Module 4: Application Deployment Using Elastic Beanstalk

Demo Document 4

edureka!

Creating cron-job on beanstalk instances using .ebextensions

Demo steps:

Step 1: Create an 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**

```
import logging
import logging.handlers

from wsgiref.simple_server import make_server

# Create logger
logger = logging.getLogger(__name__)
logger.setLevel(logging.INFO)

# Handler
LOG_FILE = '/opt/python/log/sample-app.log'
handler = logging.handlers.RotatingFileHandler(LOG_FILE, maxBytes=1048576, backupCount=5)
handler.setLevel(logging.INFO)

# Formatter
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')

# Add Formatter to Handler
handler.setFormatter(formatter)

# add Handler to Logger
logger.addHandler(handler)

welcome = """
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <!--Copyright 2012 Amazon.com, Inc. or its affiliates. All Rights Reserved.

    Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file
except in compliance with the License. A copy of the License is located at
http://aws.Amazon/apache2.0/

    or in the "license" file accompanying this file. This file is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under the License.
  -->
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
 <title>Welcome</title>
 <style>
 body {color: #ffffff;
  background-color: #E0E0E0;
  font-family: Arial, sans-serif;
  font-size:14px;
  -moz-transition-property: text-shadow;
  -moz-transition-duration: 4s;
  -webkit-transition-property: text-shadow;
  -webkit-transition-duration: 4s;
  text-shadow: none; }
 body.blurry {
  -moz-transition-property: text-shadow;
  -moz-transition-duration: 4s;
  -webkit-transition-property: text-shadow;
  -webkit-transition-duration: 4s;
  text-shadow: #fff 0px 0px 25px;
 }
 a {color: #0188cc;}
 .textColumn, .linksColumn {
  padding: 2em;
 }
 .textColumn {
  position: absolute;
  top: 0px;
  right: 50%;
  bottom: 0px;
  left: 0px;

  text-align: right;
  padding-top: 11em;
  background-color: #1BA86D;
  background-image: -moz-radial-gradient(left top, circle, #6AF9BD 0%, #00B386 60%);
  background-image: -webkit-gradient(radial, 0 0, 1, 0 0, 500, from(#6AF9BD), to(#00B386)); }

.textColumn p {
  width: 75%;
  float:right;
 }
 .linksColumn {
  position: absolute;
  top:0px;
  right: 0px;
  bottom: 0px;
  left: 50%;
  background-color: #E0E0E0;}
```

```
 h1 {font-size: 500%;
   font-weight: normal;
   margin-bottom: 0em;}
 h2 { font-size: 200%;
   font-weight: normal;
   margin-bottom: 0em; }
 ul {padding-left: 1em;
   margin: 0px; }
 li {margin: 1em 0em;}
 </style></head>
<body id="sample">
 <div class="textColumn">
   <h1>Congratulations</h1>
   <p>Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud</p>
 </div>
<div class="linksColumn">
   <h2>What's Next?</h2>
<ul>
   <li><a href="http://docs.amazonwebservices.com/elasticbeanstalk/latest/dg/">AWS Elastic Beanstalk overview</a></li>
   <li><a
href="http://docs.amazonwebservices.com/elasticbeanstalk/latest/dg/index.html?concepts.html">
AWS Elastic Beanstalk concepts</a></li>
<li><a
href="http://docs.amazonwebservices.com/elasticbeanstalk/latest/dg/create_deploy_Python_django.html">Deploy a Django Application to AWS Elastic Beanstalk</a></li>
   <li><a
href="http://docs.amazonwebservices.com/elasticbeanstalk/latest/dg/create_deploy_Python_flask.html">Deploy a Flask Application to AWS Elastic Beanstalk</a></li>
   <li><a
href="http://docs.amazonwebservices.com/elasticbeanstalk/latest/dg/create_deploy_Python_custom_container.html">Customizing and Configuring a Python Container</a></li>
   <li><a href="http://docs.amazonwebservices.com/elasticbeanstalk/latest/dg/using-features.loggingS3.title.html">Working with Logs</a></li>
```

```
</ul>
 </div>
</body>
</html>
"""

def application(environ, start_response):
   path    = environ['PATH_INFO']
   method  = environ['REQUEST_METHOD']
   if method == 'POST':
      try:
         if path == '/':
            request_body_size = int(environ['CONTENT_LENGTH'])
            request_body = environ['wsgi.input'].read(request_body_size).decode()
            logger.info("Received message: %s" % request_body)
         elif path == '/scheduled':
            logger.info("Received task %s scheduled at %s",
environ['HTTP_X_AWS_SQSD_TASKNAME'], environ['HTTP_X_AWS_SQSD_SCHEDULED_AT'])
      except (TypeError, ValueError):
         logger.warning('Error retrieving request body for async work.')
      response = ''
   else:
      response = welcome
   status = '200 OK'
   headers = [('Content-type', 'text/html')]

   start_response(status, headers)
   return [response]


if __name__ == '__main__':
   httpd = make_server('', 8000, application)
   print("Serving on port 8000...")
   httpd.serve_forever()
```

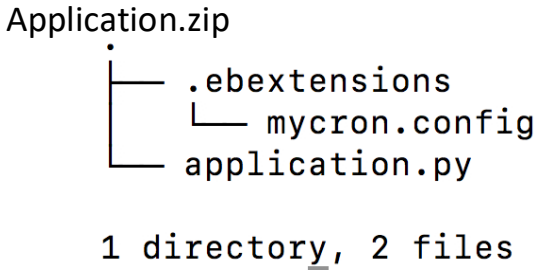
Step 2: Create a Configuration File

- Create a folder named .ebextensions
- Inside that folder create this mycorn.config file

```
files:
    "/etc/cron.d/mycron":
        mode: "000644"
        owner: root
        group: root
        content: |
            * * * * * root /usr/local/bin/myscript.sh

    "/usr/local/bin/myscript.sh":
        mode: "000755"
        owner: root
        group: root
        content: |
            #!/bin/bash

            date > /tmp/date
            # Your actual script content

            exit 0

commands:
    remove_old_cron:
        command: "rm -f /etc/cron.d/mycron.bak"
```

Step 3: Create a Source Bundle

- Zip the folder in the following structure



Step 4: Create a Beanstalk application

- Create a python beanstalk application from the console
- Upload our zipped application bundle (with ebextensions)

- Make sure you add a keypair to the instance in the environment configuration, as we would need to SSH into the instance and check the cron

Step 5: Test the working

- SSH into the beanstalk Instance and check if the cron is working by moving to the tmp directory and typing

cat date

```
[ec2-user@ip-172-31-20-37 tmp]$ pwd
/tmp
[ec2-user@ip-172-31-20-37 tmp]$ cat date
Thu Jan 10 13:31:01 UTC 2019
Thu Jan 10 13:32:01 UTC 2019
Thu Jan 10 13:33:01 UTC 2019
Thu Jan 10 13:34:01 UTC 2019
Thu Jan 10 13:35:01 UTC 2019
Thu Jan 10 13:36:01 UTC 2019
Thu Jan 10 13:37:01 UTC 2019
Thu Jan 10 13:38:01 UTC 2019
Thu Jan 10 13:39:01 UTC 2019
Thu Jan 10 13:40:01 UTC 2019
Thu Jan 10 13:41:01 UTC 2019
Thu Jan 10 13:42:01 UTC 2019
Thu Jan 10 13:43:01 UTC 2019
Thu Jan 10 13:44:01 UTC 2019
Thu Jan 10 13:45:01 UTC 2019
Thu Jan 10 13:46:01 UTC 2019
Thu Jan 10 13:47:01 UTC 2019
[ec2-user@ip-172-31-20-37 tmp]$
```

- If not check if your Cron file is created

```
[ec2-user@ip-172-31-20-37 cron.d]$ pwd
/etc/cron.d
[ec2-user@ip-172-31-20-37 cron.d]$ cat mycron
* * * * * root /usr/local/bin/myscript.sh
[ec2-user@ip-172-31-20-37 cron.d]$
```