

Roll NO. A041

## **Cloud Computing Practical No: 02**

### **Writeup:**

#### **1.Platform as a service**

Platform as a Service (PaaS) is a cloud computing service model that provides a platform allowing customers to develop, run, and manage applications without dealing with the complexity of building and maintaining the underlying infrastructure. PaaS sits between Infrastructure as a Service (IaaS) and Software as a Service (SaaS) in the cloud computing service stack.

#### **Characteristics of PaaS:**

**Development Frameworks:** PaaS provides a set of tools, services, and development frameworks that allow developers to build, test, and deploy applications more efficiently.

**Middleware Services:** PaaS often includes middleware services such as databases, messaging systems, and caching services. These services abstract the underlying infrastructure, making it easier for developers to focus on application development.

**Automated Deployment and Scaling:** PaaS platforms automate the deployment and scaling of applications. Developers can easily scale their applications up or down based on demand without having to manage the underlying infrastructure.

**Integrated Development Tools:** PaaS offers integrated development tools, including code repositories, version control, and collaboration tools, facilitating collaboration among development teams.

**Multi-Tenancy:** PaaS platforms are designed to support multiple users or tenants, allowing developers to share development tools and resources in a secure and efficient manner.

#### **Advantages of PaaS:**

**Faster Development:** PaaS accelerates the development process by providing ready-to-use components and services.

**Cost-Efficiency:** Users can avoid the costs and complexities associated with maintaining and managing infrastructure.

**Scalability:** PaaS platforms offer automatic scaling, ensuring applications can handle varying workloads.

#### **Disadvantages of PaaS:**

**Less Control:** Developers have less control over the underlying infrastructure compared to IaaS.

**Dependency on Provider:** Users are dependent on the PaaS provider for updates, security patches, and overall platform stability.

## 2. Amazon Elastic Beanstalk

Amazon Elastic Beanstalk is a web infrastructure management service. It handles deployment and scaling for web applications and services. Elastic Beanstalk can automatically manage setup, configuration, scaling and provisioning for other AWS services. AWS Elastic Beanstalk is an AWS-managed service for web applications. Elastic Beanstalk is a pre-configured EC2 server that can directly take up your application code and environment configurations and use it to automatically provision and deploy the required resources within AWS to run the web application. Unlike EC2 which is Infrastructure as a service, Elastic Beanstalk is a Platform As A Service (PAAS) as it allows users to directly use a pre-configured server for their application. Of course, you can deploy applications without ever having to use elastic beanstalk but that would mean having to choose the appropriate service from the vast array of services offered by AWS, manually provisioning these AWS resources, and stitching them up together to form a complete web application. Elastic Beanstalk abstracts the underlying configuration work and allows you as a user to focus on more pressing matters.

### 3.) Components of Amazon Elastic Beanstalk

- **Application:** Elastic Beanstalk directly takes in our project code. So Elastic Beanstalk application is named the same as your project home directory.
- **Application Environments:** Users may want their application to run on different environments like DEV, UAT, and PROD. You can create and configure different environments to run applications on different stages.
- **Environment Health:** One of the most lucrative features of running applications on AWS or most of the other cloud platforms is automated health checks. AWS runs automatic health checks on all EC-2 deployments (Elastic Beanstalk is a managed EC-2 service) which can be monitored from the AWS console. For example, in the case of web applications AWS will regularly, as scheduled by the developers, ping the application to check if the response is status code 200 and if the application is running as expected. Health check responses:
  - **Red:** The application failed all health tests.
  - **Yellow:** The application failed some of the health tests.
  - **Grey:** The application is updating.
  - **Green:** The application passed the health check successfully.
- **Isolated:** All environments within a single application are isolated from each other (independent of each others' running states). Needless to say, two different applications are also isolated.
- **Scalability:** Using Auto-Scaling within Elastic Beanstalk makes the application dynamically scalable.
- **Elastic Load Balancing:** All the web requests to the application are not directly relayed to application instances. They first hit the Elastic Load Balancer (ELB), which, as the name suggests, balances the load across all the application instances.
- **Language support:** Elastic Beanstalk supports the applications developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

- **Pricing:** There is no extra charge for using Elastic Beanstalk. Users are only required to pay for the services and resources provisioned by Elastic Beanstalk Service.
- **Automatic Provisioning:** Elastic Beanstalk takes away the burden of choosing the right services and configuring their security groups to work together.
- **Impossible to Outgrow:** AWS claims that since Elastic Beanstalk uses the Auto Scaling feature it can, in theory, handle any amount of internet traffic.

#### 4.)IAM:

IAM stands for Identity and Access Management. In the context of Amazon Web Services (AWS), IAM refers to the service that allows you to manage access to AWS resources securely. IAM enables you to control who (authentication) can do what (authorization) in your AWS environment.

Here are key aspects of AWS Identity and Access Management (IAM):

##### **Users and Groups:**

**Users:** Represent individuals or entities that interact with AWS services. Each user has a unique set of security credentials.

**Groups:** Users can be organized into groups, and permissions can be assigned to groups, making it easier to manage access.

**Roles:** IAM roles define a set of permissions for making AWS service requests. Roles are not associated with a specific user or group but can be assumed by users, applications, or services when needed.

**Policies:** IAM policies are JSON documents that define permissions. They can be attached to users, groups, or roles, specifying what actions are allowed or denied on what resources.

**Access Keys:** IAM provides access keys (access key ID and secret access key) for programmatic access to AWS services. These keys are often used by developers and applications.

**Multi-Factor Authentication (MFA):** IAM supports MFA, an additional layer of security that requires users to provide a second form of authentication (such as a code from a virtual or hardware MFA device) in addition to their password.

**Identity Federation:** IAM allows you to integrate with external identity providers, such as Active Directory or social identity providers, to grant temporary access to AWS resources.

**Resource-Level Permissions:** IAM policies can define permissions not only at the service level but also at the resource level. This allows fine-grained control over access to specific AWS resources.

**IAM Roles for EC2 Instances:** IAM roles can be assigned to EC2 instances, allowing applications running on those instances to securely access AWS resources without embedding credentials in the code.

**Policy Conditions:** IAM policies can include conditions that must be met for the policy to be in effect. Conditions can be based on factors such as the time of day, the source IP address, or the use of MFA.

**Implement paas using elastic beanstalk for the following.**

**1. Server**

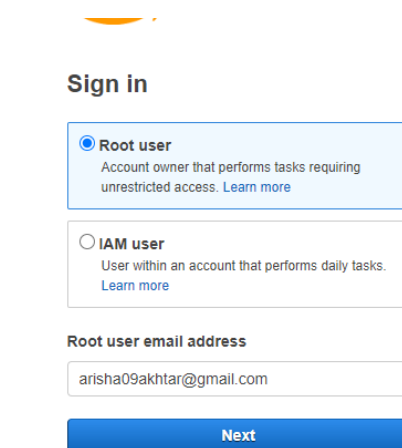
**2. Java**

**3. Python**

**4. Node.js**

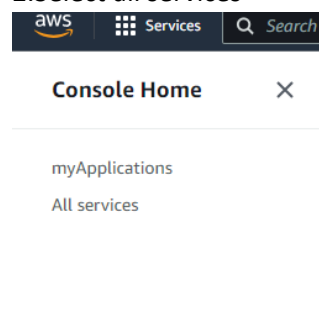
**For Server**

1) Sign In to your aws acc



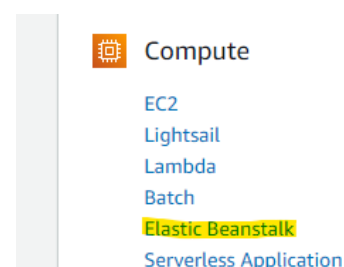
The image shows the AWS sign-in interface. At the top is the AWS logo. Below it is the text "Sign in". There are two radio button options: "Root user" (selected) and "IAM user". The "Root user" option has a description: "Account owner that performs tasks requiring unrestricted access. [Learn more](#)". The "IAM user" option has a description: "User within an account that performs daily tasks. [Learn more](#)". Below these options is a text input field labeled "Root user email address" containing the email "arisha09akhtar@gmail.com". At the bottom is a blue "Next" button.

2. Select all services



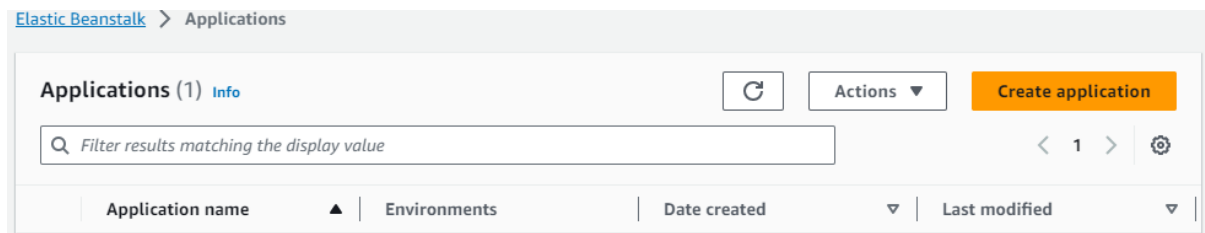
The image shows the AWS Services console. At the top is the AWS logo, a "Services" tab, and a search bar. Below the search bar is a "Console Home" section with a close button (X). Under "Console Home", there are two links: "myApplications" and "All services".

3. Select Elastic Beanstalk



The image shows a list of AWS Compute services. The list includes: "Compute" (with a gear icon), "EC2", "Lightsail", "Lambda", "Batch", "Elastic Beanstalk" (highlighted in yellow), and "Serverless Application".

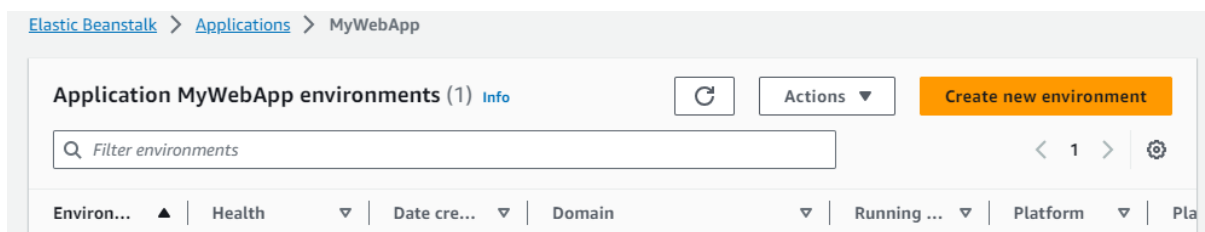
#### 4. Click on Create Application



#### 5. Enter application name and description and then click on create

The screenshot shows the 'Create new application' form. The title is 'Create new application Info'. Under the 'Application information' section, there are two input fields. The first is 'Application name' with the value 'MyWebApp' and a note 'Maximum length of 100 characters.' The second is 'Description' with the value 'my first demo'.

#### 6. Now create environment



#### 7. For Configure environment everything will be default except platform since we are doing for server we will choose .Net on Windows server Platform branch and platform version will be default. Click on next.

The screenshot shows the 'Environment tier' section with two options: 'Web server environment' (selected) and 'Worker environment'. Below is the 'Application information' section with 'Application name' set to 'webapp'. The 'Environment information' section shows 'Environment name' as 'Webapp-env' and 'Domain' as 'eu-north-1.elasticbeanstalk.com'.The screenshot shows the 'Platform' configuration section. Under 'Platform type', 'Managed platform' is selected. The 'Platform' dropdown is set to '.NET on Windows Server'. The 'Platform branch' dropdown is set to 'IIS 10.0 running on 64bit Windows Server 2019'. The 'Platform version' dropdown is set to '2.13.2 (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)

☐ High availability

☐ High availability (using spot and on-demand instances)

☐ Custom configuration

Cancel

Next

8. For Service access you have to create a role since it is by default over here you can create your own role .

### 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.

aws-elasticbeanstalk-service-role

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.

View permission details

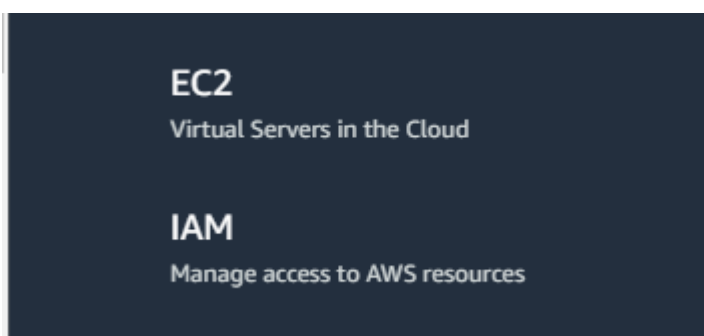
Cancel

Skip to review

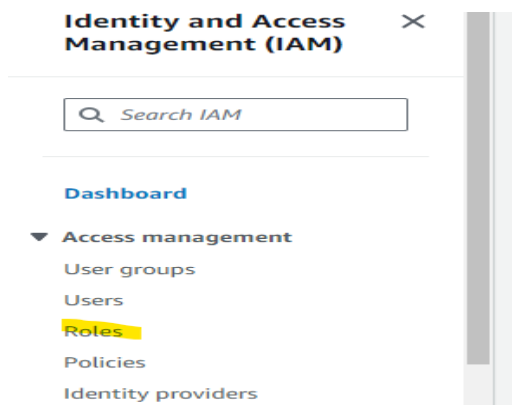
Previous

Next

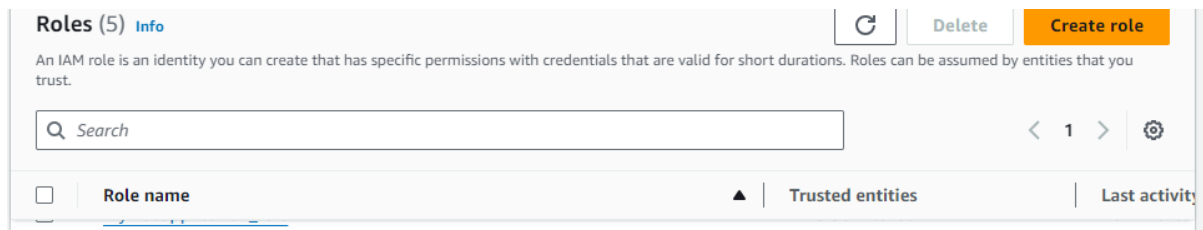
9. To create role go to services IAM



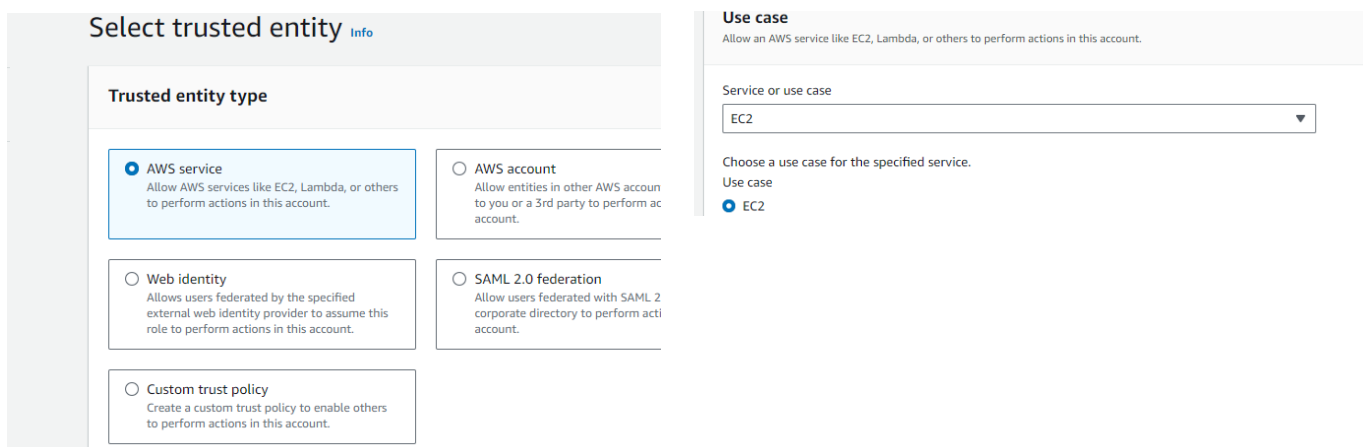
10. Inside IAM dashboard go to roles



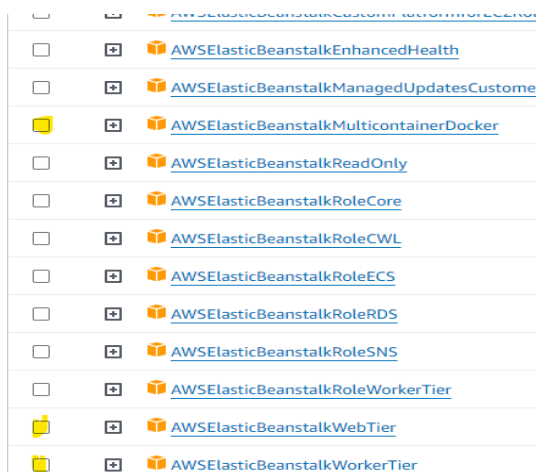
11. Create role



12. It will ask for entity select AWS Service and for use case select EC2 and click on next



13. Search elasticbeanstalk on top select this three opt and click on next



14. Give role name and click on create role

**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.

**Add new tag**

You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create role](#)

15. Role will get created

**Role MyWebAPP created.** [View role](#)

**Roles (6)** [Info](#) [Refresh](#) [Delete](#) [Create role](#)

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](#)

16. Now you can again go to your service access that is your 8 step now inside existing service roles refresh it and click on dropdown you can see over there your role name which you have created above automatically comes there click on that

**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.

[Refresh](#)

**EC2 key pair**

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

[Refresh](#)

**EC2 instance profile**

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

[Refresh](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

17. For instance profile click on dropdown your role name will come select that then enter next

**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.

[Refresh](#)

**EC2 key pair**

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

[Refresh](#)

**EC2 instance profile**

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

[Refresh](#)

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)



18. After clicking on next it will ask for VPC click on dropdown and select which is default not down the ip address of this

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-08f10f70caf202076 | (172.31.0.0/16) ▼

[Create custom VPC](#)

19. The ip address you have selected above same over here select that ip address and click on next

**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
<input type="checkbox"/>	ap-south-1a	subnet-0099214b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-058970f16...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07fe5d047...	172.31.16.0/20	

**Database** [info](#)  
Integrate an RDS SQL database with your environment. [Learn more](#)

**Database subnets**  
If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. [Learn more](#)

**Choose database subnets (3)**

Filter database subnets

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-0099214b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-058970f16...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07fe5d047...	172.31.16.0/20	

20. for configure instance everything will be by default enter next

Configure instance traffic and scaling - *optional* [info](#)

**Instances** [info](#)  
Configure the Amazon EC2 instances that run your application.

**Root volume (boot device)**

Root volume type  
(Container default) ▼

Size  
The number of gigabytes of the root volume attached to each instance.  
8 GB

IOPS  
Input/output operations per second for a provisioned IOPS (SSD) volume.  
100 IOPS

Throughput  
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance  
125 MB/s

**Amazon CloudWatch monitoring**  
The time interval between when metrics are reported from the EC2 instances  
Monitoring interval  
5 minute ▼

**Instance metadata service (IMDS)**  
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

IMDSv1

21. For configure update also everything will be default click to next

## Configure updates, monitoring, and logging - *optional* [Info](#)

▼ Monitoring [Info](#)

### Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#) [↗](#)

System

☐ Basic

☒ Enhanced

CloudWatch Custom Metrics - Instance

Choose metrics ▼

22. Review will display click on submit

## Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	MyWebApp
Environment name	Application code
MyWebApp-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/Java 8 running on 64bit Windows Server 2019/2.13.2	

Step 2: Configure service access [Edit](#)

Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::836706723086:role/MyWebAPP	MyWebAPP

Step 3: Set up networking, database, and tags [Edit](#)

23. You can see it is getting launch

search [Alt+S]

×

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

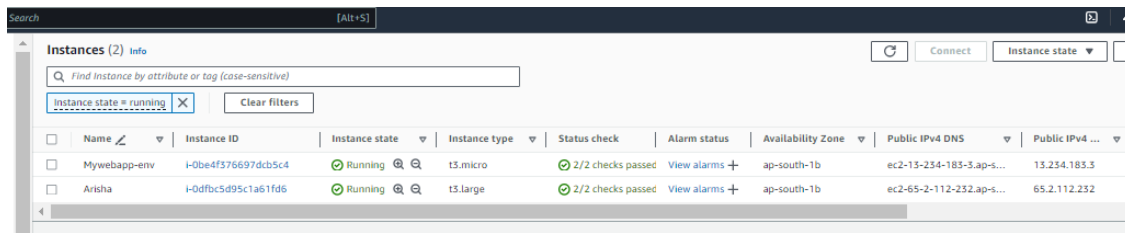
[Elastic Beanstalk](#) > [Environments](#) > Arisha

Arisha [Info](#)

Environment overview

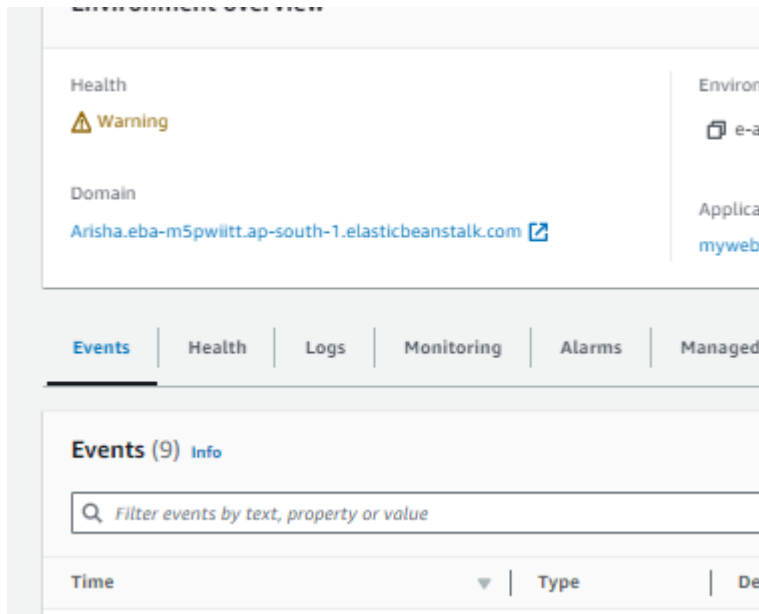
Platform

24. Now you can go to instance from my service you can see The instance is running.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	Mywebapp-env	i-0be4f376697dcb5c4	Running	t3.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-234-183-3.ap-s...	13.234.183.3
<input type="checkbox"/>	Arisha	i-0dfbc5d95c1a61fd6	Running	t3.large	2/2 checks passed	View alarms +	ap-south-1b	ec2-65-2-112-232.ap-s...	65.2.112.232

25. Click on domain



Health

Warning

Domain

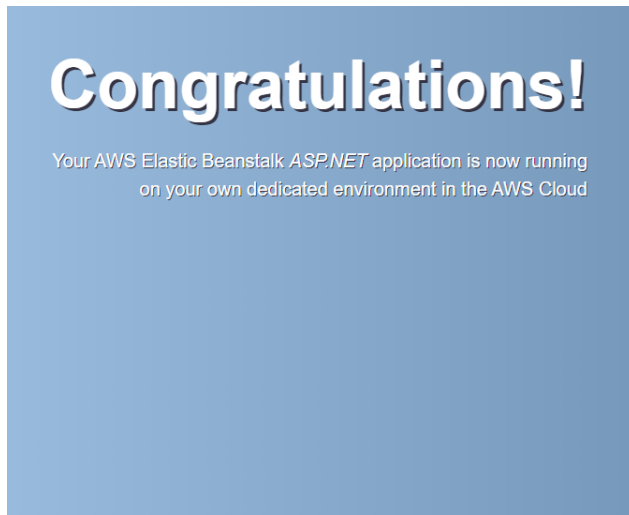
Arisha.eba-m5pwiitt.ap-south-1.elasticbeanstalk.com

Events (9)

Filter events by text, property or value

Time Type Del

26. It will get open into another browser now Elastic Beanstalk Server is running on your own dedicated environment.



# Congratulations!

Your AWS Elastic Beanstalk *ASP.NET* application is now running on your own dedicated environment in the AWS Cloud

## What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploying Applications in .NET Using AWS Toolkit for Visual Studio](#)
- [Managing .NET Environment Settings](#)
- [Working with Logs](#)

## AWS SDK for .NET

- [AWS SDK for .NET home](#)
- [AWS Toolkit for Visual Studio home](#)
- [Windows and .NET developer center](#)
- [AWS SDK for .NET documentation](#)
- [AWS SDK for .NET on GitHub](#)

## AWS .NET Services

- [Generate test events for AWS X-Ray Service](#)

## For Java

### 1.) Sign In to your aws acc

#### Sign in

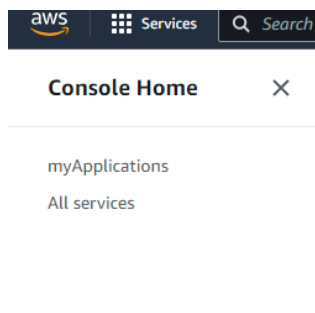
☒ **Root user**  
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**  
User within an account that performs daily tasks. [Learn more](#)

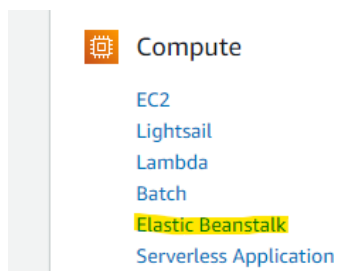
**Root user email address**

**Next**

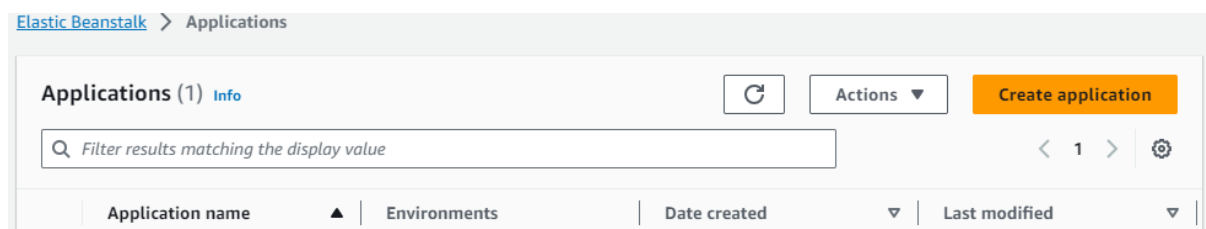
### 2.) Select all services



### 3.) Select Elastic Beanstalk



### 4.) Click on Create Application



## 5.) Enter application name and description and then click on create

**Create new application** [Info](#)

**Application information**

Application name

JAVA

Maximum length of 100 characters.

Description

my first demo

**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.

Add new tag

You can add 50 more tags.

Cancel

Create

## 6. Now create environment

[Elastic Beanstalk](#) > [Applications](#) > MyWebApp

**Application MyWebApp environments (1)** [Info](#)

Filter environments

< 1 > ⚙

Environ...

▲

Health

▼

Date cre...

▼

Domain

▼

Running ...

▼

Platform

▼

Pla

7. For Configure environment everything will be default except platform since we are doing for Java we will choose .Net on Windows server Platform branch and platform version will be default. Click on next.

**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](#)

**Application information** [Info](#)

Application name

JAVA

Maximum length of 100 characters.

**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

Java

▼

Platform branch

Corretto 21 running on 64bit Amazon Linux 2023

▼

Platform version

4.2.0 (Recommended)

▼

8.) For Service access you have to create a role since it is by default over here you can create your own role .

**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.

aws-elasticbeanstalk-service-role

**EC2 key pair**

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

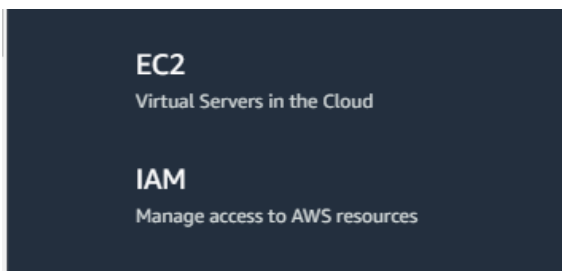
**EC2 instance profile**

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

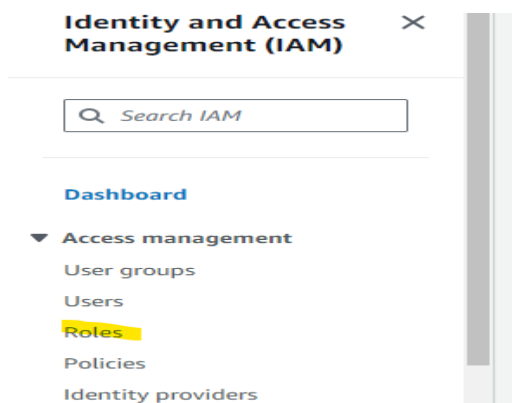
View permission details

Cancel Skip to review Previous Next

9.)To create role go to services IAM



10. Inside IAM dashboard go to roles



11. Create role

**Roles (5)** [Info](#) [Refresh](#) [Delete](#) [Create role](#)

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.

Search

<input type="checkbox"/>	Role name	Trusted entities	Last activity
--------------------------	-----------	------------------	---------------

12.It will ask for entity select AWS Service and for use case select EC2 and click on next

### 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 account to you or a 3rd party to perform ac 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**  
Allows users federated with SAML 2 corporate directory to perform acti 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**

13.Search elasticbeanstalk on top select this three opt and click on next

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkCustomActionMetrics</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkEnhancedHealth</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkManagedUpdatesCustom</a>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkMulticontainerDocker</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkReadOnly</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCore</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCWL</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleECS</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleRDS</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleSNS</a>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleWorkerTier</a>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkWebTier</a>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkWorkerTier</a>

14.Give role name and click on create role

## Name, review, and create

### Role details

**Role name**  
Enter a meaningful name to identify this role.

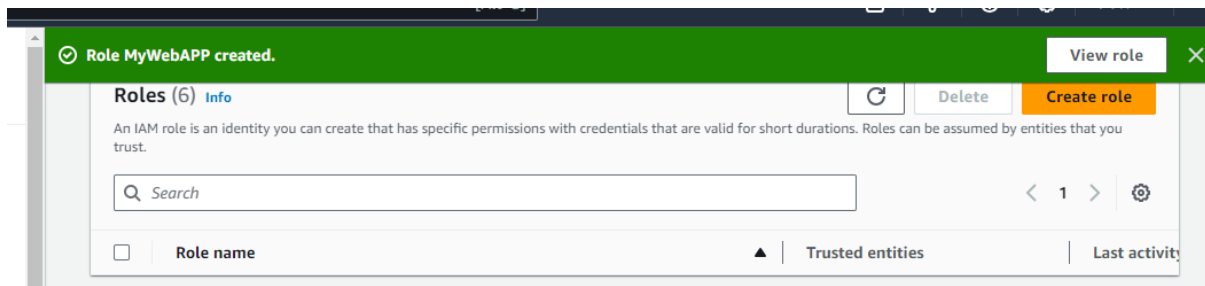
Java

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

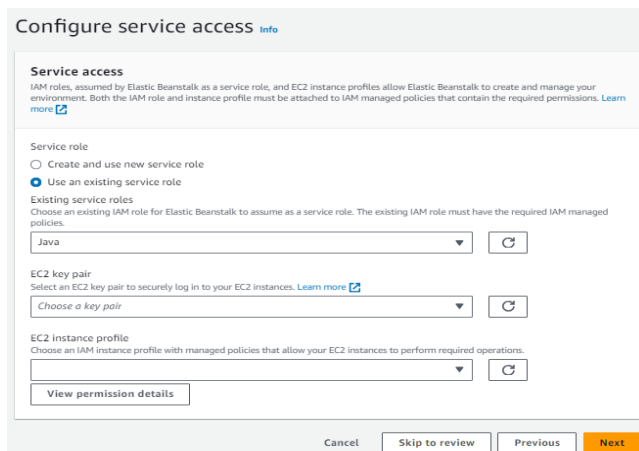
**Description**  
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

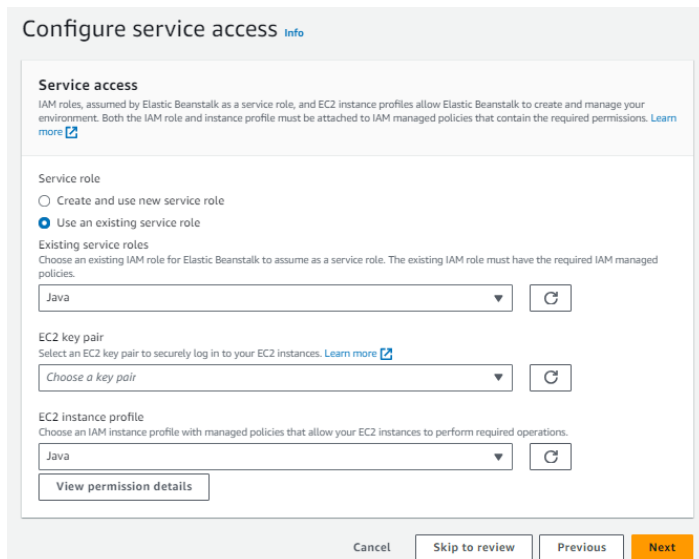
## 15.Role will get created



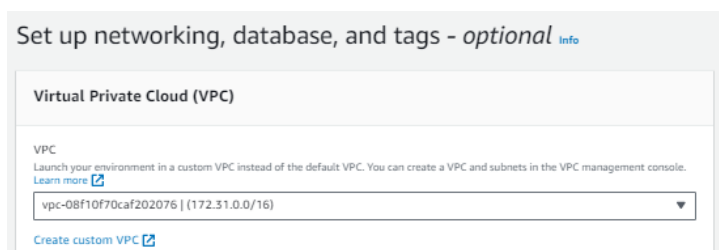
16.Now you can again go to your service access that is your 8 step now inside existing service roles refresh it and click on dropdown you can see over there your role name which you have created above automatically comes there click on that



17.For instance profile click on dropdown your role name will come select that then enter next



18.After clicking on next it will ask for VPC click on dropdown and select which is default not down the ip address of this





19.The ip address you have selected above same over here select that ip address and click on next

### 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**

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-0099214b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-058970f16...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07fe5d047...	172.31.16.0/20	

**Database** [info](#)

Integrate an RDS SQL database with your environment. [Learn more](#)

**Database subnets**  
If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. [Learn more](#)

**Choose database subnets (3)**

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-0099214b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-058970f16...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07fe5d047...	172.31.16.0/20	

20.for configure instance everything will be by default enter next

### Configure instance traffic and scaling - optional [info](#)

**Instances** [info](#)  
Configure the Amazon EC2 instances that run your application.

**Root volume (boot device)**

Root volume type  
(Container default)

Size  
The number of gigabytes of the root volume attached to each instance.  
 GB

IOPS  
Input/output operations per second for a provisioned IOPS (SSD) volume.  
 IOPS

Throughput  
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance  
 MIB/s

**Amazon CloudWatch monitoring**  
The time interval between when metrics are reported from the EC2 instances

Monitoring interval

**Instance metadata service (IMDS)**  
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

IMDSv1

21. For configure update also everything will be default click to next

## Configure updates, monitoring, and logging - *optional* [Info](#)

▼ Monitoring [Info](#)

### Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#) [↗](#)

System

☐ Basic

☒ Enhanced

CloudWatch Custom Metrics - Instance

Choose metrics ▼

22. Review will display click on submit

## Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	MyWebApp
Environment name	Application code
MyWebApp-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1:platform/java:10.0 running on 64bit Windows Server 2019/2.13.2	

Step 2: Configure service access [Edit](#)

Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::836706723086:role/MyWebAPP	MyWebAPP

Step 3: Set up networking, database, and tags [Edit](#)

23. You can see it is getting launch

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

Elastic Beanstalk > Environments > JAVA-env

JAVA-env [Info](#)

Environment overview

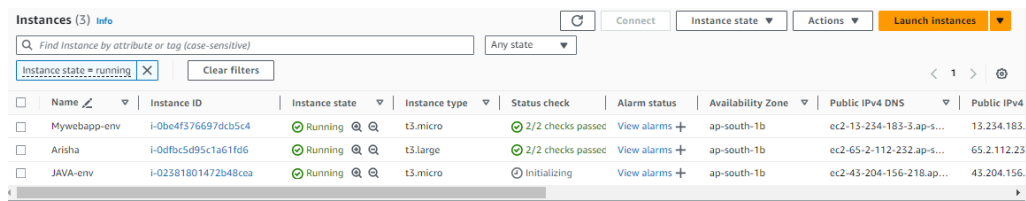
Health	Environment ID
Unknown	e-kj2ix5xkpg
Domain	Application name
-	java

Platform [Change version](#)

Platform	
Corretto 21 running on 64bit Amazon Linux 2023/4.2.0	
Running version	Platform state
-	Supported

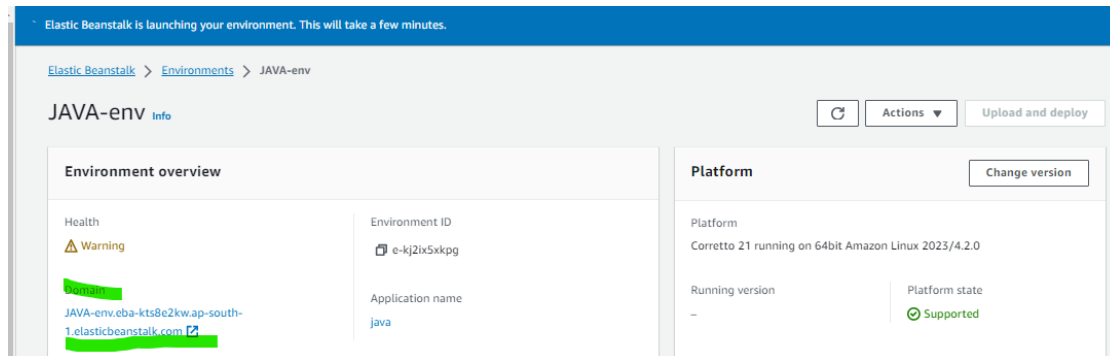
Upload and deploy

24. Now you can go to EC2 from my service you can see The instance is running.



<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
<input type="checkbox"/>	Mywebapp-env	i-0be4f376697dc5c4	Running	t3.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-234-183-3.ap-s...	13.234.183.
<input type="checkbox"/>	Arisha	i-0dfbc5d95c1a61fd6	Running	t3.large	2/2 checks passed	View alarms +	ap-south-1b	ec2-65-2-112-232.ap-s...	65.2.112.23
<input checked="" type="checkbox"/>	JAVA-env	i-02381801472b48cea	Running	t3.micro	Initializing	View alarms +	ap-south-1b	ec2-43-204-156-218.ap...	43.204.156.

25. Click on domain



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

Elastic Beanstalk > Environments > JAVA-env

### JAVA-env

Environment overview

Health: Warning

Environment ID: e-kj2ix5xkpg

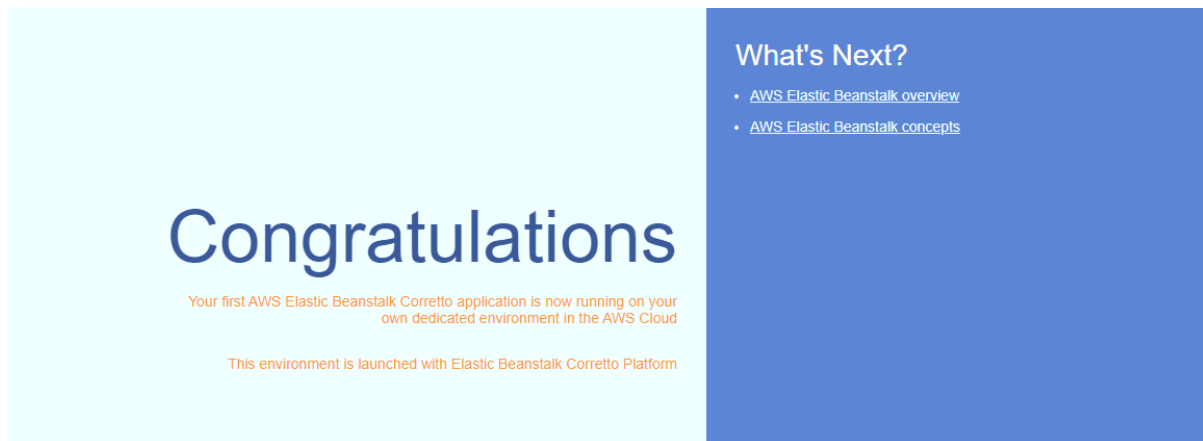
Application name: java

Platform: Corretto 21 running on 64bit Amazon Linux 2023/4.2.0

Running version: -

Platform state: Supported

26. It will get open into another browser now Elastic Beanstalk Server is running on your own dedicated environment.



# Congratulations

Your first AWS Elastic Beanstalk Corretto application is now running on your own dedicated environment in the AWS Cloud

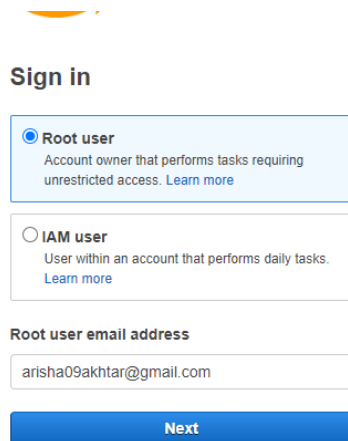
This environment is launched with Elastic Beanstalk Corretto Platform

## What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)

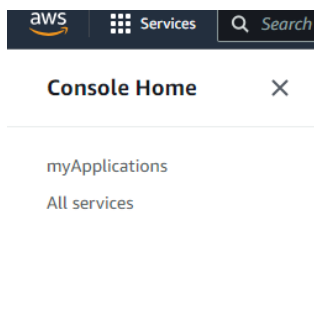
## For Python

1.) Sign In to your aws acc

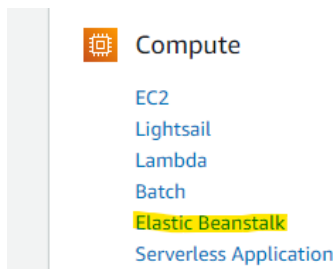


The screenshot shows the AWS Sign in page. At the top, there is a 'Sign in' heading. Below it, there are two radio button options: 'Root user' (selected) and 'IAM user'. The 'Root user' option has a description: 'Account owner that performs tasks requiring unrestricted access. [Learn more](#)'. The 'IAM user' option has a description: 'User within an account that performs daily tasks. [Learn more](#)'. Below these options, there is a label 'Root user email address' and a text input field containing 'arisha09akhtar@gmail.com'. At the bottom, there is a blue 'Next' button.

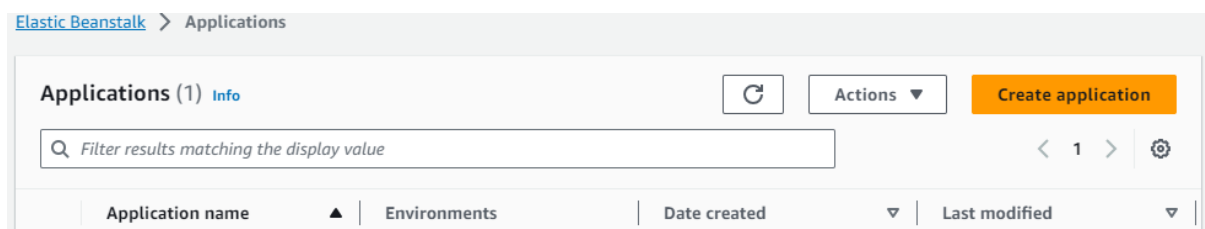
2.) Select all services



3.) Select Elastic Beanstalk



4.) Click on Create Application



## 5.) Enter application name and description and then click on create

**Create new application** [Info](#)

**Application information**

Application name

Maximum length of 100 characters.

Description

**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.

Add new tag

You can add 50 more tags.

Cancel

Create

## 6. Now create environment

[Elastic Beanstalk](#) > [Applications](#) > MyWebApp

**Application MyWebApp environments (1)** [Info](#)

Filter environments

< 1 > ⚙

Environ...

Health

Date cre...

Domain

Running ...

Platform

Pla

## 7. For Configure environment everything will be default except platform since we are doing for python we will choose python Platform branch and platform version will be default. Click on next.

**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

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

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.

**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

Platform branch

Platform version

**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.

8.) For Service access you have to create a role since it is by default over here you can create your own role .

**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.

aws-elasticbeanstalk-service-role

**EC2 key pair**

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair

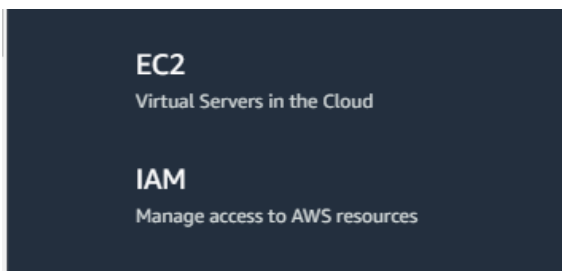
**EC2 instance profile**

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

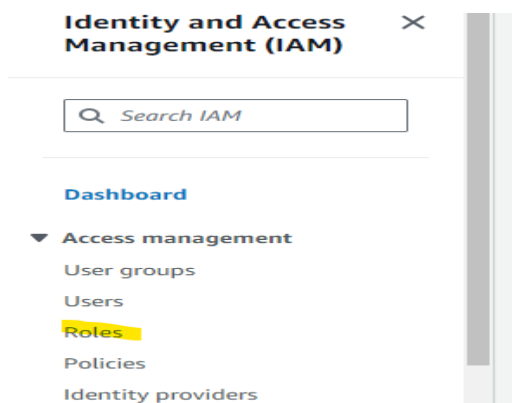
View permission details

Cancel Skip to review Previous Next

9.)To create role go to services IAM



10. Inside IAM dashboard go to roles



11. Create role

**Roles (5)** [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.

Search

1

<input type="checkbox"/>	Role name	Trusted entities	Last activity
--------------------------	-----------	------------------	---------------

12. It will ask for entity select AWS Service and for use case select EC2 and click on next

The screenshot shows the 'Select trusted entity' and 'Use case' steps in the AWS IAM console. The 'Select trusted entity' section has a title bar with 'Select trusted entity' and an 'Info' link. Below it is a 'Trusted entity type' section with five radio button options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Each option has a brief description. The 'Use case' section on the right has a title bar with 'Use case' and a subtitle 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' It features a 'Service or use case' dropdown menu with 'EC2' selected and a 'Choose a use case for the specified service.' section with a 'Use case' radio button for 'EC2' selected.

13. Search elasticbeanstalk on top select this three opt and click on next

The screenshot shows a list of search results for 'elasticbeanstalk'. The results are displayed in a table with columns for a checkbox, an icon, and a link. The links are: 'AWS\_ElasticBeanstalk\_RoleCore', 'AWS\_ElasticBeanstalk\_EnhancedHealth', 'AWS\_ElasticBeanstalk\_ManagedUpdatesCustom', 'AWS\_ElasticBeanstalk\_MulticontainerDocker', 'AWS\_ElasticBeanstalk\_ReadOnly', 'AWS\_ElasticBeanstalk\_RoleCore', 'AWS\_ElasticBeanstalk\_RoleCWL', 'AWS\_ElasticBeanstalk\_RoleECS', 'AWS\_ElasticBeanstalk\_RoleRDS', 'AWS\_ElasticBeanstalk\_RoleSNS', 'AWS\_ElasticBeanstalk\_RoleWorkerTier', 'AWS\_ElasticBeanstalk\_WebTier', and 'AWS\_ElasticBeanstalk\_WorkerTier'. The first three results are selected with checkboxes.

14. Give role name and click on create role

The screenshot shows the 'Name, review, and create' step in the AWS IAM console. It has a title bar with 'Name, review, and create'. Below it is a 'Role details' section with two input fields: 'Role name' and 'Description'. The 'Role name' field contains 'PYTHON' and has a subtitle 'Enter a meaningful name to identify this role.' and a note 'Maximum 64 characters. Use alphanumeric and '+=, @, \_' characters.' The 'Description' field contains 'Allows EC2 instances to call AWS services on your behalf.' and has a subtitle 'Add a short explanation for this role.' and a note 'Maximum 1000 characters. Use alphanumeric and '+=, @, \_' characters.'

15. Role will get created

The screenshot shows the AWS IAM console interface. At the top, a green banner displays the message 'Role PYTHON created.' with a 'View role' button. Below the banner, the 'IAM > Roles' breadcrumb is visible. The main content area shows 'Roles (9)' with an 'Info' link and a subtitle '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.' There is a search bar and a 'Create role' button.

16. Now you can again go to your service access that is your 8 step now inside existing service roles refresh it and click on dropdown you can see over there your role name which you have created above automatically comes there click on that

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.

PYTHON

**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.

17. For instance profile click on dropdown your role name will come select that then enter next

**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.

PYTHON

18. After clicking on next it will ask for VPC click on dropdown and select which is default not down the ip address of this

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-08f10f70caf202076 | (172.31.0.0/16)

[Create custom VPC](#)

19. The ip address you have selected above same over here select that ip address and click on next



**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**

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-0099214b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-058970f16...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07fe5d047...	172.31.16.0/20	

**Database** [info](#)

Integrate an RDS SQL database with your environment. [Learn more](#)

**Database subnets**

If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. [Learn more](#)

**Choose database subnets (3)**

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-0099214b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-058970f16...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-07fe5d047...	172.31.16.0/20	

20.for configure instance everything will be by default enter next

**Configure instance traffic and scaling - optional** [info](#)

**Instances** [info](#)

Configure the Amazon EC2 instances that run your application.

**Root volume (boot device)**

Root volume type

(Container default) ▼

Size

The number of gigabytes of the root volume attached to each instance.

8 GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

100 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

125 MB/s

**Amazon CloudWatch monitoring**

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

5 minute ▼

**Instance metadata service (IMDS)**

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

IMDSv1

21.For configure update also everthing will be default click to next

## Configure updates, monitoring, and logging - *optional* [Info](#)

▼ **Monitoring** [Info](#)

### Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#).

System

☐ Basic
 ☒ Enhanced

CloudWatch Custom Metrics - Instance

Choose metrics ▼

22. Review will display click on submit

## Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	PYTHON
Environment name	Application code
PYTHON-env	Sample application
Platform	
armawselasticbeanstalk:ap-south-1:platform/Python	
3.11 running on 64bit Amazon Linux 2023/4.0.8	

Step 2: Configure service access [Edit](#)

Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::836706723086:role/PYTHON	PYTHON

23. You can see it is getting launch

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

Elastic Beanstalk > Environments > PYTHON-env

## PYTHON-env [Info](#)

[Refresh](#)
[Actions](#)
[Upload and deploy](#)

Environment overview

Health	Environment ID
Unknown	e-2mu2g6icky
Domain	Application name
-	PYTHON

Platform [Change version](#)

Platform	Python 3.11 running on 64bit Amazon Linux 2023/4.0.8
Running version	Platform state
-	Supported

24. Now you can go to EC2 from my service you can see The instance is running.

**Instances (4)** [Info](#)

[Refresh](#)
[Connect](#)
[Instance state](#)
[Actions](#)
[Launch instances](#)

Find Instance by attribute or tag (case-sensitive)

Any state

Instance state = running

Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elas
<input type="checkbox"/>	Mywebapp-env	i-0be4f376697dcb5c4	Running	t3.micro	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1b	ec2-13-234-183-3.ap-s...	13.234.183.3	13.2
<input type="checkbox"/>	Arisha	i-0dfbc5d95c1a61fd6	Running	t3.large	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1b	ec2-65-2-112-232.ap-s...	65.2.112.232	65.2
<input type="checkbox"/>	JAVA-env	i-02381801472b48coa	Running	t3.micro	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1b	ec2-43-204-156-218.ap...	43.204.156.218	43.2
<input type="checkbox"/>	PYTHON-env	i-0458f92852efb5a91	Running	t3.micro	Initializing	<a href="#">View alarms</a>	ap-south-1b	-	-	-

25. Click on domain

The screenshot shows the AWS Elastic Beanstalk console for the environment 'PYTHON-env'. The 'Environment overview' section displays the health status as 'Pending', the environment ID as 'e-2mu2g6icky', the domain as 'PYTHON-env.eba-kweqws3d.ap-south-1.elasticbeanstalk.com', and the application name as 'PYTHON'. The 'Platform' section shows 'Python 3.11 running on 64bit Amazon Linux 2023/4.0.8' and a 'Supported' platform state.

26. It will get open into another browser now Elastic Beanstalk Server is running on your own dedicated environment.

The screenshot shows a 'Congratulations' message: 'Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud'. Below this, it states 'This environment is launched with Elastic Beanstalk Python Platform'. To the right, under 'What's Next?', there are several links: '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', and 'Working with Logs'.

27. Last step don't forget to terminate the instances that have been created

The screenshot shows the AWS Management Console 'Instances' page. It lists four instances: 'Mywebapp-env' (t3.micro, Running), 'Arisha' (t3.large, Running), 'JAVA-env' (t3.micro, Running), and 'PYTHON-env' (t3.micro, Initializing). The 'PYTHON-env' instance is highlighted, and a context menu is open showing options like 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', and 'Terminate instance'.