are limited to what's required by the lab and in some cases are even further limited as an intentional aspect of the lab design. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

**Start Lab**

1. At the top of your screen, launch your lab by clicking Start Lab

This will start the process of provisioning your lab resources. An estimated amount of time to provision your lab resources will be displayed. You must wait for your resources to be provisioned before continuing.

If you are prompted for a token, use the one distributed to you (or credits you have purchased).

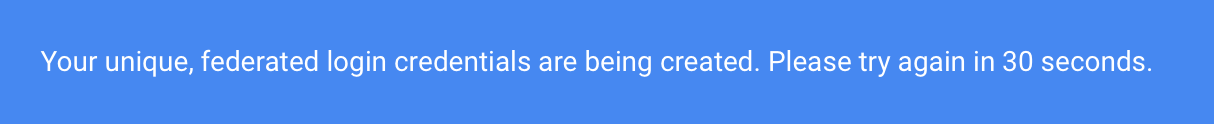
1. Open your lab by clicking Open Console

This will automatically log you into the AWS Management Console.

**Please do not change the Region unless instructed**.

**Common login errors**

**Error : Federated login credentials**



If you see this message:

* Close the browser tab to return to your initial lab window
* Wait a few seconds
* Click Open Console again

You should now be able to access the AWS Management Console.

**Error: You must first log out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Click **click here**
* Close your browser tab to return to your initial Qwiklabs window
* Click Open Console again

**Task 1: Connect to Your Amazon EC2 Instance**

**Windows Users: Using SSH to Connect**

These instructions are for Windows users only.

If you are using Mac or Linux, [skip to the next section](about:reader?url=https%3A%2F%2Famazon.qwiklabs.com%2Ffocuses%2F14187%3Fcatalog_rank%3D%257B%2522rank%2522%253A2%252C%2522num_filters%2522%253A0%252C%2522has_search%2522%253Atrue%257D%26parent%3Dcatalog%26search_id%3D7463934#ssh-MACLinux).

1. To the left of the instructions you are currently reading, click **Download PPK**.
2. Save the file to the directory of your choice.

You will use PuTTY to SSH to Amazon EC2 instances.

If you do not have PuTTY installed on your computer, [download it here](https://the.earth.li/%7Esgtatham/putty/latest/w64/putty.exe).

1. Open PuTTY.exe
2. Configure the PuTTY to not timeout:

* Click **Connection**
* Set **Seconds between keepalives** to

This allows you to keep the PuTTY session open for a longer period of time.

1. Configure your PuTTY session:

* Click **Session**
* **Host Name (or IP address):** Copy and paste the **EC2PublicIp** of your instance
* In the **Connection** list, expand **SSH**
* Click **Auth** (don't expand it)
* Click Browse
* Browse to and select the PPK file that you downloaded
* Click Open to select it
* Click Open

1. Click **Yes**, to trust the host and connect to it.
2. When prompted **login as**, enter:

This will connect to your EC2 instance.

1. [Windows Users: Click here to skip ahead to the next task.](about:reader?url=https%3A%2F%2Famazon.qwiklabs.com%2Ffocuses%2F14187%3Fcatalog_rank%3D%257B%2522rank%2522%253A2%252C%2522num_filters%2522%253A0%252C%2522has_search%2522%253Atrue%257D%26parent%3Dcatalog%26search_id%3D7463934#ssh-after)

**Mac and Linux Users**

These instructions are for Mac/Linux users only. If you are a Windows user, [skip ahead to the next task.](about:reader?url=https%3A%2F%2Famazon.qwiklabs.com%2Ffocuses%2F14187%3Fcatalog_rank%3D%257B%2522rank%2522%253A2%252C%2522num_filters%2522%253A0%252C%2522has_search%2522%253Atrue%257D%26parent%3Dcatalog%26search_id%3D7463934#ssh-after)

1. To the left of the instructions you are currently reading, click **Download PEM**.
2. Save the file to the directory of your choice.
3. Copy this command to a text editor:

chmod 400 KEYPAIR.pem

ssh -i KEYPAIR.pem ec2-user@EC2PublicIP

1. Replace *KEYPAIR.pem* with the path to the PEM file you downloaded.
2. Replace *EC2PublicIP* with the **EC2PublicIp** of your instance.
3. Paste the updated command into the Terminal window and run it.
4. Type when prompted to allow a first connection to this remote SSH server.

Because you are using a key pair for authentication, you will not be prompted for a password.

**Task 2: Configure the EB CLI (All Users)**

In this task, you will install and configure the Elastic Beanstalk CLI.

1. Install the EB CLI.

git clone https://github.com/aws/aws-elastic-beanstalk-cli-setup.git

./aws-elastic-beanstalk-cli-setup/scripts/bundled\_installer --location /home/ec2-user

echo 'export PATH="/home/ec2-user/.ebcli-virtual-env/executables:$PATH"' >> ~/.bash\_profile && source ~/.bash\_profile

This will take a few minutes to complete.

1. Create a working directory for the Elastic Beanstalk files and then navigate to them.

mkdir ~/eb-lab

cd ~/eb-lab

1. Initialize an Elastic Beanstalk project.

eb init

After executing this command, you will configure EB CLI interactively and create an application that is a component of AWS Elastic Beanstalk.

1. Using the interactive wizard, configure:

* **Select a default region:** Enter the number that corresponds your **REGION** value located to the left of these instructions
* **Enter Application Name:** Press *Enter*
* **Select a platform:** Enter the number for *PHP*
* **Select a platform version:** Enter the number for *PHP 7.4 running on 64bit Amazon Linux 2*
* **Do you want to set up SSH for your instances?**
  + Enter
  + Press **Enter**

**Task 3: Deploy Your Application**

At this point, the AWS Elastic Beanstalk configuration for your application has been created. The configuration is stored in the ".elasticbeanstalk" sub-directory under your application directory "eb-lab". If you are curious, you can inspect its contents to see what kind of information it contains.

In this task, you will create the AWS Elastic Beanstalk environment and deploy the source code to it.

1. Navigate to the /tmp directory and download the source code package to deploy.

cd /tmp

wget https://us-west-2-aws-training.s3.us-west-2.amazonaws.com/awsu-spl/spl-47/1.4.8.prod/scripts/ec2-info.zip

1. Unpack the zip file.

unzip ec2-info.zip

1. Copy the source code to your Elastic Beanstalk working directory.

cp -r ec2-info/\* ~/eb-lab/

1. Navigate to your Elastic Beanstalk working directory.

cd ~/eb-lab/

1. Create AWS Elastic Beanstalk environment.

eb create --vpc --service-role aws-elasticbeanstalk-service-role

**Note:** The role **aws-elasticbeanstalk-service-role** was created as part of the lab setup. You're passing that role to the Elastic Beanstalk service which provides it with the necessary permissions to launch the environment needed for this lab.

1. Using the interactive wizard, configure:

* **Enter Environment Name:** Press *Enter*
* **Enter DNS CNAME prefix:** Enter
* Replace **NUMBER** with a *long* random string of numbers, then press **Enter**
* **Select a load balancer type:** Enter the number for **application**
* **Would you like to enable Spot Fleet requests...** Enter
* **Enter the VPC ID:** Paste the value of **VPCID** located to the left of these instructions
* **Do you want to associate a public IP address?:**
* **Enter a comma-separated list of Amazon EC2 subnets:**
  + Paste
  + Replace SUBNETA with the value of SUBNETA located to the left of these instructions
  + Replace SUBNETB with the value of SUBNETB located to the left of these instructions
  + Press **Enter**
* **Enter a comma-separated list of Amazon ELB subnets:** Press **Enter**
* **Do you want the load balancer to be public? (Select no for internal):**
* **Enter a comma-separated list of Amazon VPC security groups:** Press **Enter**

The prompt will start outputting the status of an environment as it uploads your project and builds the environment.

1. Once you see **(Safe to Ctrl+C**, press **Ctrl + C** to escape the prompt.
2. Check the status of your deployment.

eb status

1. Wait for the following to be true before proceeding to the next step:

* **Status:** *Ready*
* **Health:** *Green*

The deployment may take 5-7 minutes.

1. Install a text-based web browser to view your application.

sudo yum -y install lynx

1. Open the web page you deployed using the Elastic Beanstalk CLI.

eb open

Verify that there is information about ec2 metadata.

1. Exit the program by:

* Pressing
* Pressing

1. In the **AWS Management Console**, on the Services menu, click **Elastic Beanstalk**.

You should see the application that you deployed.

1. Click on your application.
2. Copy the **URL** at the top of the page to a text editor.
3. Click the URL.

It should look similar to: *myeb45345345.us-east-1.elasticbeanstalk.com*.

You should see the same metadata.

**Task 4: Perform System Operations Using Elastic Beanstalk CLI**

You have deployed version "v1" of your web application to Elastic Beanstalk and you are able to see it in your browser.

In this task, you will manage your Elastic Beanstalk Environments with the EB CLI. You will:

* Monitor the environment
* Scale your infrastructure
* Pull logs and events
* Update the configuration

1. In your SSH session check your working directory.

pwd

You should be in **/home/ec2-user/eb-lab**

1. If you are not in **/home/ec2-user/eb-lab**, run the following command:

cd ~/eb-lab

1. Display the events output by Elastic Beanstalk.

eb events

This shows the Elastic Beanstalk monitors the environment launch process.

1. Verify that **ERROR** events do not occur.
2. Escape from this prompt entering
3. View the most recent health for your environment.

LC\_TIME="en\_US.UTF-8" eb health --refresh

This command outputs information about your environment health (e.g. The average of CPU utilization for each instance, information from the web server logs on each instance, and so on).

1. Escape from this prompt typing

**LC\_TIME** is Date and time formats and if it's not **en\_US.UTF-8**, the error occurs. The default setting of some AMIs' LC\_TIME is not **en\_US.UTF-8** so that the locale is set before "eb health" on this step.

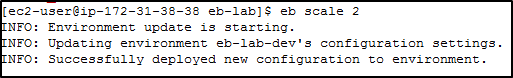
1. Pull logs from an instance in your environment and display them in standard output.

eb logs

1. After viewing the logs, escape the prompt by entering
2. Scale the AWS Elastic Beanstalk Environment.

eb scale 2

1. Wait for output that shows the following:



1. Verify that the scaling operation has finished.

eb status

* **Status** should display: *Ready*
* **Health** should display: *Green*

1. View the Elastic Beanstalk configuration options (e.g. Auto Scaling settings, ELB health check settings).

eb config

1. Escape from the output typing **Ctrl + X**
2. Save the current settings:

eb config save --cfg eb-lab-config

After executing this command, Elastic Beanstalk saves the file to the folder **.elasticbeanstalk/saved\_configs/**. You can also save the file with an upload the file an Amazon S3 Bucket using:

1. Configure the **EDITOR** environment variable to open eb config output in vim.

export EDITOR=vim

1. View the configuration again.

eb config

1. Press **/** Then enter
2. Change to insert mode by pressing
3. Between the quotes after **Application Healthcheck URL:**, add
4. Save the file by:

* Pressing **Escape**
* Entering

The Elastic Load Balancer used to query for instance health will change.

1. Verify the configuration change has finished.

eb status

* **Status** should display: *Ready*
* **Health** should display: *Green*

**Task 5: Update Your Application Using Rolling Deployments**

You deployed your application on two EC2 instances, behind a load balancer. In this task, you will change the source code and update the environment using in-place deployment pattern (Rolling Deployment pattern).

**What is In-place Deployments (Rolling Deployments)**

In-place deployments are the deployment pattern that keep infrastructure used by current version and deploy new version of source code to the same infrastructure. The merit of it is quickness and cost saving because using the existing instances where the OS installation, configuration and launch has been already done.

With rolling deployments, Elastic Beanstalk splits the environments into batches and deploys to one batch at a time, leaving the rest of the instances in the environment running the old application version. In the middle of a rolling deployment, some instances can be serving requests with the old version, while others from an already completed batch could be serving other requests with the new version at the same time.

1. In your SSH session, navigate to the working directory.

cd ~/eb-lab

1. Open index.php using nano.

nano index.php

1. Change to
2. Save the file by:

* Pressing **Ctrl + X**
* Entering
* Pressing **Enter**

1. Deploy the new version of source code.

eb deploy

1. Once you see **(Safe to Ctrl+C**, press **Ctrl + C** to escape the prompt.
2. Verify that the deployment is finished.

eb status

* **Status** should display: *Ready*
* **Health** should display: *Green*

1. Open the web page.

eb open

1. Verify that header changed to **Instance Metadata v1.1 with eb deployment**.
2. Exit the metadata page:

* Enter
* Enter

**Task 6: Update Your Application using Blue/Green(Red/Black) Deployments**

Your application is running on Amazon EC2 instances as version 1.1.

In this task, you will deploy the new version of application using Blue/Green(Red/Black) Deployments as version 2.0.

**What is Blue/Green (Red/Black) Deployments**

Blue/Green (Red/Black) Deployments are the deployment patter that uses two environments called "Blue", it means the current version and "Green", it means the new version. It uses new infrastructures (called "Green") for the new revision source code and replace the current environment with the new one. There are several patterns of Blue/Green Deployments. One of its patterns, called Red/Black Deployments, attempts to deploy an application revision to as many instances as possible at once. The merit of Blue/Green (Red/Black) Deployments is stability because we don't change the current environments and it's easy to rollback.

With Blue/Green (Red/Black) Deployments using AWS Elastic Beanstalk, you will deploy the new version to a separate environment, and then swap CNAMEs of the two environments to redirect traffic to the new version instantly.

1. In your SSH session, navigate to the working directory.

cd ~/eb-lab

1. Open **index.php**.

nano index.php

1. Change to
2. Save the file by:

* Pressing **Ctrl + X**
* Entering
* Pressing **Enter**

1. Create a new AWS Elastic Beanstalk Environment.

eb create --vpc --service-role aws-elasticbeanstalk-service-role New-EB -c New-EB

1. Using the interactive wizard, configure:

* **Enter the VPC ID:** Paste the value of **VPCID** located to the left of these instructions
* **Do you want to associate a public IP address?:**
* **Enter a comma-separated list of Amazon EC2 subnets:**
  + Paste
  + Replace SUBNETA with the value of SUBNETA located to the left of these instructions
  + Replace SUBNETB with the value of SUBNETB located to the left of these instructions
  + Press **Enter**
* **Enter a comma-separated list of Amazon ELB subnets:** Press **Enter**
* **Do you want the load balancer to be public? (Select no for internal):**
* **Enter a comma-separated list of Amazon VPC security groups:** Press **Enter**

1. Once you see **(Safe to Ctrl+C**, press **Ctrl + C** to escape the prompt.
2. Verify that the deployment is finished.

eb status New-EB

1. Wait till the following is true:

* **Status** should display: *Ready*
* **Health** should display: *Green*

1. Open the web page to your new environment.

eb open New-EB

1. Verify that the header changed to **Instance Metadata v2.0 with eb swap**.
2. In the **AWS Management Console**, on the Services menu, click **Elastic Beanstalk**.
3. Click On **New-EB** URL.

This will open another tab showing you the new environment that you deployed.

1. In your SSH session replace the environment you created with the new environment:

eb swap New-EB --destination\_name eb-lab-dev

1. Wait for a message in the output that shows *Completed swapping CNAMEs for environments*
2. Verify the swap.

eb status New-EB

1. The following should be true:

* **Status** should display: *Ready*
* **Health** should display: *Green*

1. Open the URL for your **eb-lab-dev** environment.
2. Verify that the header is **Instance Metadata v2.0 with eb swap**

You don't need the old environment anymore. Therefore, you can terminate it.

1. In your SSH session, enter:

eb terminate eb-lab-dev --force

1. You can watch the progress of termination.

eb status eb-lab-dev

If you see the message: **ERROR: Environment <first environment> not Found**, it means your environment is terminated.

**End Lab**

Follow these steps to close the console, end your lab, and evaluate the experience.

1. Return to the AWS Management Console.
2. On the navigation bar, click **awsstudent@<AccountNumber>**, and then click **Sign Out**.
3. Click End Lab
4. Click OK
5. (Optional):

* Select the applicable number of stars
* Type a comment
* Click **Submit**
  + 1 star = Very dissatisfied
  + 2 stars = Dissatisfied
  + 3 stars = Neutral
  + 4 stars = Satisfied
  + 5 stars = Very satisfied

You may close the dialog if you don't want to provide feedback.

You may close the dialog if you don't want to provide feedback.

**Conclusion**

Congratulations! You have now successfully:

* Deployed a web application to AWS using AWS Elastic Beanstalk
* Used the eb command line interface to iteratively improve and deploy successive versions of your application
* Learned how to operate your environment completely from the command line
* Updated your environment in two ways, Rolling Deployments and Blue/Green Deployments

**Additional Resources**

* [EB CLI](http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb-cli3.html)
* [AWS Training and Certification](http://aws.amazon.com/training/)

For more information about AWS Training and Certification, see [*http://aws.amazon.com/training/*](http://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).