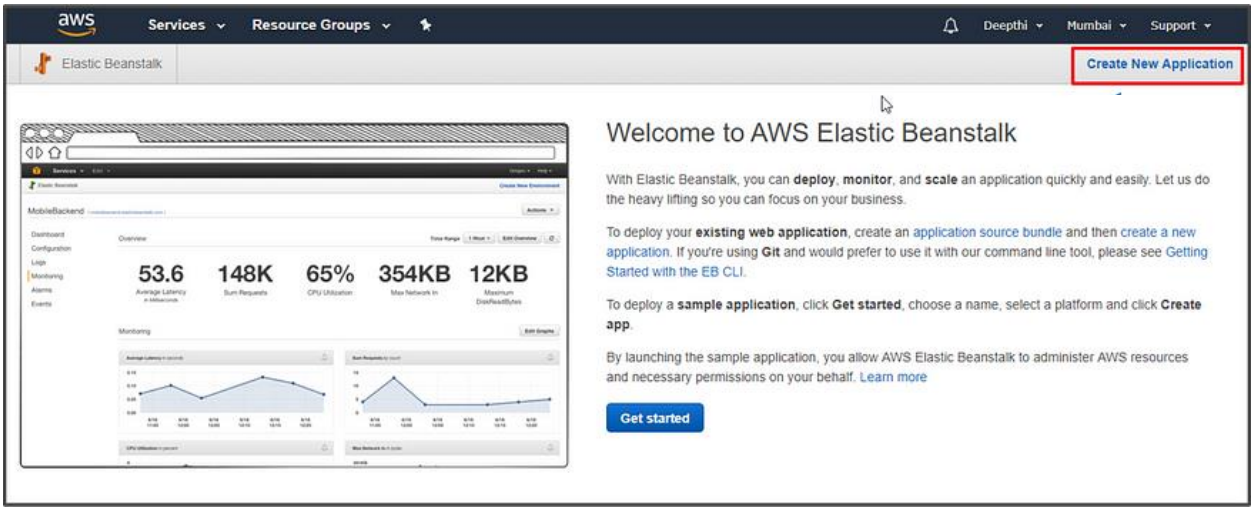
CONTENTS

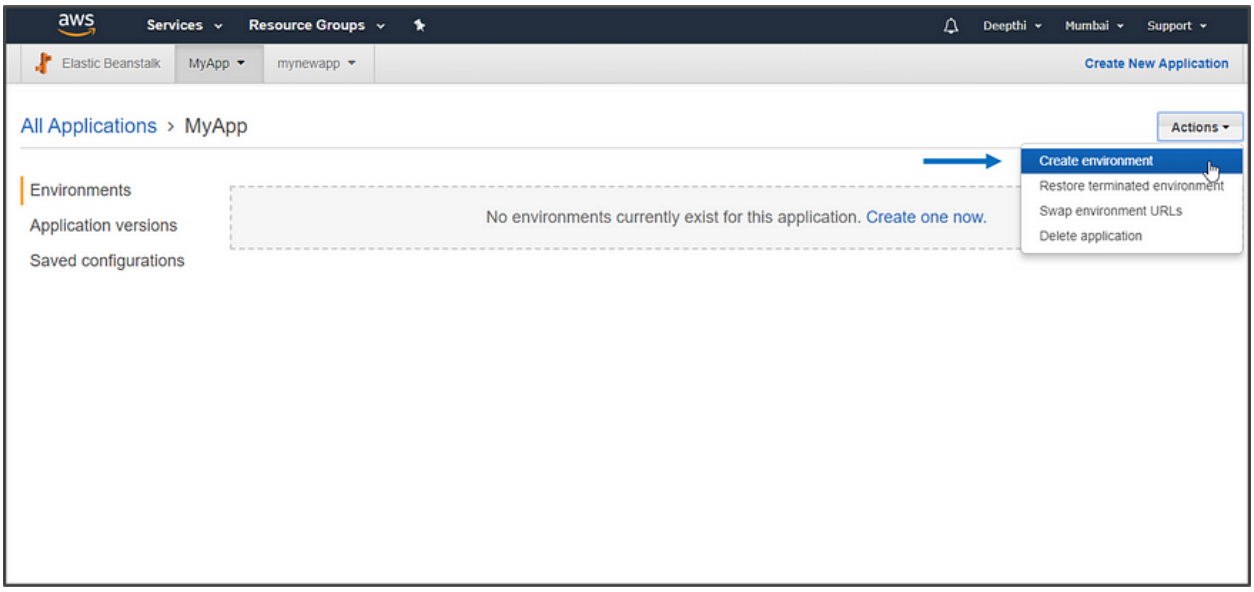
Deploy an Application on Elastic Beanstalk

Deploying an application on Elastic Beanstalk is a fairly simple process. Let's see how to deploy an application stepwise.

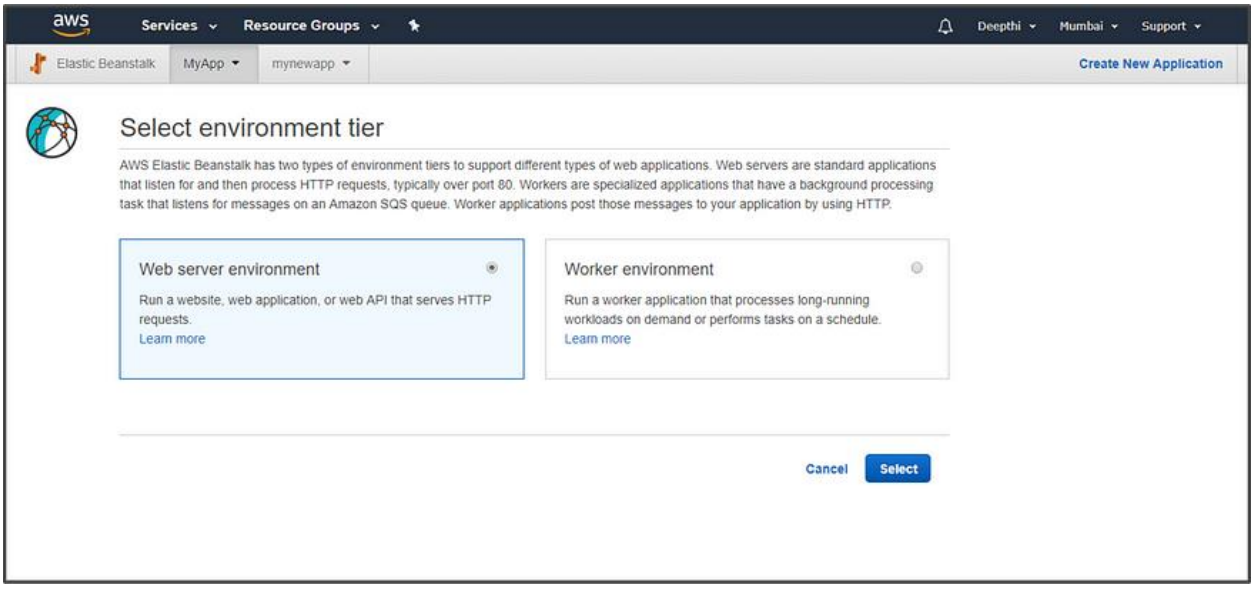
Step 1: On the Elastic Beanstalk console click on Create New Application option. A dialog box appears where you can give a name and appropriate description for your application.



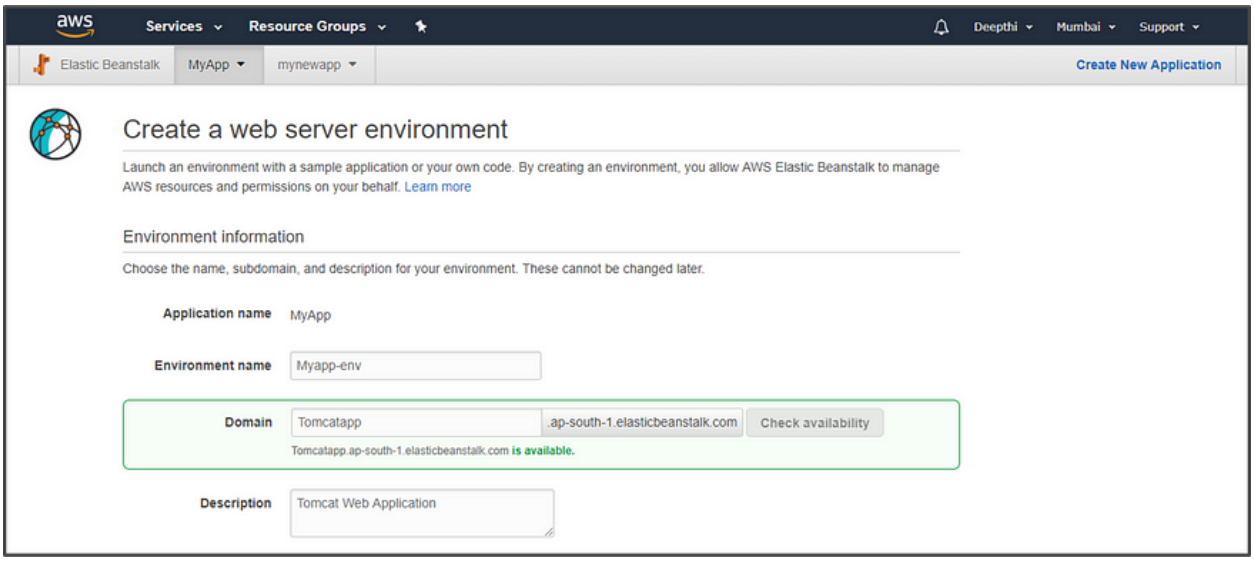
Step 2: Now that the application folder has been created, you can click on the Actions tab and select Create Environment option. Beanstalk provides you with an option where you can create multiple Environments for your application.



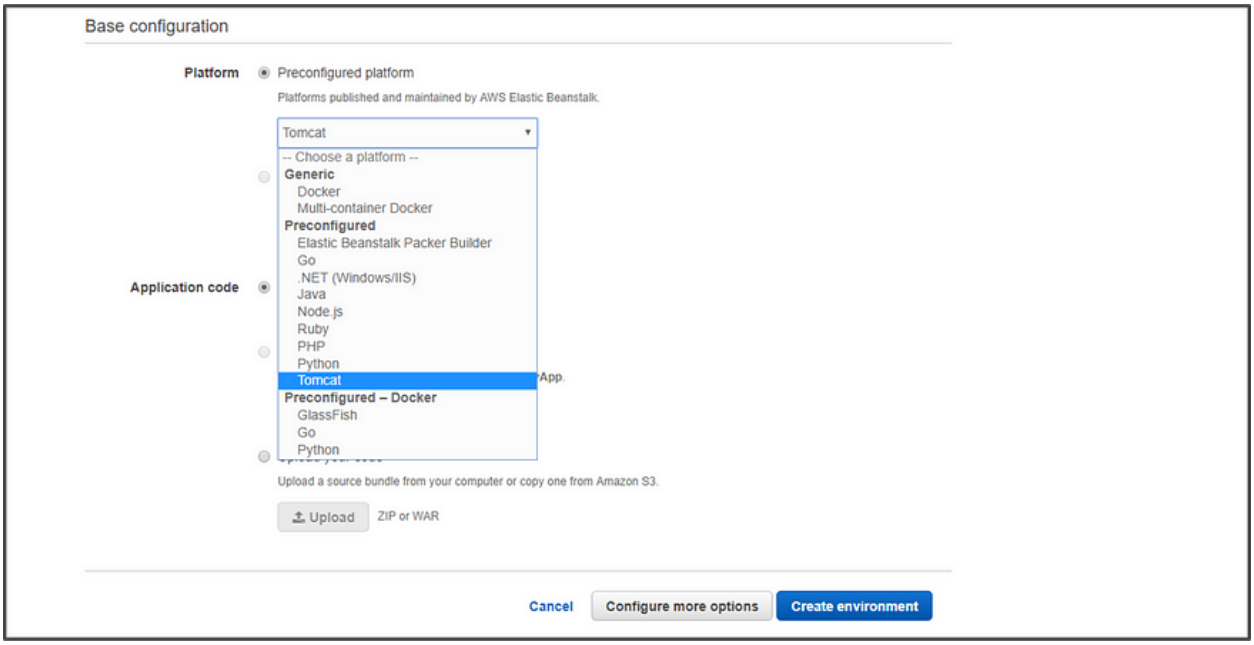
Step 3: Choose among two different Environment Tier options. Choose Web Server Environment if you want your application to handle HTTP requests or choose Worker Environment to handle background tasks.



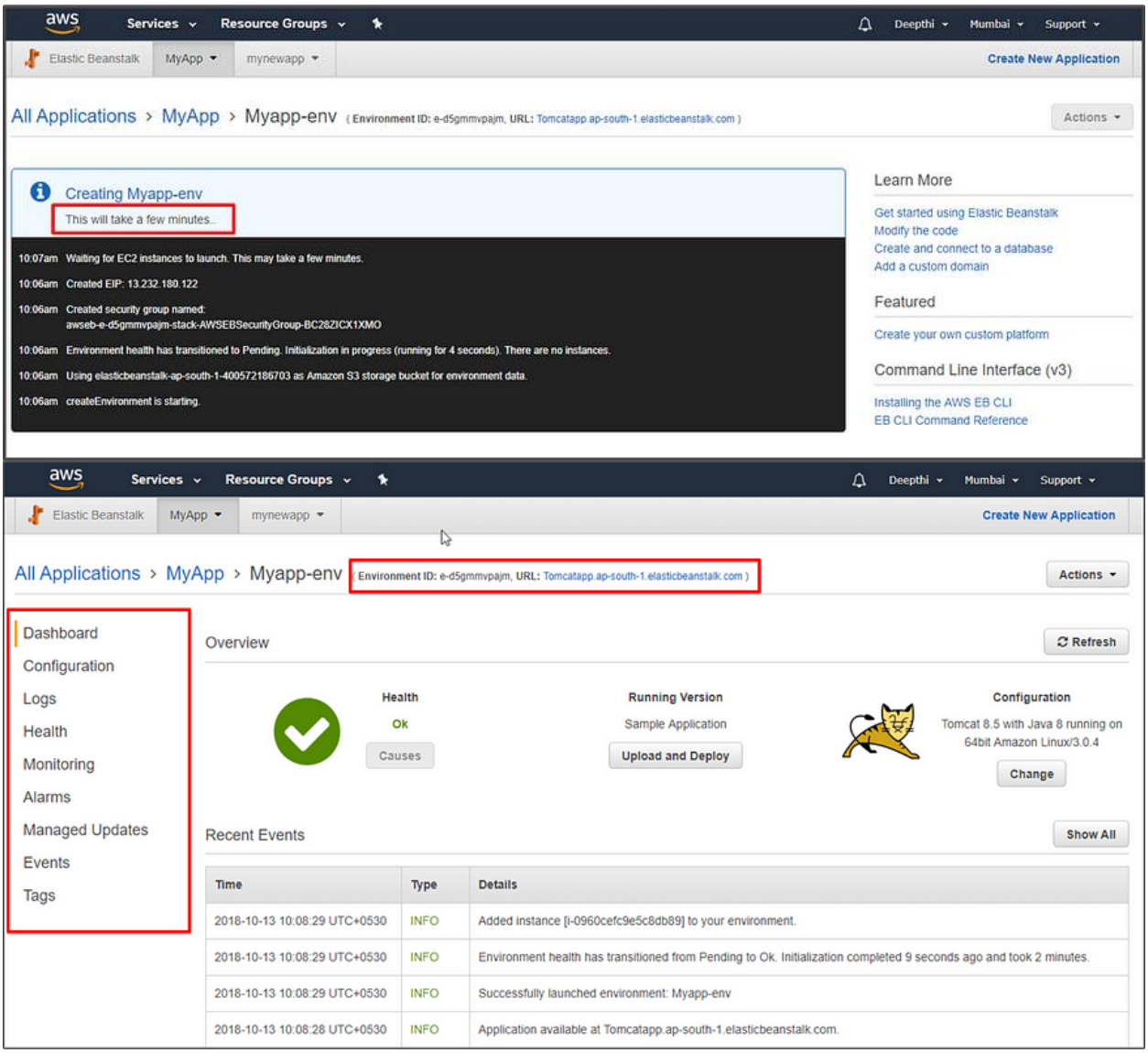
Step 4: Another dialog appears, where you need to provide a domain name and description for your application.



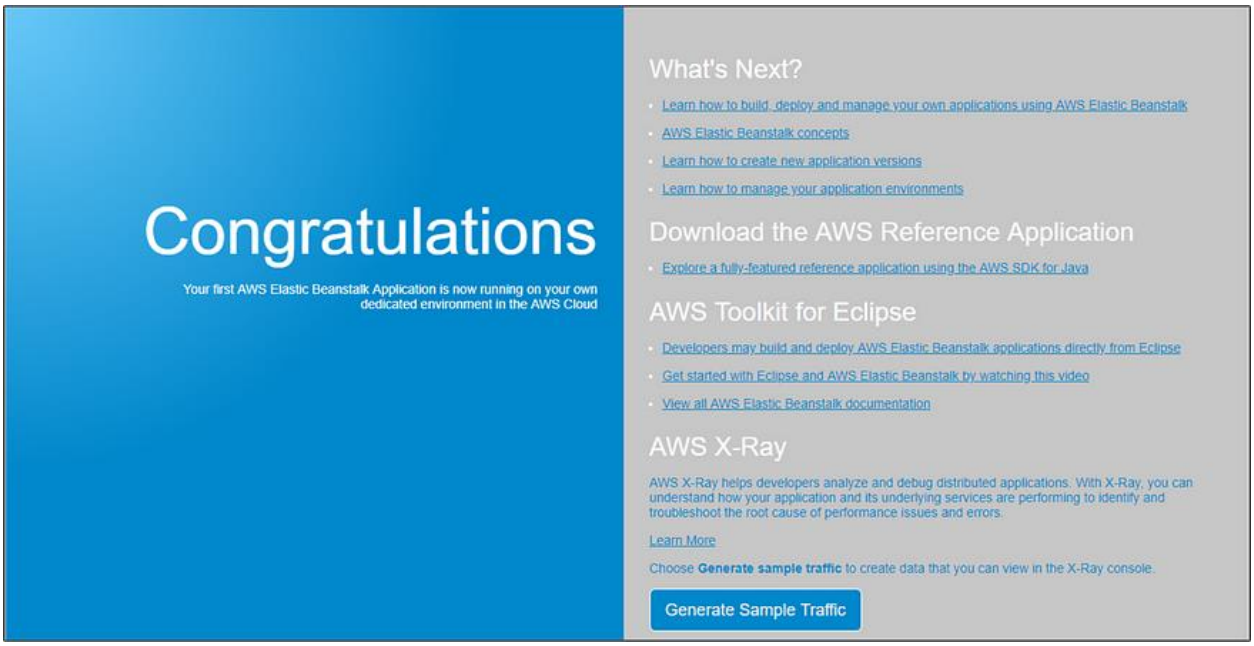
Step 5: Choose a platform of your choice for your application. Elastic Beanstalk will provide you with multiple options. You can choose a sample application provided by Beanstalk, or upload a file which has code for your application.



Beanstalk will take a few minutes to launch an Environment. Once the Environment is launched, on the navigation pane you can see multiple options where you can change the configuration of your application, view log files, and events. Since you're already on Environment page, try exploring different features that Beanstalk offers.



Step 6: On the top right corner, you will find the URL of your application version. Click on that URL. You will be taken to a page which will confirm that you have successfully launched your application on Elastic Beanstalk.



I hope now you have a clear picture of Elastic Beanstalk and how you can use Beanstalk to deploy your applications.

Python Optional Open Source Codes

Here's a basic example using the Flask web framework and the Serverless Framework to deploy the application to AWS Lambda:

1. Install Flask and Serverless Framework:

   pip install Flask serverless

2. Create a Python script (app.py) for your Flask web application:

```
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/')
def index():
    return jsonify({"message": "Hello, this is your serverless Flask app on AWS Lambda!"})

if __name__ == '__main__':
    app.run(debug=True)
```

3. Create a serverless.yml file for your Serverless Framework configuration:

```
service: my-flask-app

provider:
  name: aws
  runtime: python3.8

functions:
  app:
    handler: app.index
    events:
     - http:
         path: /
         method: ANY
```

4. Deploy your application using Serverless:

```
sls deploy
```

This will package your Flask application and deploy it to AWS Lambda. After deployment, Serverless will provide you with an API Gateway URL. Visit the provided URL in your web browser or use a tool like curl or Postman to test the endpoint.