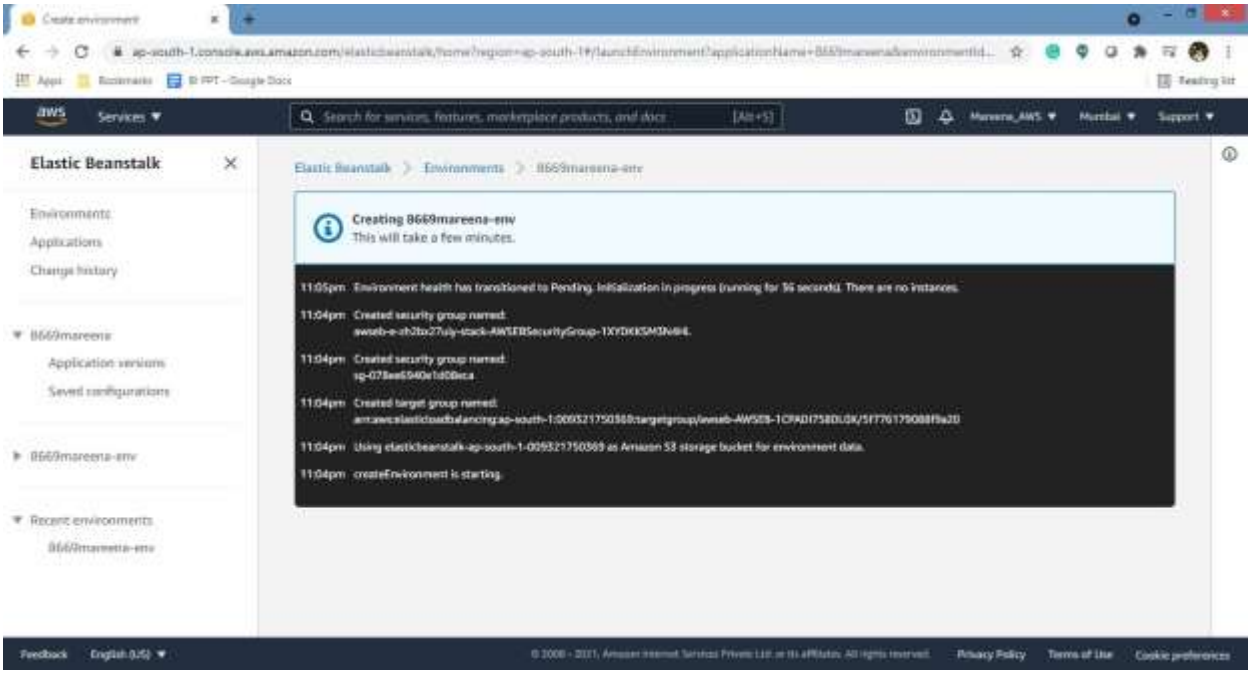
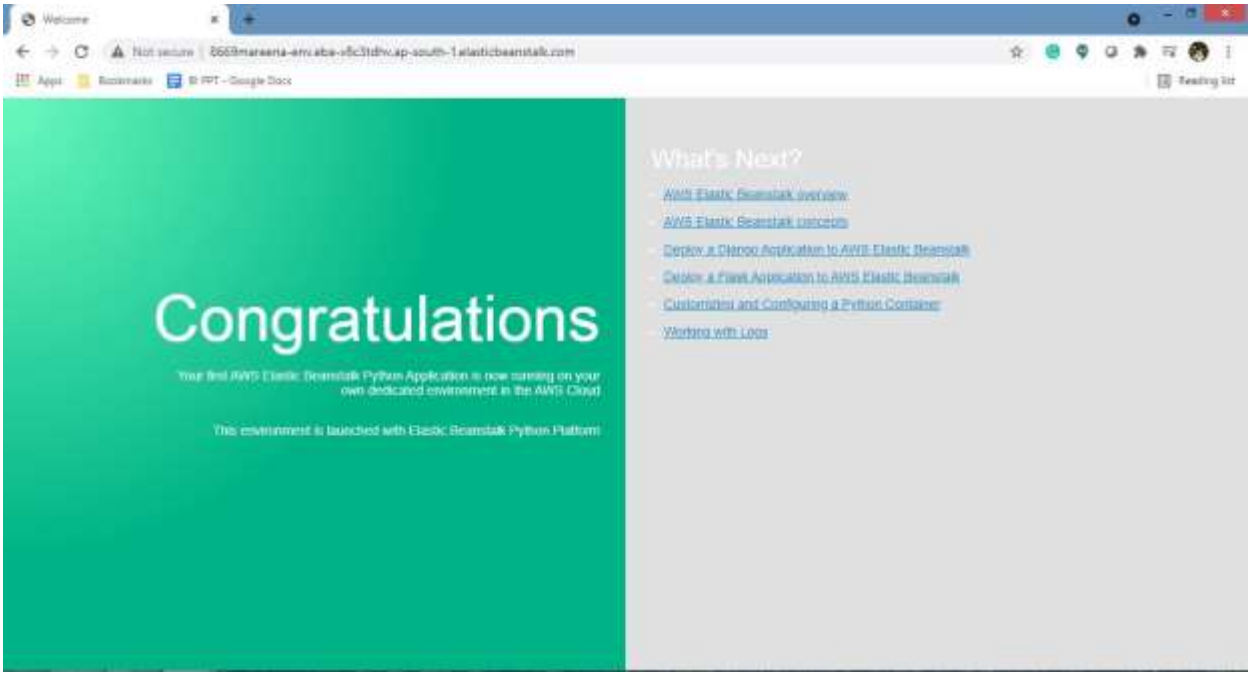
Name of candidate:	Mareena Mark Fernandes	
Roll no.: 8669	**Year: TE**	**Semester:-VI**
Branch: IT	**Subject: Cloud Service Design Lab**	
Experiment No.: 8	**Date of performance:**	**Date of submission:**
CO's Covered: LO4		

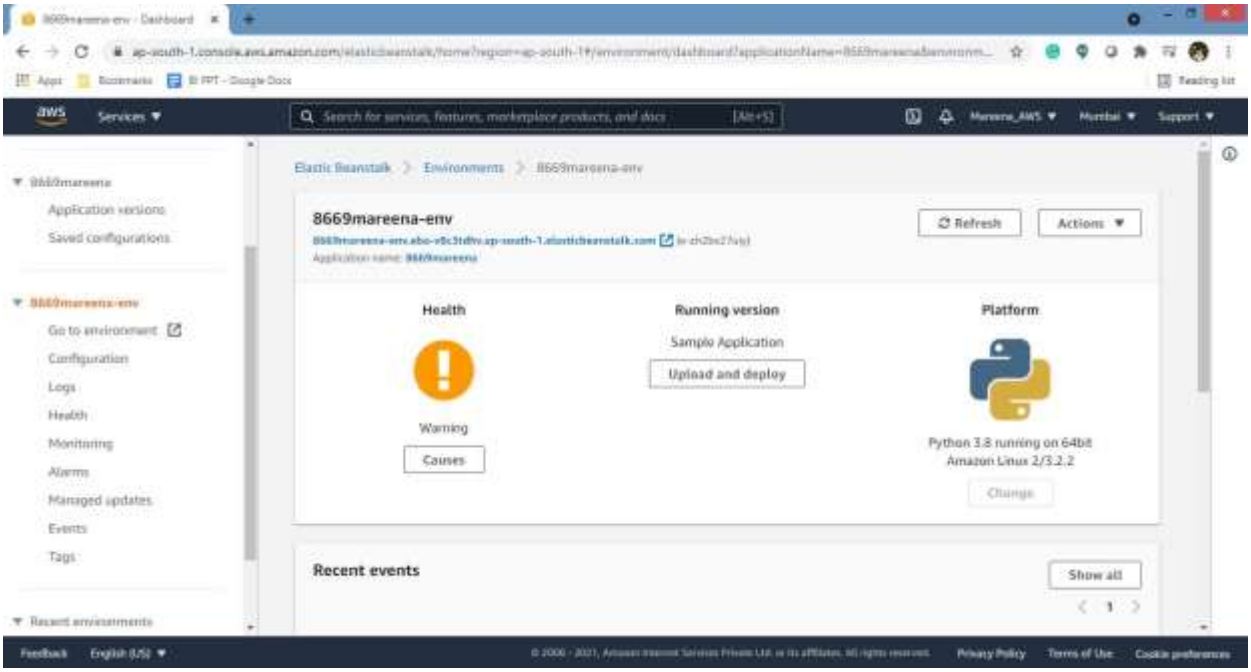
Rubrics for Practical

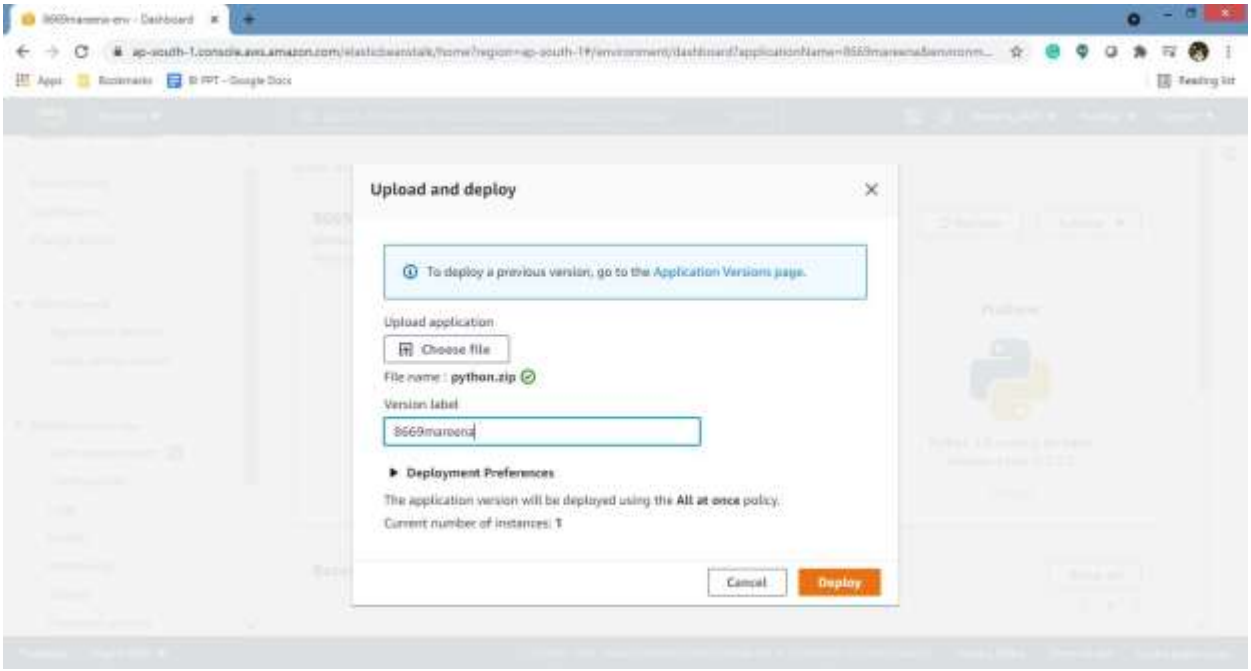
Indicator	**Poor**	**Average**	**Good**	**Excellent**
Timeline (3)	More than two weeks late (0)	Two weeks late (1)	One week late (2)	Early or on time (3)
Knowledge (3)	Not Able to answer any Question (0)	Able to answer a Question (1)	Able to answer few Questions (2)	Able to answer all questions (3)
Performance (4)	Able to partially perform the experiment (1)	Able to perform the experiment for certain extent (2)	Able to perform the experiment with support (3)	Able to perform the experiment considering all aspects (4)
Rubrics	**Timeline(3)**	**Knowledge(3)**	**Performance(4)**	**Total (10)**
Score				

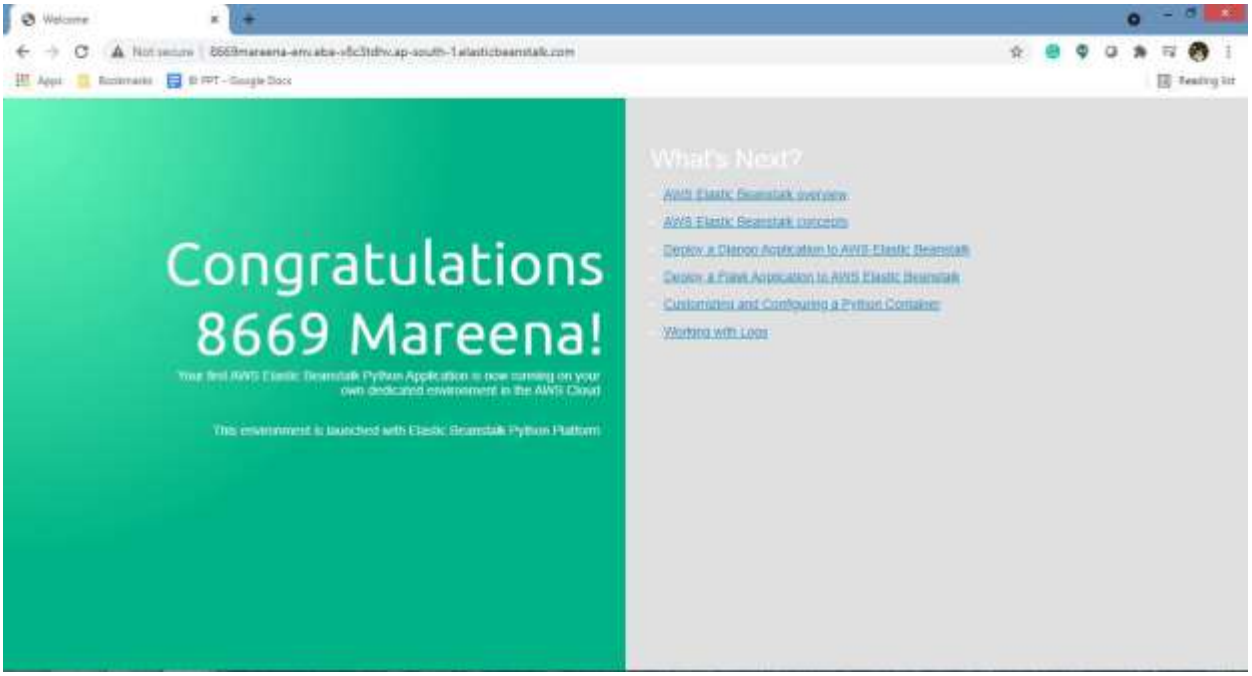
Signature of faculty:


Elastic Beanstalk
Environments
Applications
Change history
8669mareena
Application versions
Saved configurations
8669mareena-env
Recent environments
8669mareena-env
Elastic Beanstalk > Environments > 8669mareena-env
Creating 8669mareena-env
This will take a few minutes.
11:05pm Environment health has transitioned to Pending. Initialization in progress (running for 56 seconds). There are no instances.
11:04pm Created security group named:
11:04pm Created security group named:
11:04pm Created target group named:
11:04pm Using elasticbeanstalk-ap-south-1-009521750369 as Amazon S3 storage bucket for environment data.
11:04pm createEnvironment is starting.


Congratulations
Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud
This environment is launched with Elastic Beanstalk Python Platform
What's Next?
AWS Elastic Beanstalk overview
AWS Elastic Beanstalk concepts
Deploy a Django Application to AWS Elastic Beanstalk
Deploy a Flask Application to AWS Elastic Beanstalk
Customizing and Configuring a Python Container
Working with Logs


Elastic Beanstalk
Environments
8669mareena-env
8669mareena-env
Refresh
Actions
Health
Warning
Causes
Running version
Sample Application
Upload and deploy
Platform
Python 3.8 running on 64bit
Amazon Linux 2/3.2.2
Change
Recent events
Show all
Application versions
Saved configurations
Go to environment
Configuration
Logs
Health
Monitoring
Alarms
Managed updates
Events
Tags


Upload and deploy
To deploy a previous version, go to the Application Versions page.
Upload application
Choose file
File name : python.zip
Version label
8669mareena
Deployment Preferences
The application version will be deployed using the All at once policy.
Current number of instances: 1
Cancel
Deploy



Conclusion:

1. AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
2. You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring.
3. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

Post labs:

- Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define.
- You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
- You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.
- As one of AWS's foundational services, Amazon VPC makes it easy to customize your VPC's network configuration.
- You can create a public-facing subnet for your web servers that have access to the internet. It also lets you place your backend systems, such as databases or application servers, in a private-facing subnet with no internet access.
- Amazon VPC lets you to use multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

  Benefits:

  1. Secure and monitored network connections
  2. Simple set-up and use
  3. Customizable virtual network

  User Cases:

  1. Host a simple, public-facing website
  2. Host multi-tier web applications
  3. Back up and recover your data after a disaster