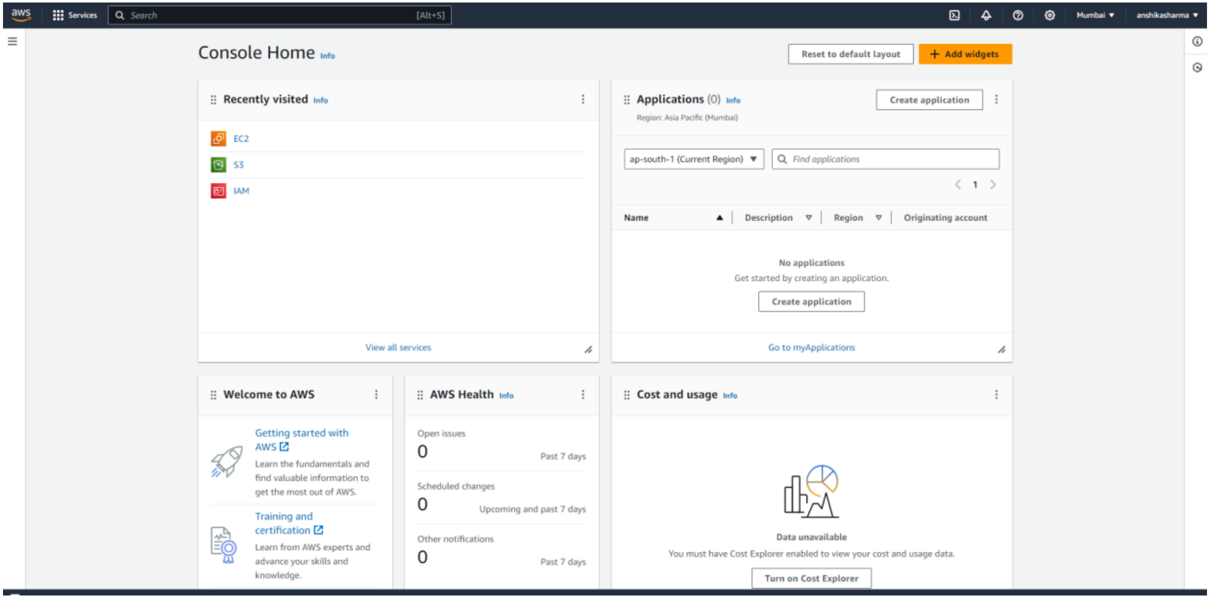
Practical-5 To configure Elastic Beanstalk

Name: Anshika Sharma

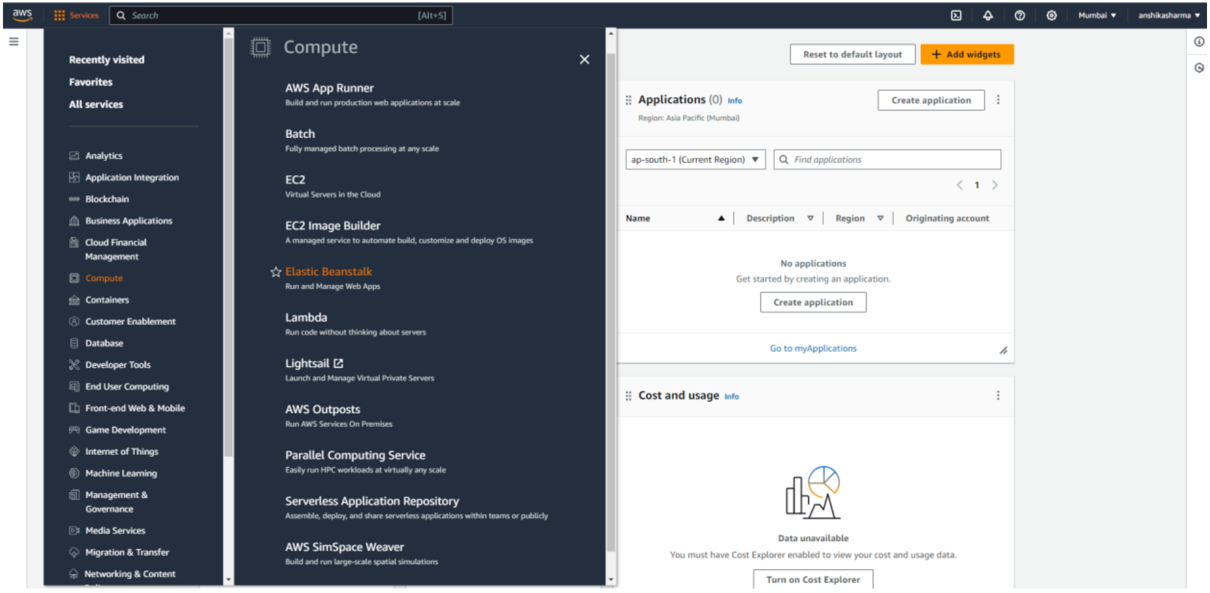
SAP ID: 86062300034

Roll No: A061

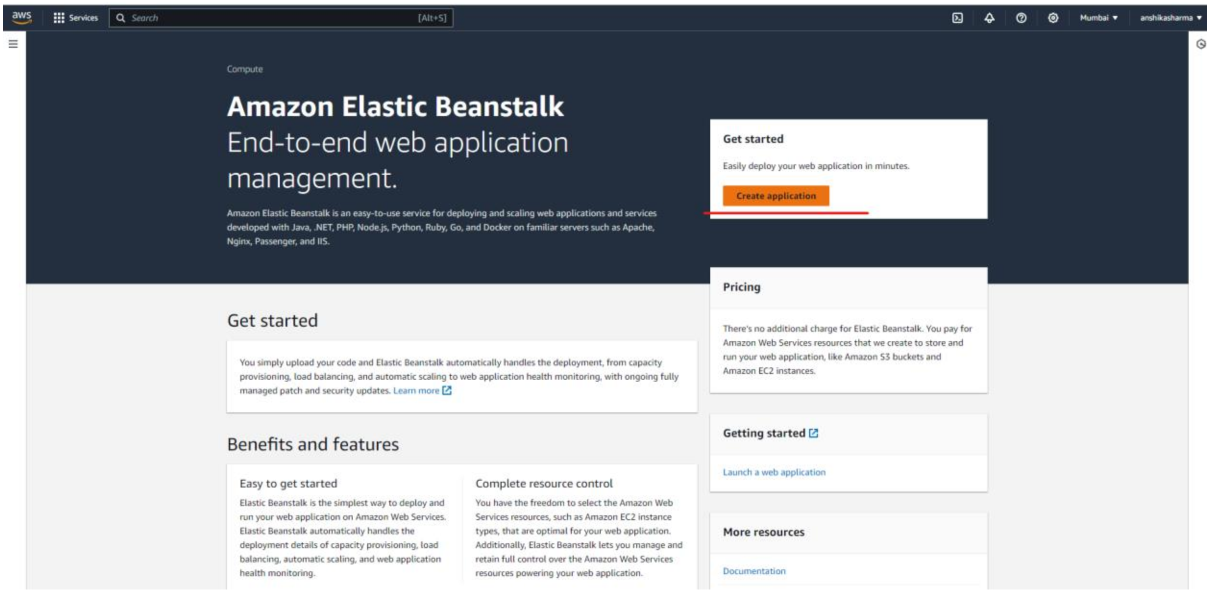
1.Open the AWS console



2.Select services (6 dots) then select “**Compute**”, select “**Elastic Beanstalk**”.



3.Click on “**create application**”.



Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon 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.

Get started

Easily deploy your web application in minutes.

Create application

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

Getting started

Launch a web application

More resources

Documentation

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. Learn more

Benefits and features

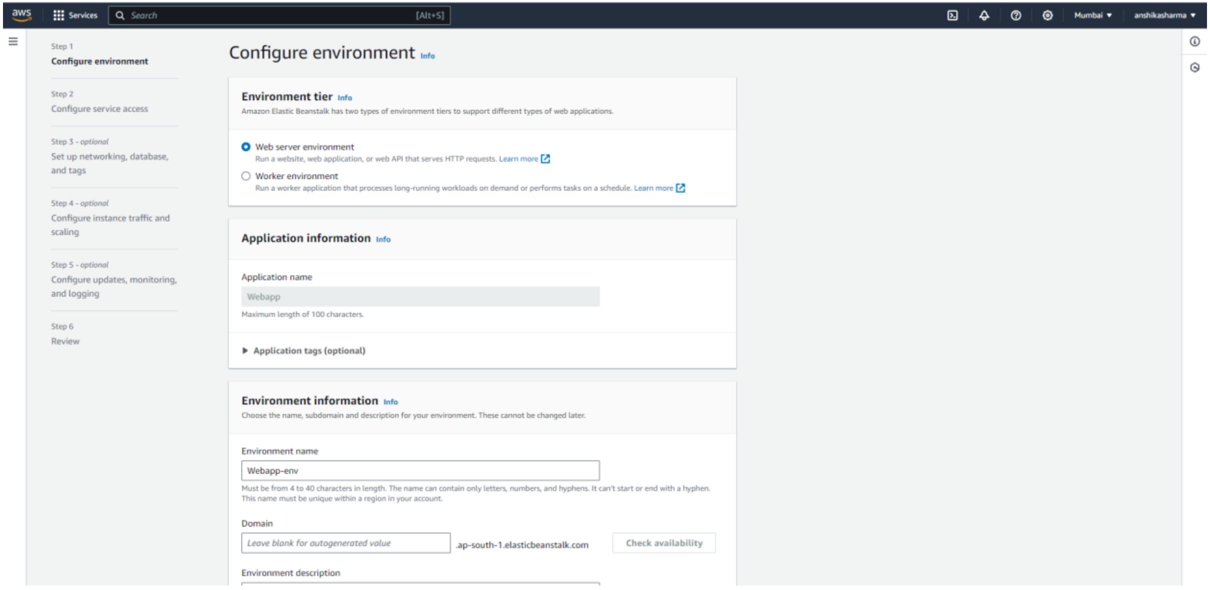
Easy to get started

Elastic Beanstalk is the simplest way to deploy and run your web application on Amazon Web Services. Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, automatic scaling, and web application health monitoring.

Complete resource control

You have the freedom to select the Amazon Web Services resources, such as Amazon EC2 instance types, that are optimal for your web application. Additionally, Elastic Beanstalk lets you manage and retain full control over the Amazon Web Services resources powering your web application.

4.Step 1 appears where you need to configure environment. Create a web page by giving **application name as Webapp and environment name as Webapp-env.**



Step 1
Configure environment

Step 2
Configure service access

Step 3 - *optional*
Set up networking, database, and tags

Step 4 - *optional*
Configure instance traffic and scaling

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

● Web server environment
Run a website, web application, or web API that serves HTTP requests. Learn more

○ Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more

Application information Info

Application name

Webapp

Maximum length of 100 characters.

▸ Application tags (optional)

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Webapp-env

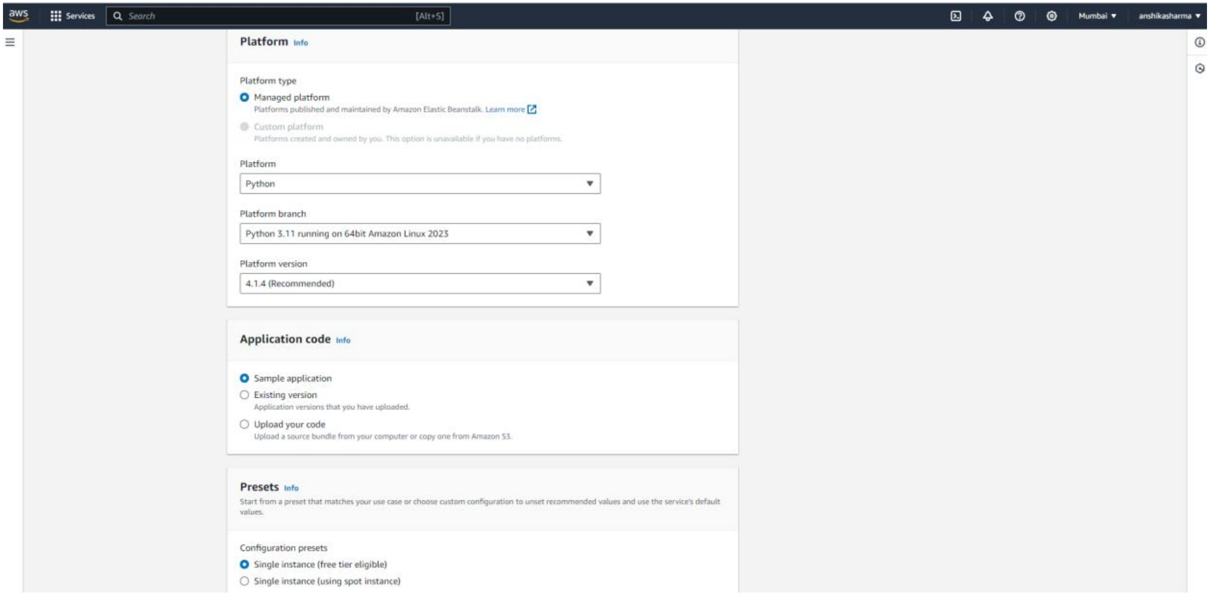
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

Leave blank for autogenerated value .ap-south-1.elasticbeanstalk.com Check availability

Environment description

5.Select **Python** platform and under Application Code select **Sample Application**.



Platform Info

Platform type

● Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

○ Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform: Python ▾

Platform branch: Python 3.11 running on 64bit Amazon Linux 2023 ▾

Platform version: 4.1.4 (Recommended) ▾

Application code Info

● Sample application

○ Existing version
Application versions that you have uploaded.

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

Presets Info

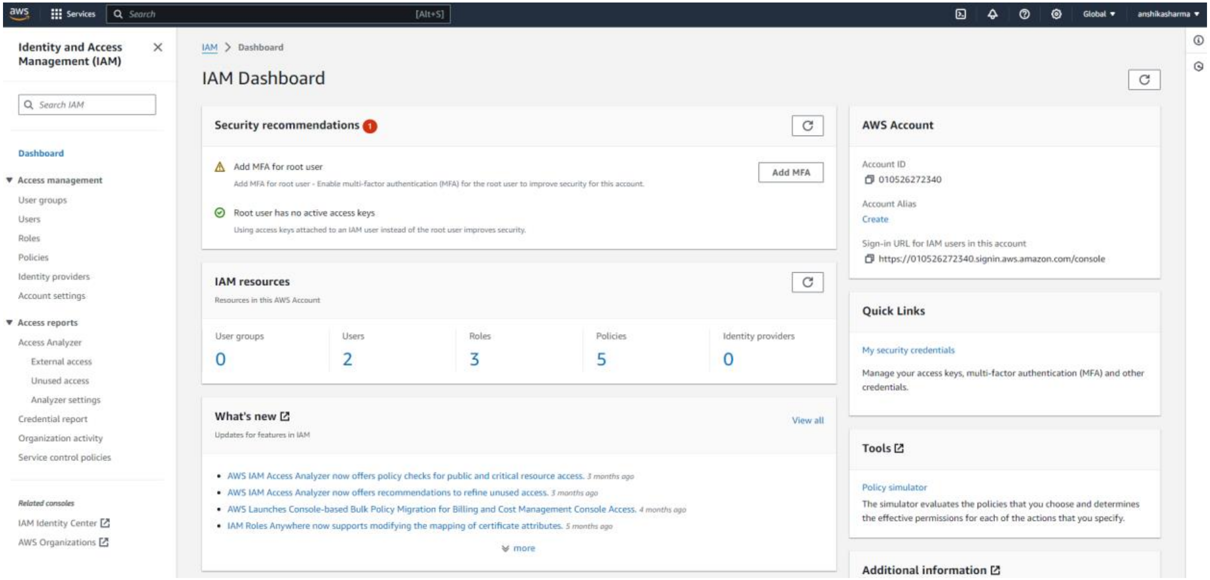
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

● Single instance (free tier eligible)

○ Single instance (using spot instance)

6. Now navigate to IAM dashboard and select **Roles** under **Access management** from the left pane.



IAM Dashboard

Security recommendations 1

⚠ Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
Add MFA

✓ Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources
Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	2	3	5	0

What's new
Updates for features in IAM
View all

- AWS IAM Access Analyzer now offers policy checks for public and critical resource access. *3 months ago*
- AWS IAM Access Analyzer now offers recommendations to refine unused access. *3 months ago*
- AWS Launches Console-based Bulk Policy Migration for Billing and Cost Management Console Access. *4 months ago*
- IAM Roles Anywhere now supports modifying the mapping of certificate attributes. *5 months ago*

more

AWS Account

Account ID
010526272340

Account Alias
Create

Sign-in URL for IAM users in this account
https://010526272340.signin.aws.amazon.com/console

Quick Links

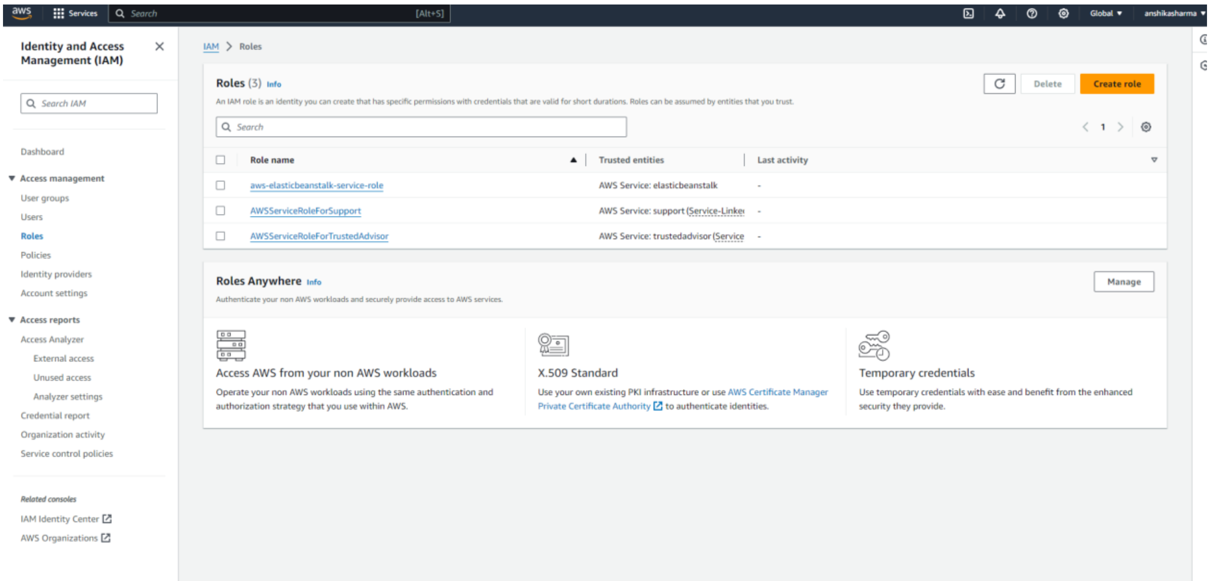
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Tools

Policy simulator
The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.

Additional information

7.Click on **Create Role.**



Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities	Last activity
□	aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

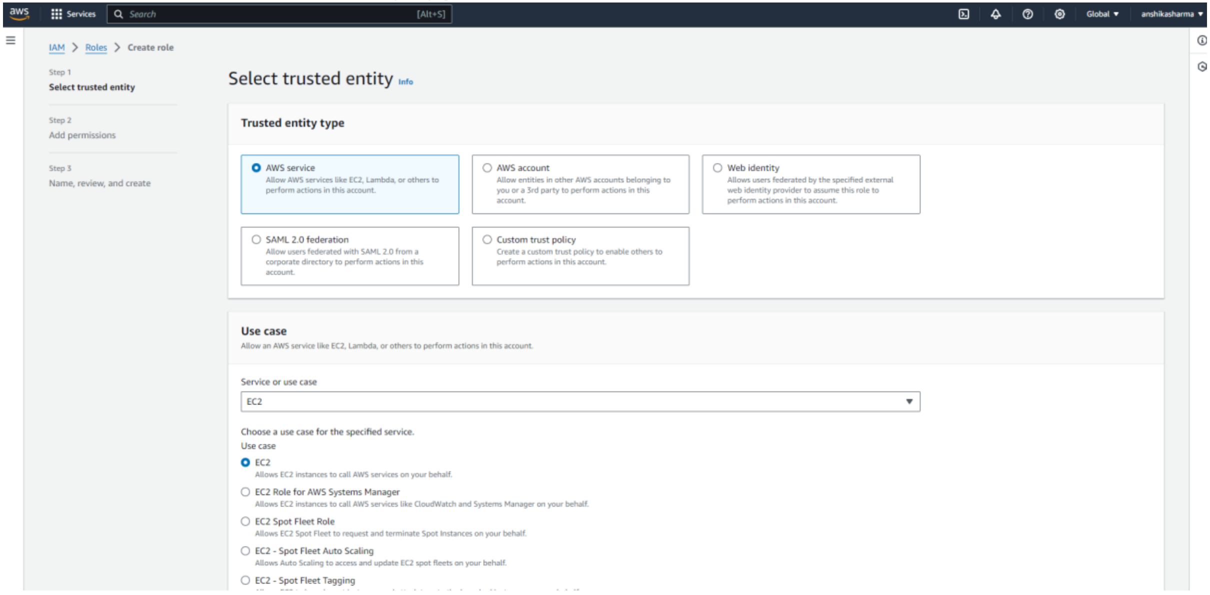
X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

8.In step 1 the ‘trusted Entity type’ should be **AWS service**. Select service as **EC2**.



Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case: EC2 ▾

Choose a use case for the specified service.

Use case

● EC2
Allows EC2 instances to call AWS services on your behalf.

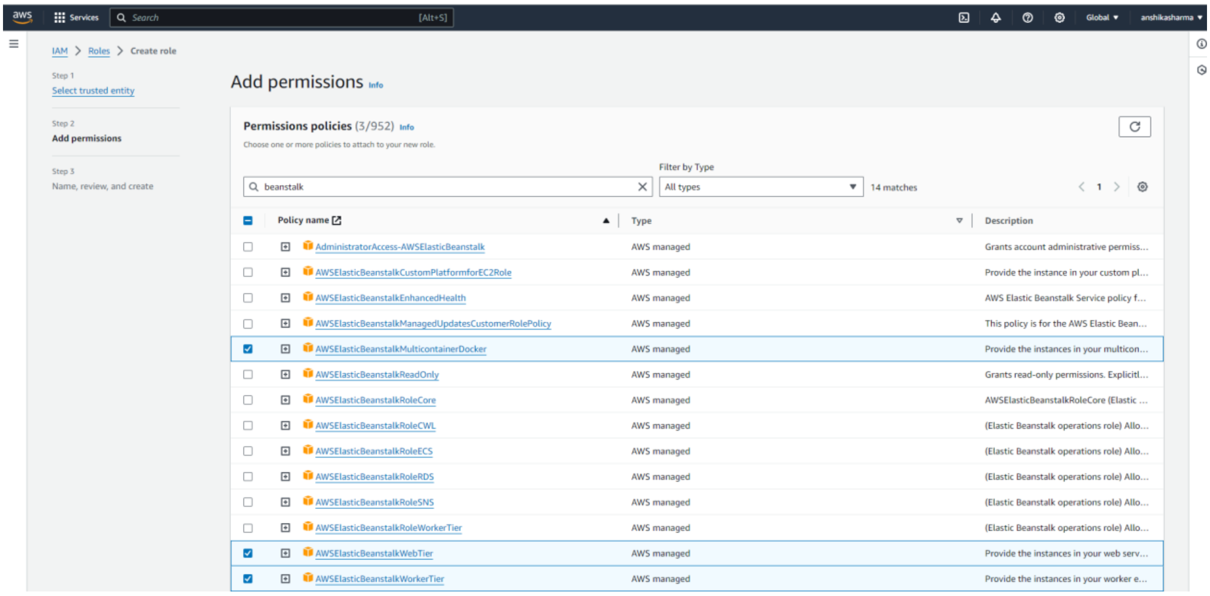
○ EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

○ EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

○ EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

○ EC2 - Spot Fleet Tagging

9.In step 2 ‘Add Permissions’. Select the below permission policies for elastic beanstalk.



Add permissions Info

Permissions policies (3/952) Info

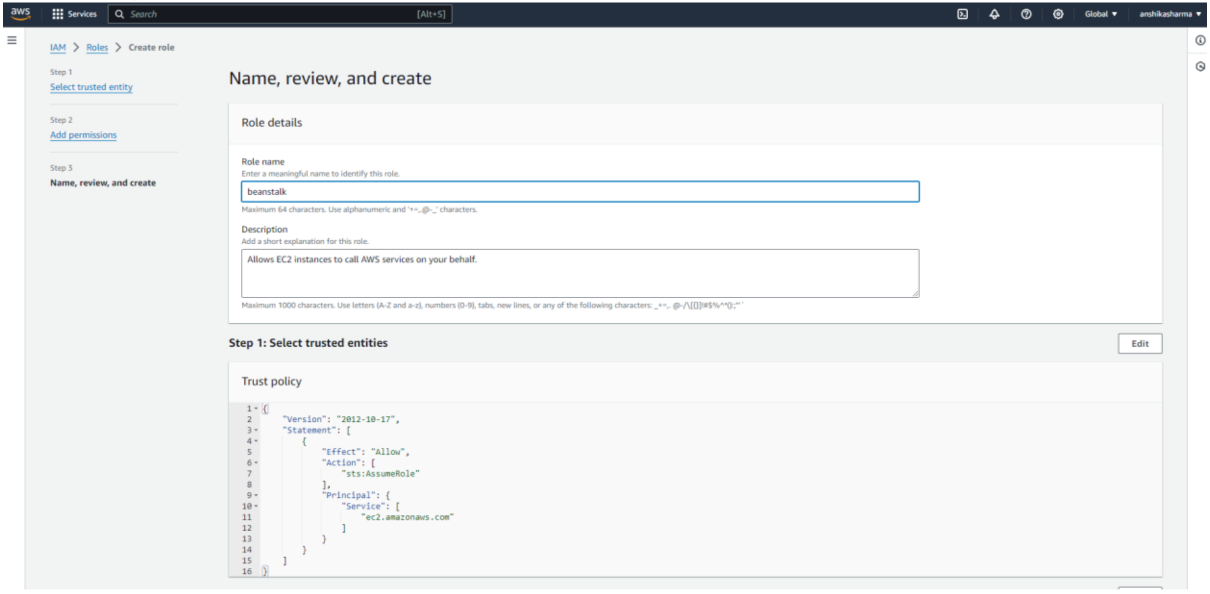
Choose one or more policies to attach to your new role.

Filter by Type

beanstalk | All types | 14 matches

	Policy name	Type	Description
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permiss...
□	AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	Provide the instance in your custom pl...
□	AWSElasticBeanstalkEnhancedHealth	AWS managed	AWS Elastic Beanstalk Service policy f...
□	AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	This policy is for the AWS Elastic Bean...
☑	AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicon...
□	AWSElasticBeanstalkReadOnly	AWS managed	Grants read-only permissions. Explicitl...
□	AWSElasticBeanstalkRoleCore	AWS managed	AWSElasticBeanstalkRoleCore (Elastic ...
□	AWSElasticBeanstalkRoleCWL	AWS managed	(Elastic Beanstalk operations role) Allo...
□	AWSElasticBeanstalkRoleECS	AWS managed	(Elastic Beanstalk operations role) Allo...
□	AWSElasticBeanstalkRoleRDS	AWS managed	(Elastic Beanstalk operations role) Allo...
□	AWSElasticBeanstalkRoleSNS	AWS managed	(Elastic Beanstalk operations role) Allo...
□	AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allo...
☑	AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web serv...
☑	AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker e...

10.In Step 3 Give the role name , review and create.



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

beanstalk

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

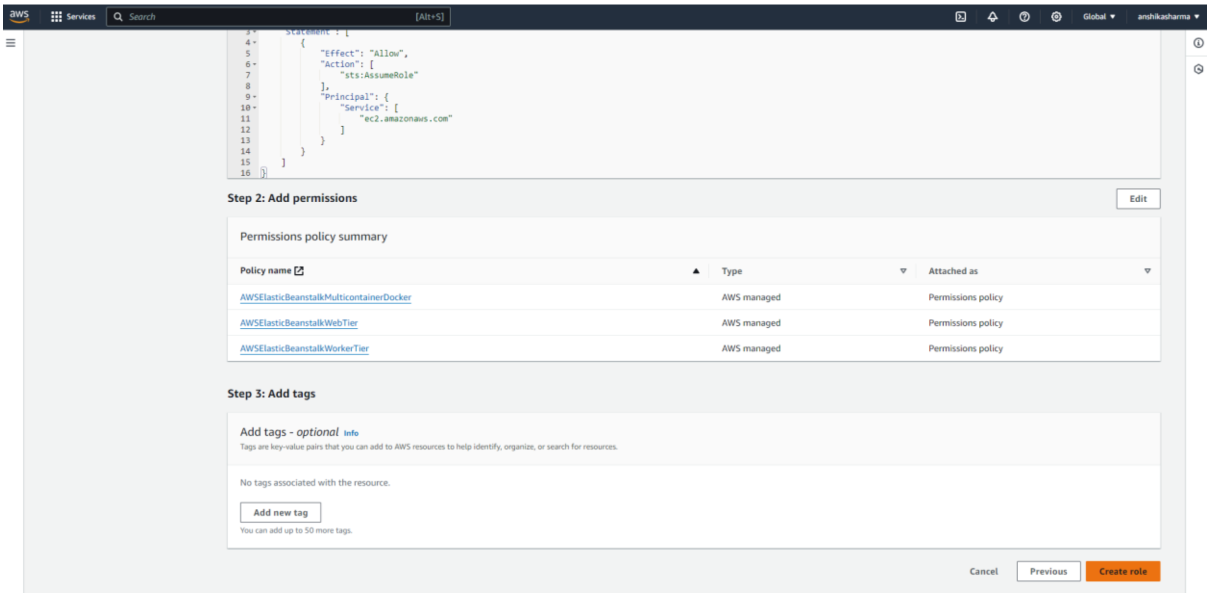
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'`

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    ]
}
```

11. Click on **Create Role**.



```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    ]
}
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

Step 3: Add tags

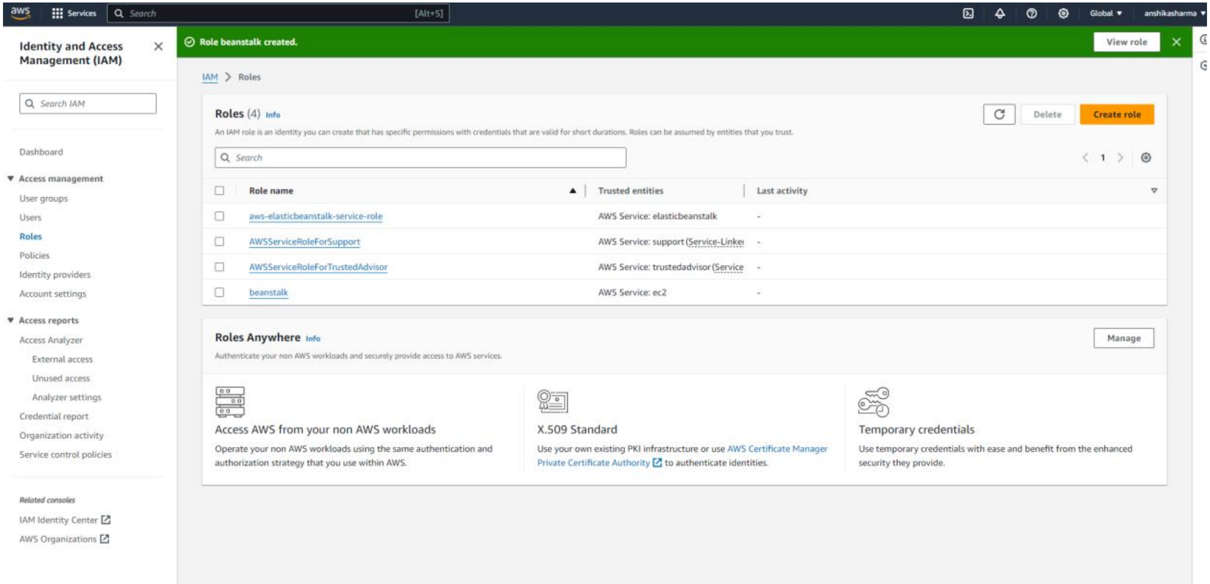
Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

You can add up to 50 more tags.

12. The Role gets Created.13



Role beanstalk created.

Roles (4) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities	Last activity
□	aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	beanstalk	AWS Service: ec2	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

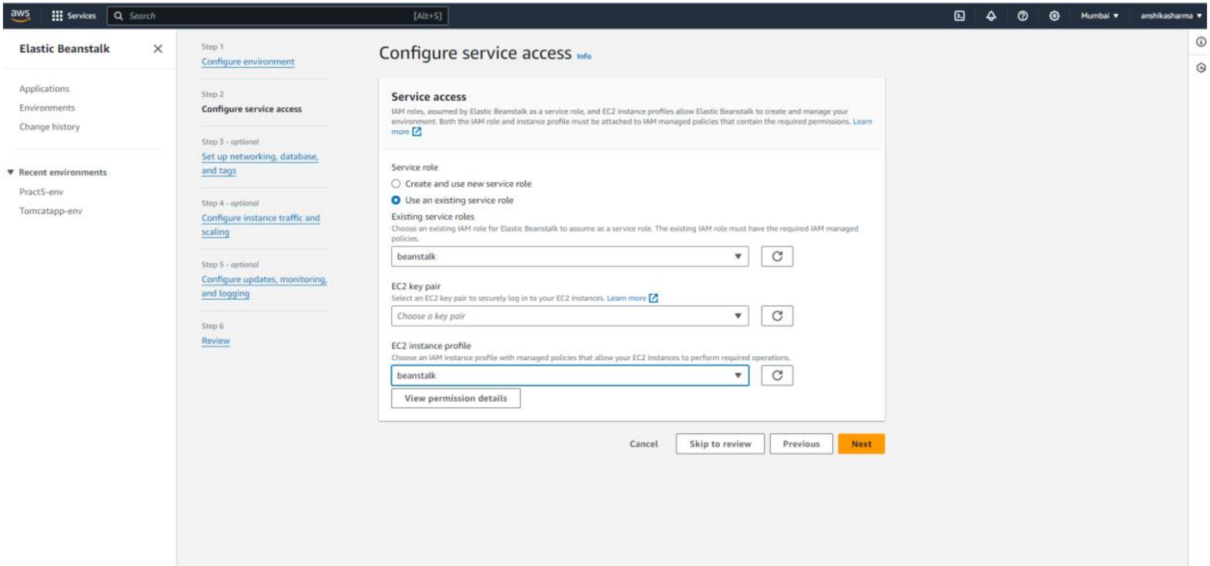
X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

13. Navigate back to the previous steps where we configured the environment in step 1. Now in Step 2 we configure service access. Select ‘**Use existing service role**’ under **Service role**. Now use **beanstalk role** which you created on IAM.



Configure service access Info

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more

Service role

○ Create and use new service role

● Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

beanstalk

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

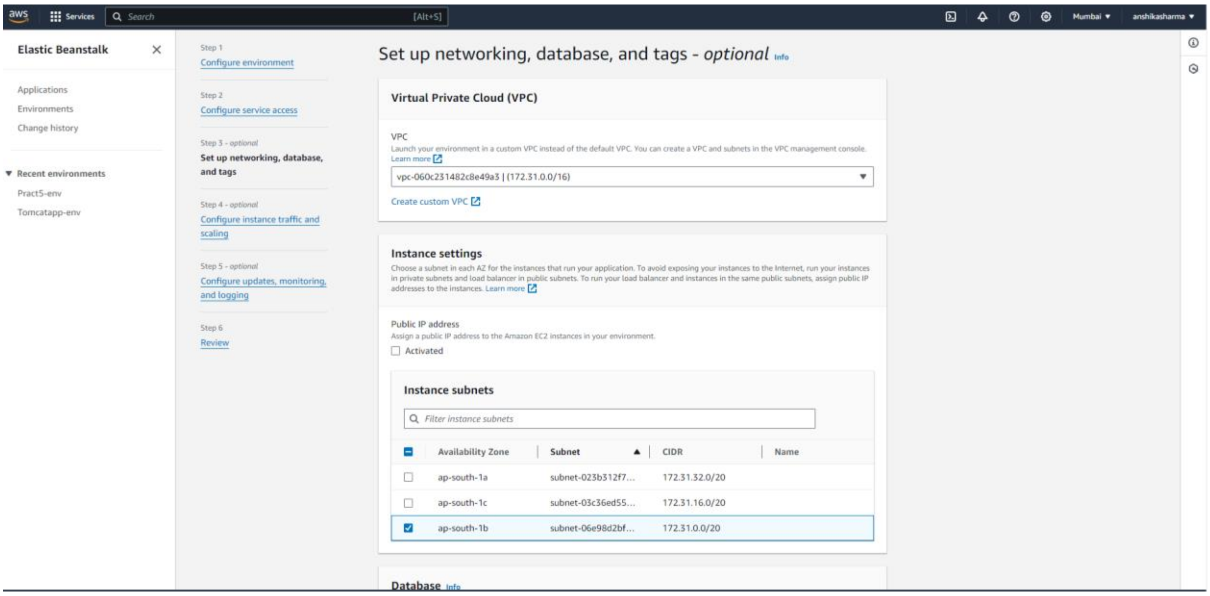
Choose a key pair

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

beanstalk

14.In step 3 set up networking, database and tags.



Set up networking, database, and tags - *optional* Info

Virtual Private Cloud (VPC)

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. Learn more

vpc-060c231482c8e49a3 | (172.31.0.0/16)

Create custom VPC

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. Learn more

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

□ Activated

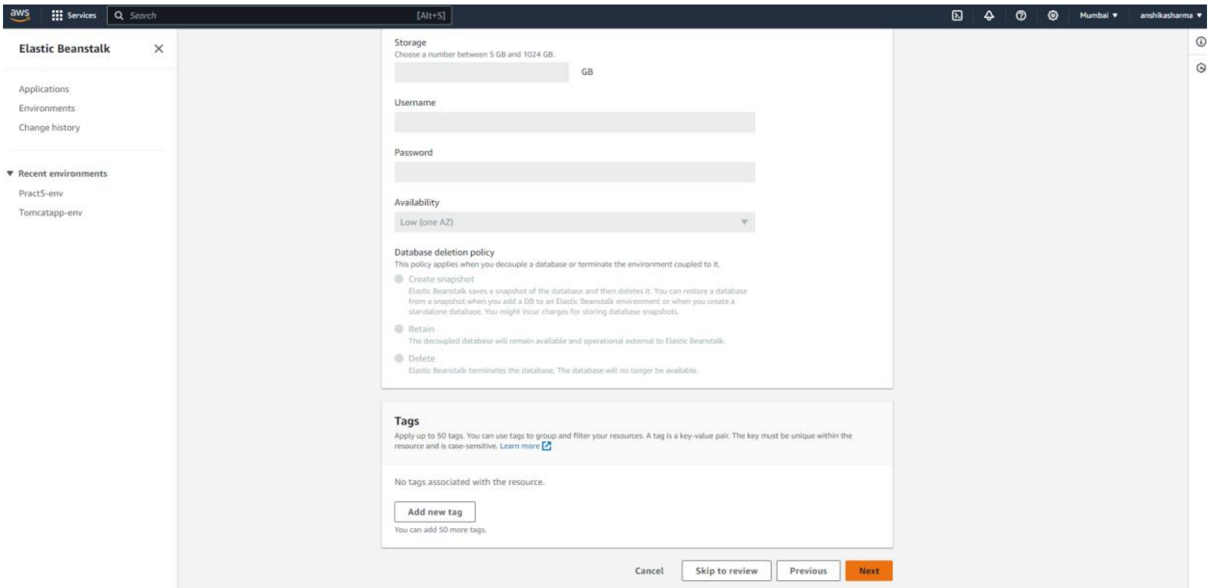
Instance subnets

Filter instance subnets

	Availability Zone	Subnet	CIDR	Name
□	ap-south-1a	subnet-023b312f7...	172.31.32.0/20	
□	ap-south-1c	subnet-03c36ed55...	172.31.16.0/20	
☑	ap-south-1b	subnet-06e98d2bf...	172.31.0.0/20	

Database Info

15. Click on next.



Storage

Choose a number between 5 GB and 1024 GB.

GB

Username

Password

Availability

Low (one AZ)

Database deletion policy

This policy applies when you decouple a database or terminate the environment coupled to it.

○ Create snapshot

Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.

○ Retain

The decoupled database will remain available and operational external to Elastic Beanstalk.

○ Delete

Elastic Beanstalk terminates the database. The database will no longer be available.

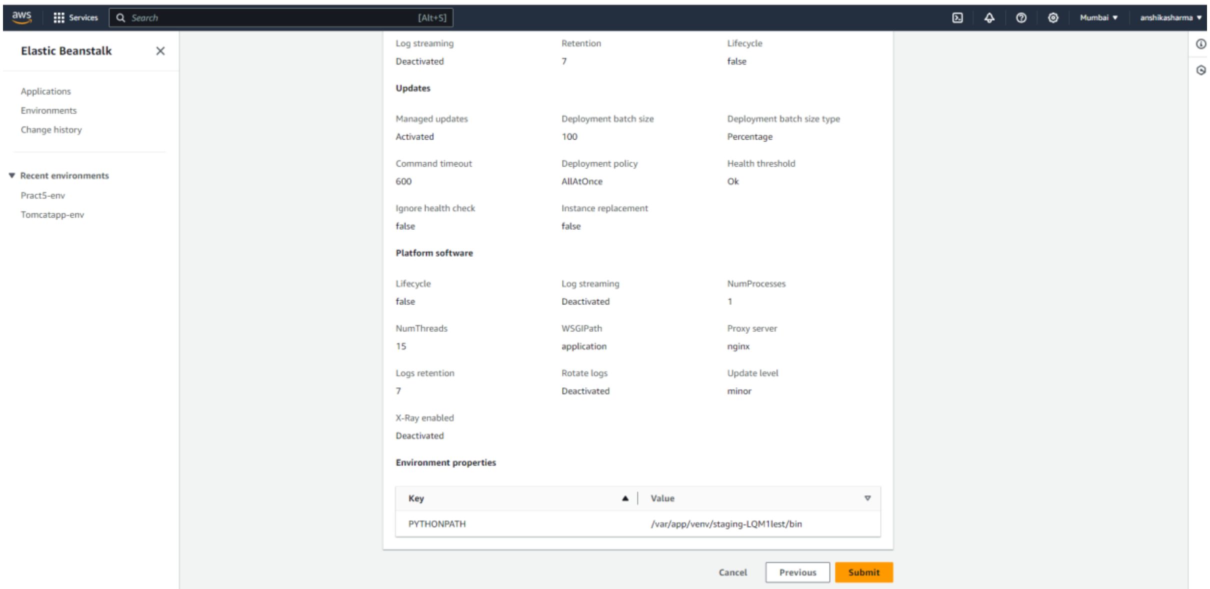
Tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

No tags associated with the resource.

You can add 50 more tags.

16. Click Submit.



Log streaming
Deactivated

Retention
7

Lifecycle
false

Updates

Managed updates
Activated

Deployment batch size
100

Deployment batch size type
Percentage

Command timeout
600

Deployment policy
AllAtOnce

Health threshold
Ok

Ignore health check
false

Instance replacement
false

Platform software

Lifecycle
false

Log streaming
Deactivated

NumProcesses
1

NumThreads
15

WSGIPath
application

Proxy server
nginx

Logs retention
7

Rotate logs
Deactivated

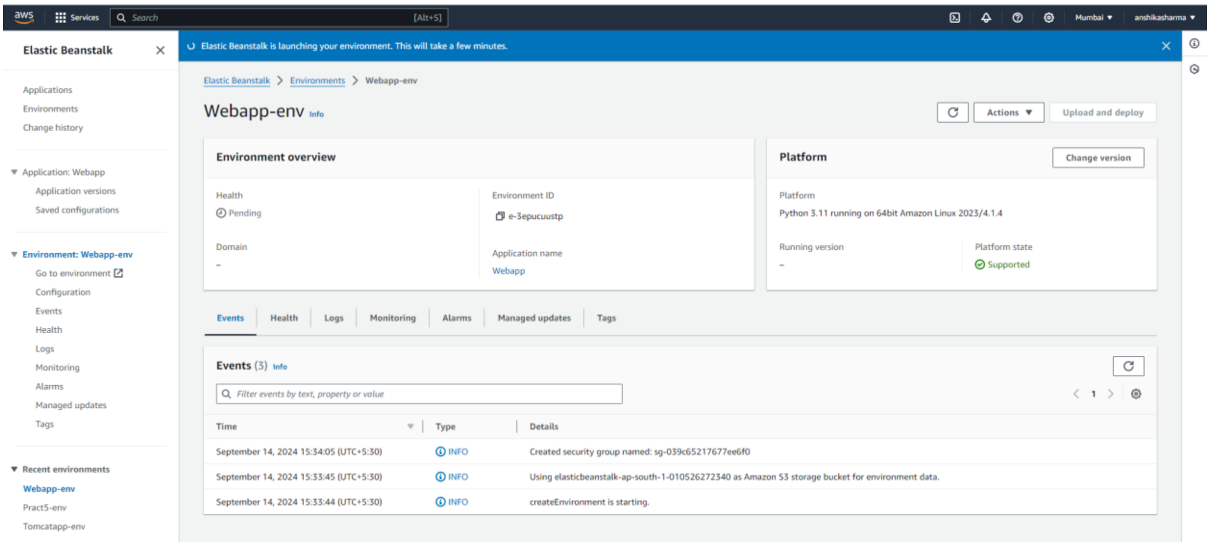
Update level
minor

X-Ray enabled
Deactivated

Environment properties

Key	Value
PYTHONPATH	/var/app/venv/staging-LQM1lest/bin

17. Elastic Beanstalk gets launches in the Environment.



Elastic Beanstalk is launching your environment. This will take a few minutes.

Elastic Beanstalk > Environments > Webapp-env

Webapp-env Info

Environment overview

Health: Pending

Environment ID: e-3epucuustp

Domain: –

Application name: Webapp

Platform

Platform: Python 3.11 running on 64bit Amazon Linux 2023/4.1.4

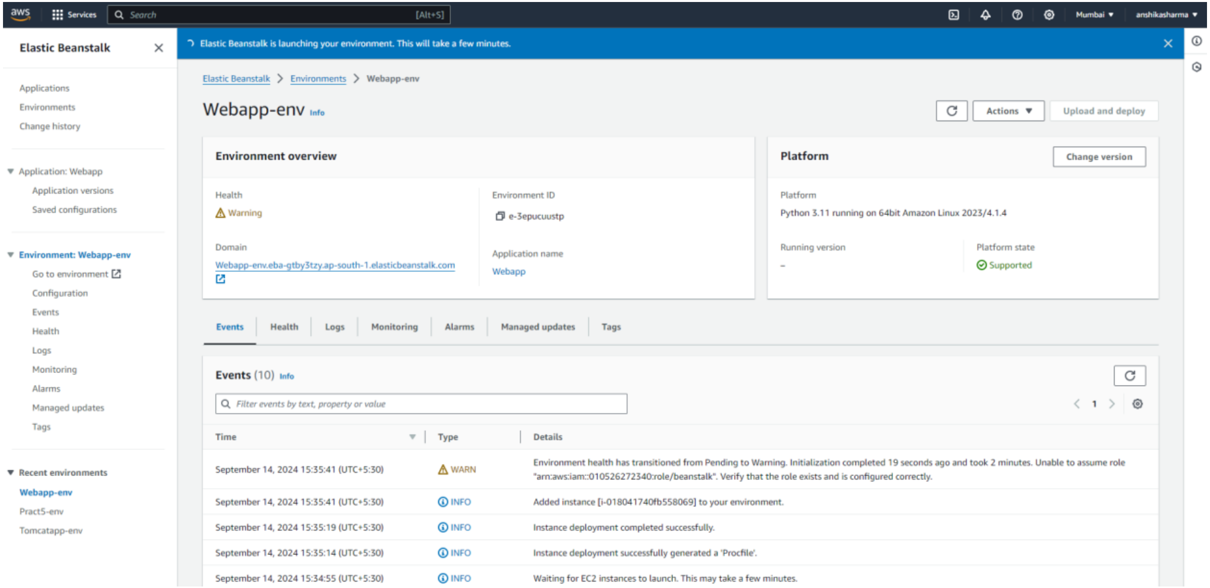
Running version: –

Platform state: Supported

Events (3) Info

Time	Type	Details
September 14, 2024 15:34:05 (UTC+5:30)	INFO	Created security group named: sg-039c65217677ee6f0
September 14, 2024 15:33:45 (UTC+5:30)	INFO	Using elasticbeanstalk-ap-south-1-010526272340 as Amazon S3 storage bucket for environment data.
September 14, 2024 15:33:44 (UTC+5:30)	INFO	createEnvironment is starting.

18. Copy the **Domain** address and paste it in the browser.



Elastic Beanstalk is launching your environment. This will take a few minutes.

Elastic Beanstalk > Environments > Webapp-env

Webapp-env Info

Environment overview

Health: Warning

Environment ID: e-3epucuustp

Domain: Webapp-env.eba-gtby3tzy.ap-south-1.elasticbeanstalk.com

Application name: Webapp

Platform

Platform: Python 3.11 running on 64bit Amazon Linux 2023/4.1.4

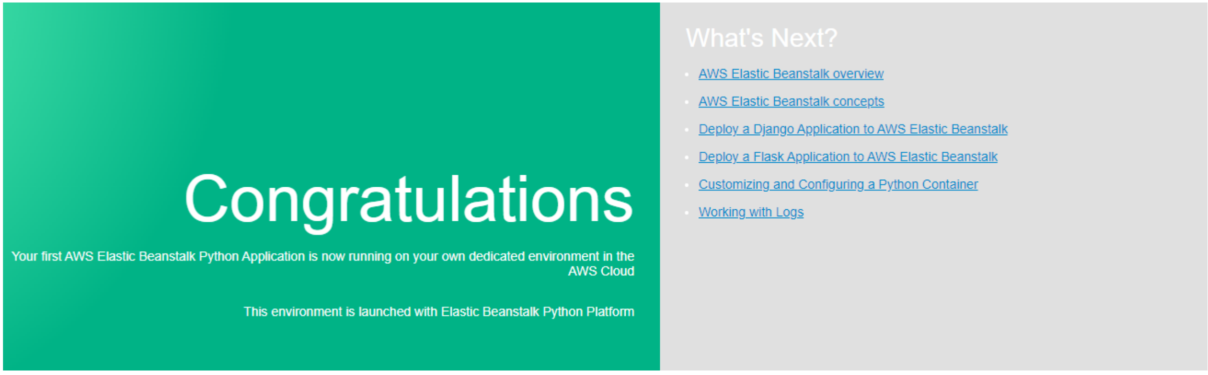
Running version: –

Platform state: Supported

Events (10) Info

Time	Type	Details
September 14, 2024 15:35:41 (UTC+5:30)	WARN	Environment health has transitioned from Pending to Warning. Initialization completed 19 seconds ago and took 2 minutes. Unable to assume role "arn:aws:iam::010526272340:role/beanstalk". Verify that the role exists and is configured correctly.
September 14, 2024 15:35:41 (UTC+5:30)	INFO	Added instance [i-018041740fb558069] to your environment.
September 14, 2024 15:35:19 (UTC+5:30)	INFO	Instance deployment completed successfully.
September 14, 2024 15:35:14 (UTC+5:30)	INFO	Instance deployment successfully generated a 'Procfile'.
September 14, 2024 15:34:55 (UTC+5:30)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.

19. Below page appears. Created successfully.



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