

# Experiment 08 - Cloud Computing

February 01, 2023

## Aim

To deploy an application in Elastic Beanstalk.

## Theory

- **Beanstalk** : AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service, CloudWatch, auto-scaling, and Elastic Load Balancers.
- **Environment** : The environment is a collection of AWS resources running an application version. One can deploy multiple environments when one needs to run multiple versions of an application. For example, one might have development, integration, and production environments.
- **Monitoring** : When one runs a production website, it is important to know that application is available and responding to requests. To assist with monitoring application's responsiveness, Elastic Beanstalk provides features that monitor statistics about application and create alerts that trigger when thresholds are exceeded.
- **Permission** : Elastic Beanstalk requires permissions not only for its own API actions, but also for several other AWS services. Elastic Beanstalk uses user permissions to launch resources in an environment. These resources include EC2 instances, an Elastic Load Balancing load balancer, and an Auto Scaling group. Elastic Beanstalk also uses user permissions to save logs and templates to Amazon Simple Storage Service (Amazon S3), send notifications to Amazon SNS, assign instance profiles, and publish metrics to CloudWatch. Elastic Beanstalk requires AWS CloudFormation permissions to orchestrate resource deployments and updates. It also requires Amazon RDS permissions to create databases when needed, and Amazon SQS permissions to create queues for worker environments.

## Results

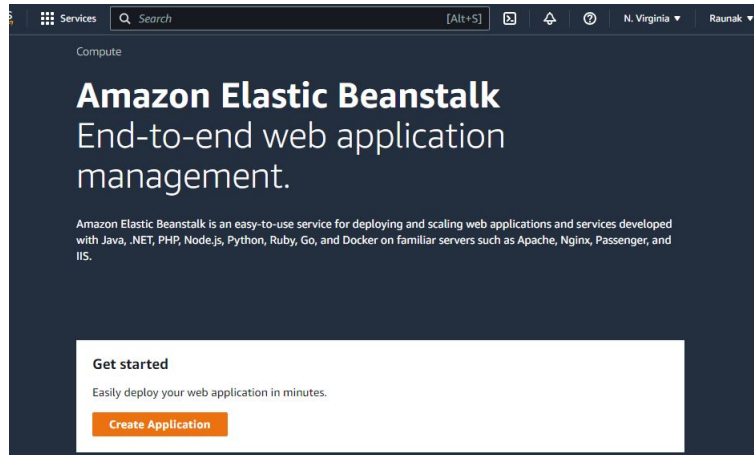


Figure 1: Start by searching for ElasticBeanStalk service in the search bar and select create application

**Create a web app**

Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. [Learn more](#)

Application information

Application name

Up to 100 Unicode characters, not including forward slash (/).

Figure 2: Arbitrary name is essential

### Platform

Platform

Python

Platform branch

Python 3.8 running on 64bit Amazon Linux 2

Platform version

3.4.4 (Recommended)

### Application code

☒ Sample application  
Get started right away with sample code.

☐ Upload your code  
Upload a source bundle from your computer or copy one from Amazon S3.

Cancel
Configure more options
Create application

Figure 3: Start with the sample application and platform of user preference. I personally have selected the Python

### Genericapp-env

Genericapp-env.eba-95eiwz35.us-east-1.elasticbeanstalk.com
(e-e-h5yx6pjpi)

Application name: genericApp

Refresh
Actions

#### Health

Ok

Causes

#### Running version

Sample Application

Upload and deploy

#### Platform

Python 3.8 running on 64bit Amazon Linux 2/3.4.4

Change

Figure 4: Successful creation of the sample application

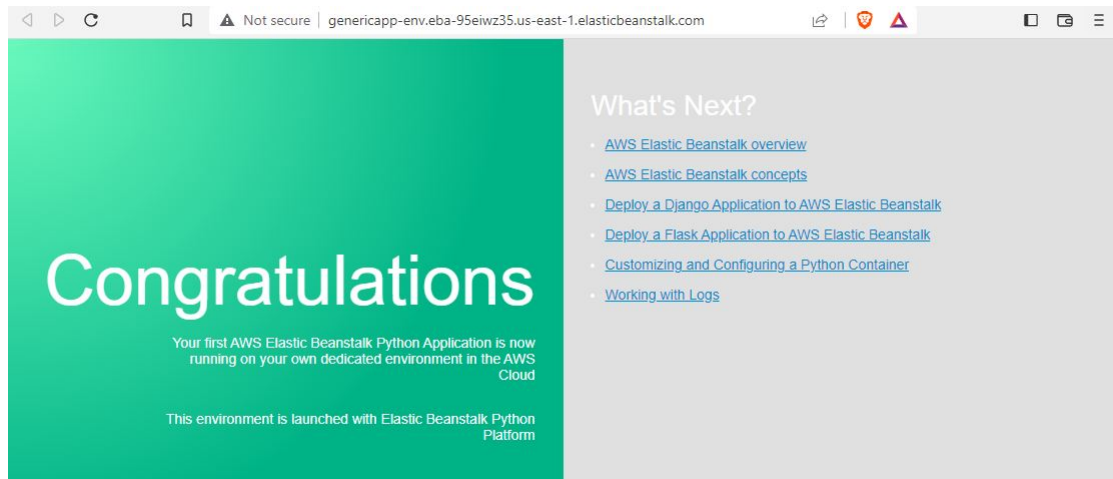


Figure 5: Clicking on the link redirects to the deployed application

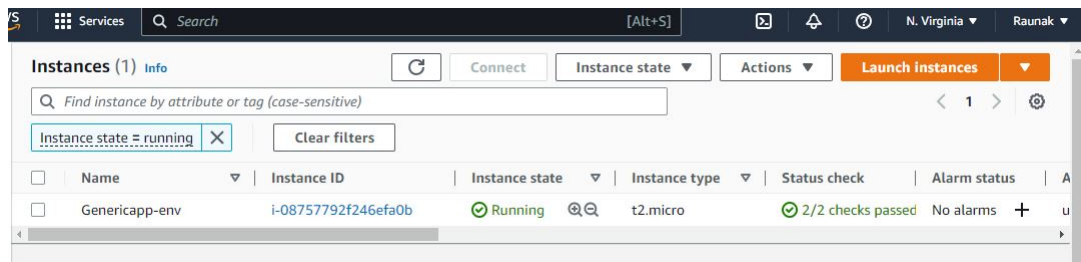


Figure 6: The instance on which the entire system runs can be also viewed in the instances panel

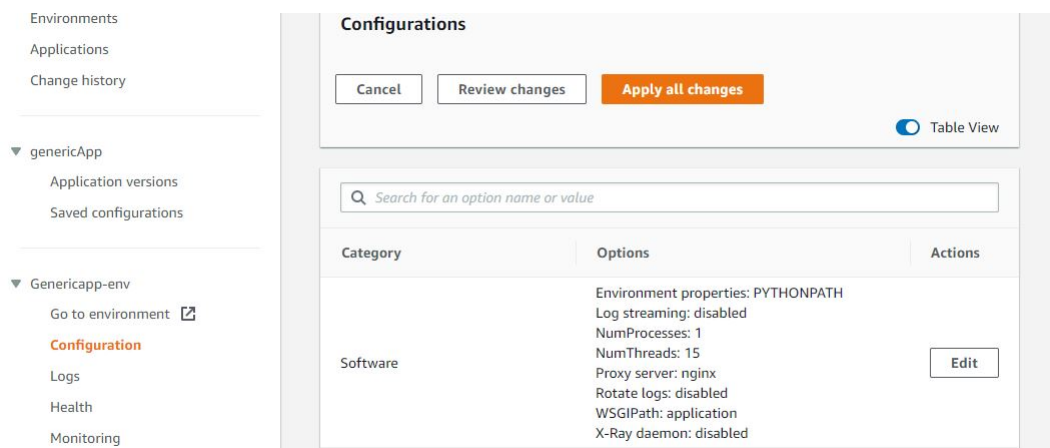


Figure 7: One can also check for all the configurations respectively

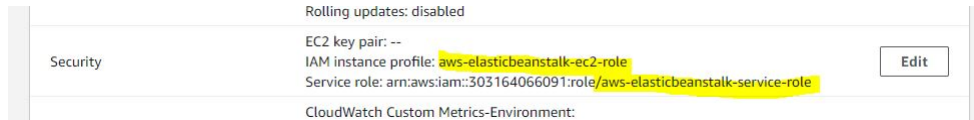


Figure 8: The security group can be also viewed

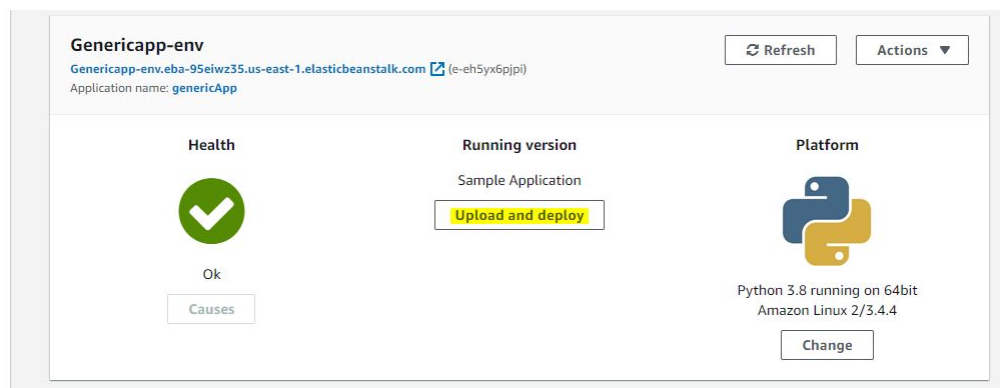


Figure 9: Updating the application can be deployed using the highlighted option

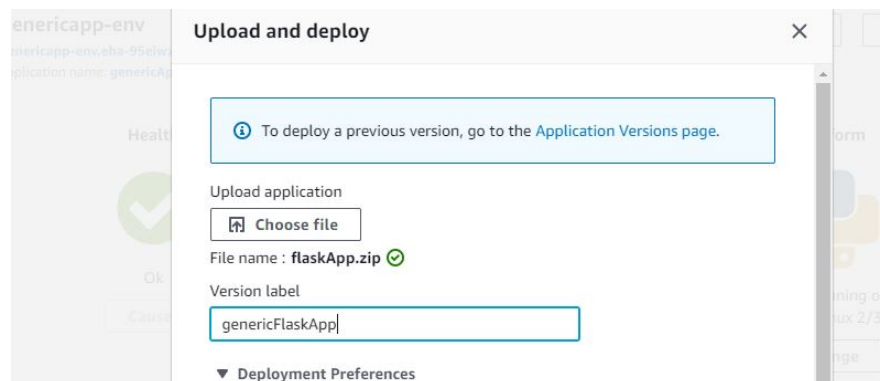


Figure 10: Uploading procedure is pretty standard and arbitrarily name can also be given



Figure 11: Checking the application after successful execution of all the required process

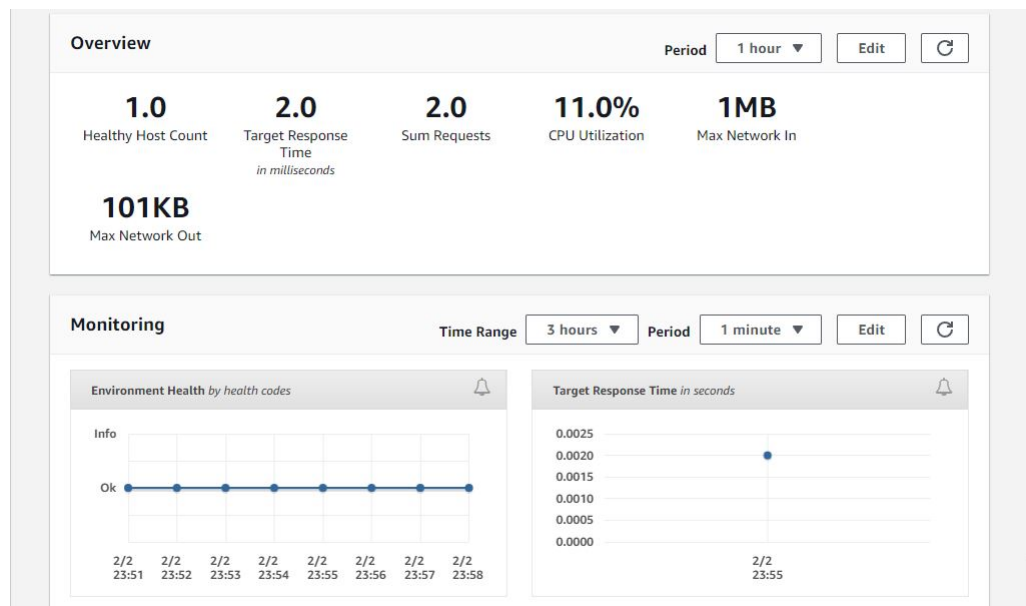


Figure 12: Monitoring section also highlights some insights about the uploaded application

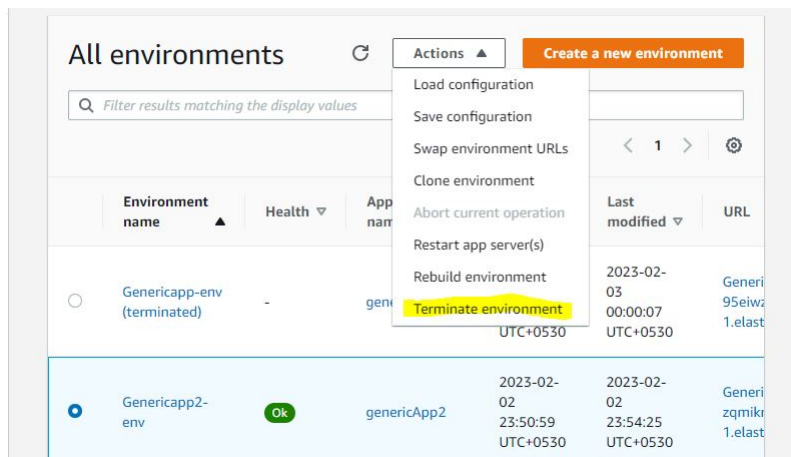


Figure 13: Terminating the deployed application by selecting the option like in respective application

## Conclusion

This experiment is successful demonstration of the AWS Elastic Bean Stalk service that considers the content from the AWS S3 bucket and uses the infrastructure using the EC2 Instance. The Beanstalk basically acts as the server that executes the following application and creates an environment oriented hosting application. The execution of the all the process has been successfully shown in the results section of this experiment respectively.