

Module 4: Application Deployment Using Elastic Beanstalk

Demo Document 3

edureka!

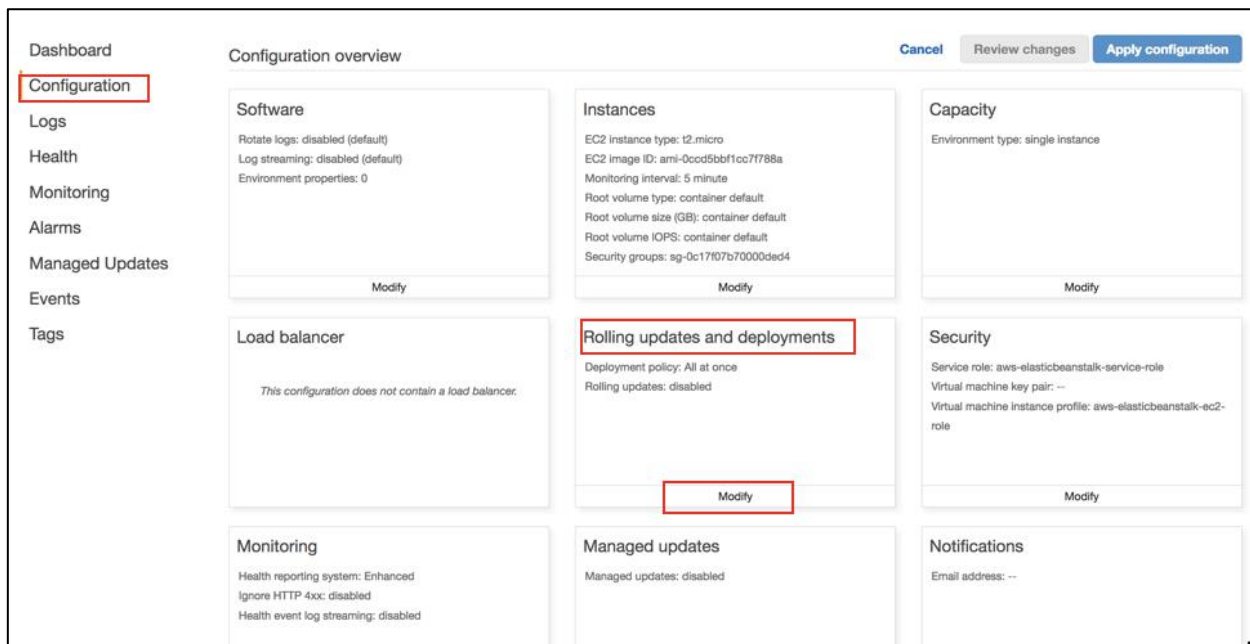
© Brain4ce Education Solutions Pvt. Ltd.

Immutable Deployment On Beanstalk Environment

Demo steps:

Step 1: Configure the Deployment Policy

- Go to your Benstalk application that you have already created in Demo 2
- In the navigation bar, Click on Configuration
- In configuration page, Under Rolling Updates and Deployments, Click on Modify



- In the Deployment policy, select Immutable
- Then, click on 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
immutable

Configuration updates

Changes to virtual machine settings and VPC configuration trigger rolling updates to replace the instances in your environment without downtime. [Learn more](#)

Rolling update type Disabled

Step 2: Create a new application

- In your local system, create a file with the name application.py
- Type the below code and save it local system in the name of **application.py**

```
from flask import Flask

# Print a nice greeting

def say_hello(username = "Guys"):

    return '<html><body background="https://bit.ly/2NuGl9Q" background-  
position=center background-repeat=no-repeat background-size=cover style="padding:  
210px 0; background-color:#000" ><font color="white"><center><h1>Hey  
%s!</h1></font><br><br><br></body>' % username

# Some bits of text for the page

header_text = '''

<html>\n<head> <title> Docker Demo</title> </head>\n<body>'''

instructions = '''

<font color="white"><h2><em> Welcome to Edureka</h2></font>\n'''

home_link = '<p><a href="/">Back</a></p>\n'

footer_text = '</body>\n</html>'

# Elastic Beanstalk looks for an 'application' that is callable by default

application = Flask(__name__)

# Add a rule for the index page

application.add_url_rule('/', 'index', (lambda: header_text +  
    say_hello() + instructions + footer_text))
```

```
# Add a rule when the page is accessed with a name appended to the site

# URL

application.add_url_rule('/<username>', 'hello', (lambda username:
    header_text + say_hello(username) + home_link + footer_text))

# Run the application

if __name__ == "__main__":
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production application.

    application.debug = True

    application.run(host="0.0.0.0")
```

Step 2: Create a Dockerfile

- In your Notepad, type the below code
- And Save it in the name of Dockerfile

```
FROM python:3.6
COPY . /app
WORKDIR /app
RUN pip install Flask==1.0.2
EXPOSE 5000
CMD ["python", "application.py"]
```

Step 3: Create a source bundle and Upload it

- Zip the Dockerfile and the application.py file to form the source bundle
- Then, name it myapp2.zpp
- Upload it

Step 4: Test the Deployment

- Once you upload it you can see the changes in the EC2 Console
- You will notice a new instance will come up with the same configuration

	DockerEnvir...		t2.micro	us-west-1a	 running	 Initializing
	DockerEnvir...	i 	t2.micro	us-west-1b	 running	 2/2 checks ...

- Once the new instance is ready and serving traffic, the older instance will be removed

	DockerEnvir...		t2.micro	us-west-1a	 running	
	DockerEnvir...		t2.micro	us-west-1b	 terminated	

- Thus, you immutable deployment is done through a new set instance in different Auto-Scaling group
- Trace all the changes in the Beanstalk 'Events' section and understand the flow