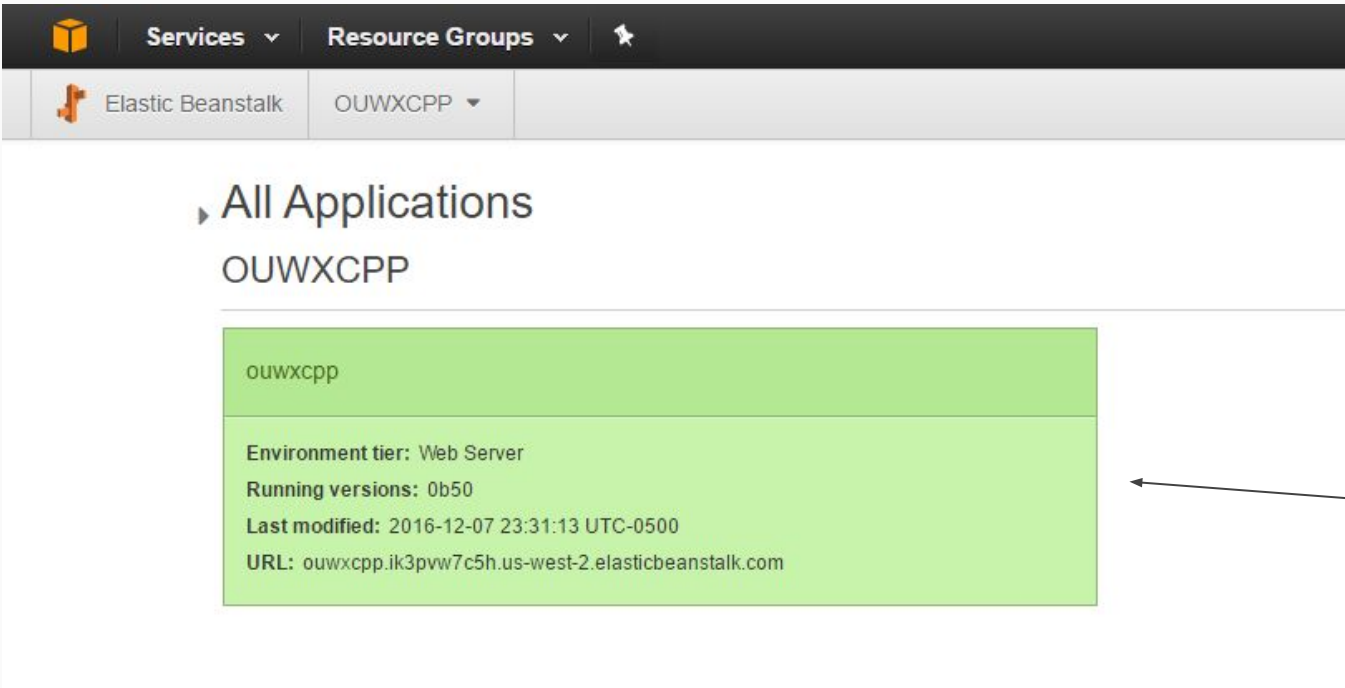


AWS Elastic Beanstalk Command Line Interface

Robert Whitmer



Background on EB - Application/Environment Interface



The screenshot displays the AWS Management Console interface for Elastic Beanstalk. At the top, there is a navigation bar with 'Services' and 'Resource Groups' dropdown menus. Below this, a breadcrumb trail shows 'Elastic Beanstalk' and 'OUWXCPP'. The main content area is titled 'All Applications' and 'OUWXCPP'. A green card represents the application 'ouwxcpp'. The card details include: 'Environment tier: Web Server', 'Running versions: 0b50', 'Last modified: 2016-12-07 23:31:13 UTC-0500', and 'URL: ouwxcpp.ik3pww7c5h.us-west-2.elasticbeanstalk.com'.

ouwxcpp
Environment tier: Web Server
Running versions: 0b50
Last modified: 2016-12-07 23:31:13 UTC-0500
URL: ouwxcpp.ik3pww7c5h.us-west-2.elasticbeanstalk.com

Each application can have multiple environments (dev, test, prod, etc)

EB Web-interface

All Applications > OUWXCPP > ouwxcpp (Environment ID: e-cbxgscp3rv, URL: ouwxcpp.ik3pww7c5h.us-west-2.elasticbeanstalk.com)

Actions ▾

Dashboard

Configuration

Logs

Health

Monitoring

Alarms

Managed Updates **NEW**

Events

Tags

Overview

Refresh



Health

Green

Causes

Running Version

0b50

Upload and Deploy



Configuration

64bit Amazon Linux 2016.03

v2.1.3 running Node.js

Newer version available

Change

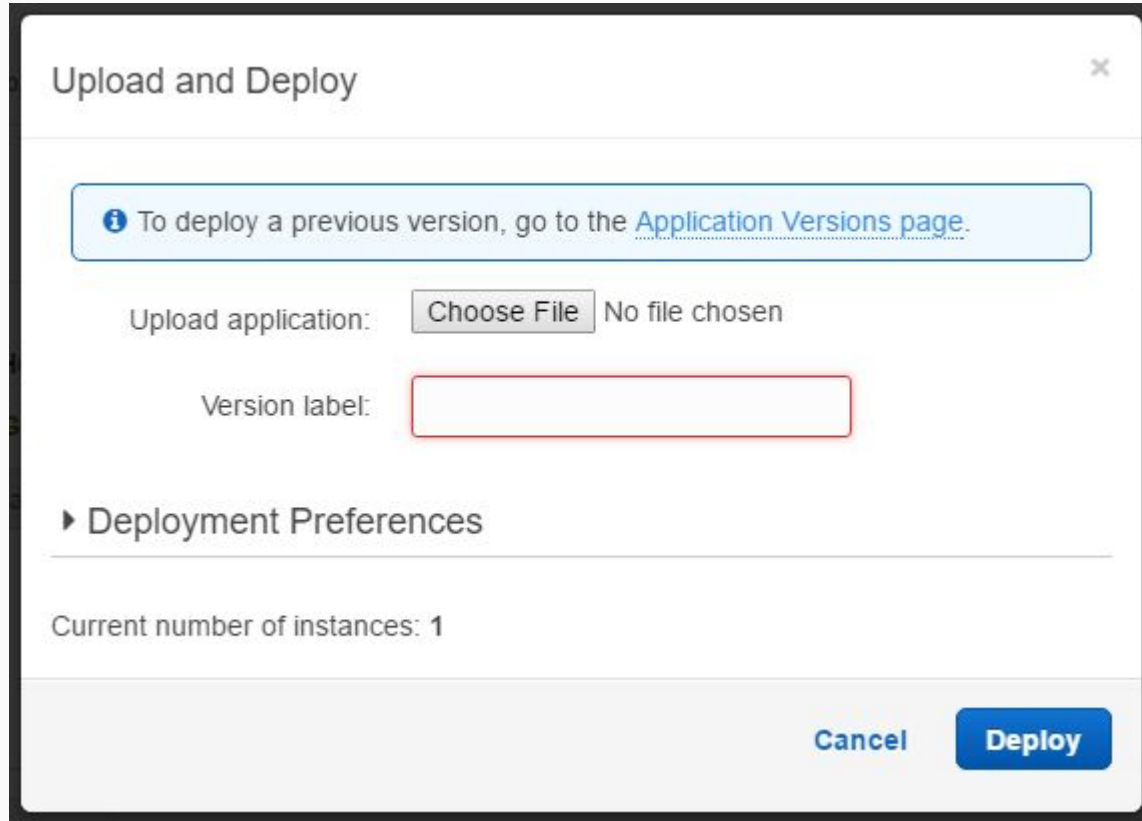
Recent Events

Show All

Time	Type	Details
2016-12-07 23:31:12 UTC-0500	INFO	Environment update completed successfully.
2016-12-07 23:31:12 UTC-0500	INFO	New application version was deployed to running EC2 instances.
2016-12-07 23:28:24 UTC-0500	INFO	Deploying new version to instance(s).
2016-12-07 23:28:18 UTC-0500	INFO	Environment update is starting.
2016-12-07 22:01:25 UTC-0500	INFO	Environment health has transitioned from YELLOW to GREEN

While the web interface has many useful tools as shown above, it lacks in ease of deployment.

Upload and Deploy



The screenshot shows a web interface titled "Upload and Deploy" with a close button (X) in the top right corner. Below the title bar is a light blue informational box containing an information icon and the text: "To deploy a previous version, go to the [Application Versions page](#)." Below this box, the "Upload application:" section features a "Choose File" button and the text "No file chosen". The "Version label:" section has a red-outlined text input field. A section titled "Deployment Preferences" is separated by a horizontal line. Below this line, it says "Current number of instances: 1". At the bottom right, there are two buttons: "Cancel" and "Deploy".

The web interface for the deployment requires your project be in a zipped archive with no top level folder.

Doing this every time you want to deploy can be tedious and cause errors depending on the structure of your project.

AWS Elastic Beanstalk CLI

On the other side, the command line interface allows for

- Rapid deployment without need to worry about archiving project folder
- Scripted utilities for managing the environment based on health or other factors
- Connection with a git repository

Command line options

```
roberto@roberto-VirtualBox: ~/cs4560/dev_deploy/OhioUniversityWXCPeformanceProgram
roberto@roberto-VirtualBox:~/cs4560/dev_deploy/OhioUniversityWXCPeformanceProgram$ clear
roberto@roberto-VirtualBox:~/cs4560/dev_deploy/OhioUniversityWXCPeformanceProgram$ eb -usage: eb (sub-
commands ...) [options ...] {arguments ...}

Welcome to the Elastic Beanstalk Command Line Interface (EB CLI).
For more information on a specific command, type "eb {cmd} --help".

commands:
  abort      Cancels an environment update or deployment.
  clone      Clones an environment.
  config     Modify an environment's configuration. Use subcommands to manage saved configurations.
  console    Opens the environment in the AWS Elastic Beanstalk Management Console.
  create     Creates a new environment.
  deploy     Deploys your source code to the environment.
  events     Gets recent events.
  health     Shows detailed environment health.
  init       Initializes your directory with the EB CLI. Creates the application.
  labs      Extra experimental commands.
  list       Lists all environments.
  local      Runs commands on your local machine.
  logs       Gets recent logs.
  open       Opens the application URL in a browser.
  platform   Manages platforms.
  printenv   Shows the environment variables.
  scale      Changes the number of running instances.
  setenv     Sets environment variables.
  ssh        Opens the SSH client to connect to an instance.
  status     Gets environment information and status.
  swap       Swaps two environment CNAMES with each other.
  terminate  Terminates the environment.
  upgrade    Updates the environment to the most recent platform version.
  use        Sets default environment.

optional arguments:
  -h, --help            show this help message and exit
  --debug               toggle debug output
  --quiet               suppress all output
  -v, --verbose          toggle verbose output
  --profile PROFILE     use a specific profile from your credential file
  -r REGION, --region REGION
                        use a specific region
  --no-verify-ssl       do not verify AWS SSL certificates
  --version             show application/version info

To get started type "eb init". Then type "eb create" and "eb open"
roberto@roberto-VirtualBox:~/cs4560/dev_deploy/OhioUniversityWXCPeformanceProgram$
```

The “eb deploy” command instantly deploys your project in the current directory to the live environment, no archiving necessary.

Connection with Git

```
roberto@roberto-VirtualBox: ~/cs4560/test_presentation
roberto@roberto-VirtualBox:~/cs4560/test_presentation$ ls
roberto@roberto-VirtualBox:~/cs4560/test_presentation$ git init
Initialized empty Git repository in /home/roberto/cs4560/test_presentation/.git/
roberto@roberto-VirtualBox:~/cs4560/test_presentation$ eb init

Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : EU (Ireland)
5) eu-central-1 : EU (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
8) ap-southeast-2 : Asia Pacific (Sydney)
9) ap-northeast-1 : Asia Pacific (Tokyo)
10) ap-northeast-2 : Asia Pacific (Seoul)
11) sa-east-1 : South America (Sao Paulo)
12) cn-north-1 : China (Beijing)
(default is 3):

Select an application to use
1) OUWXCPP
2) [ Create new Application ]
(default is 2): 1
roberto@roberto-VirtualBox:~/cs4560/test_presentation$
```

As long as there was an initialized git repo in the directory before “eb init” was ran, the ebcli will be able to connect with git.

eb deploy & eb health

```
roberto@roberto-VirtualBox:~/cs4560/dev_deploy/OhioUniversityWXCPPerformanceProgram$ eb deploy
Creating application version archive "app-c6ca-170117_174238".
Uploading OUWXCPP/app-c6ca-170117_174238.zip to S3. This may take a while.
Upload Complete.
INFO: Environment update is starting.
INFO: Deploying new version to instance(s).
INFO: New application version was deployed to running EC2 instances.
INFO: Environment update completed successfully.
```

```
ouwxcpp Unknown 2017-01-20 00:04:05
instances: 1 Total, 1 InService, 0 Other
Status: Ready Health Green
```

instance-id	EC2 Health	ELB State	ELB description	health
i-0213dda4559d69cc3	running	InService	N/A	

Clearly, all of the web tools can be matched here, and with more possibilities!

Connecting with Git

Running “eb deploy” will deploy the version of the project that is in your local repository. That is, what has been committed **locally**.

This allows a convenient development feature to be exposed:

You can develop and test locally and deploy changes (potentially to a test/dev environment, not in our case), while still maintaining the integrity of the remote repository so you don't affect others' development. If something breaks on the deployment, you can simply revert the changes. If it works on the live version, you can then push out your changes to the remote repository.

Scripting Possibilities

Because these features are on the command line, they can easily be set up in scripts to automatically do these things. For a simple example, I created a short script that commits my changes locally and deploys to the live environment. If I find that the live version has issues for some reason, another script could be used to revert the environment to the most recent version and then reset my local repository to the previous state.

Script Example

```
#!/usr/bin/env bash
```

```
if [ "$#" -ne 1 ]; then
```

```
    echo "Error: Usage: ./deploy.sh \"COMMIT MESSAGE\""
```

```
fi
```

```
commit=$1
```

```
gcommand="git commit -m \"$commit\""
```

```
echo $gcommand
```

```
git add .
```

```
eval $gcommand
```

```
eb deploy
```

Help Document for Setup

I made the following for our team to help them set up the EB CLI.

Tutorial for setting up EB CLI w/ Git.

- Robert Whitmer

1. This assumes you have some UNIX terminal equivalent (cygwin, virtualbox, mac, nitrous, etc).
 - With that, you'll probably want a package manager like apt-get.
2. Install pip (python installer) to allow easy setup of EB CLI.
 - Depending on your system, `sudo apt-get install python-pip` should work
3. run `'pip install awsebcli'`
4. Clone out the git repo from our Project Github.
5. run `'eb init'`
 - You will need to input your Access Key and Secret Access Key (in the .txt file on slack I sent a while back)
 - Select the region (Oregon is where the main application is, if you pick n. virginia you won't be able to access the live deployment)
 - Select the application (OhioUniversityWXCPPerformanceProgram is the 'live' one)
 - If you run `'eb list'` you should see the environment (only one for now -- dev).
6. You should be able to then make changes, and commit them to the git repo (not necessary to push to be able to deploy, but obviously don't forget to push out if the changes didn't break anything)
7. Once committed, run `'eb deploy'` to deploy the application. If it broke the deployment, you can always `git reset HEAD` to revert back (if you didn't push out already).
8. Bada boom bada bang you should be good to go!

My general plan for development is to do coding/testing locally and then commit to git once tested. Then deploy to the server and see if it actually runs live, if so, great, push the changes to the rest of the team. If it broke things, revert the commit and test more locally. This assumes you're committing from the command line, as it's not so easy to revert on the web interface.