

# Cloud Computing

## Prac 2: Platform as a service using AWS.

### Writeup

- Platform as a service:

Platform as a Service (PaaS) provides a runtime environment. It allows programmers to easily create, test, run, and deploy web applications. You can purchase these applications from a cloud service provider on a pay-as-per use basis and access them using the Internet connection. In PaaS, back end scalability is managed by the cloud service provider, so end- users do not need to worry about managing the infrastructure.

PaaS includes infrastructure (servers, storage, and networking) and platform (middleware, development tools, database management systems, business intelligence, and more) to support the web application life cycle.

Example: Google App Engine, Force.com, Joyent, Azure.

- Elastic beanstalk:

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.

This raises a concern that if elastic Beanstalk configures most of the resources itself and abstracts the underlying details. Can developers change the configuration if needed? The answer is Yes. Elastic Beanstalk is provided to make application deployment simpler but at no level will it restrict the developers from changing any configurations.

- Components of Beanstalk:

### **1. Deployment**

Elastic Beanstalk, for the most part, simplifies the process of deploying an application on the Amazon cloud. The service allows developers the ability to upload and manage different versions of their apps, and switch between them in different environments like development, test, and production.

### **2. Application**

An application in Elastic Beanstalk is basically a collection of environments, versions, and everything else related to them, like events. In other words, an Elastic Beanstalk application is conceptually similar to a folder. Most users normally create a separate EB application for each of their applications, and although this is not required, it does help streamline management.

### **3. Version**

A version is the deployable code of an application. Depending on your programming platform of choice, you will have a file, or a set of files that you upload, with a label and description. You can then see where it is deployed, in which environment, and even download the file or files, if needed.

### **4. Environment**

As you may have guessed, an environment is a deployed version of specific instances, load balancers, and scaling groups, etc. A typical workflow is creating one environment for testing, and another for production. Though you can, of course, create as many as you need, as much as your budget allows. Amazon provides access to your environment via a specific URL, and provides different health status so you can quickly get an idea of how things are up there. Green is okay, yellow when your environment has not responded within the last 5 minutes, red if it hasn't responded for more than 5 minutes, and grey, unknown.

### **5. Events**

Events tell you what is going on with your environment. They are either informative, warnings, or errors, letting you know details like when an environment successfully launches, or an instance is close to utilising its resources. You can view the events in a web console, or have them sent to you via email.

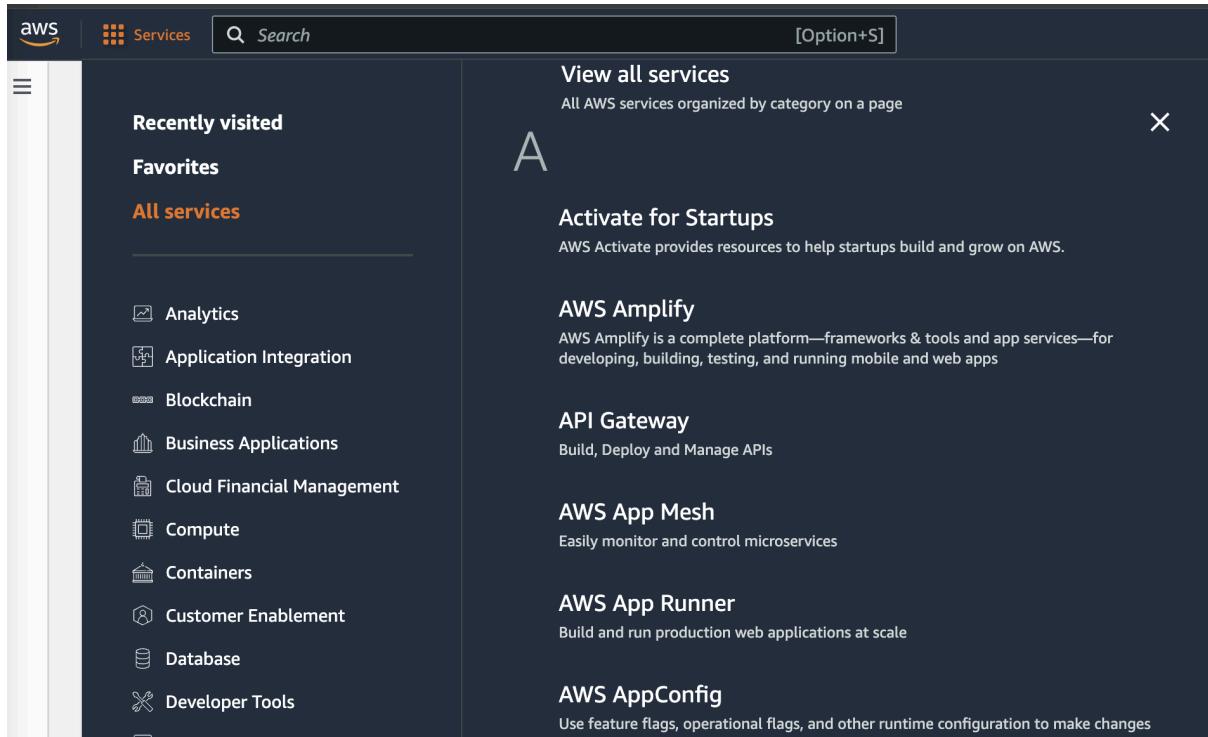
- IAM:

IAM is a combination of policies and technologies that allows organisations to identify users and provide the right form of access as and when required. There has been a burst in the market with new applications, and the requirement for an organisation to use these applications has increased drastically. The services and resources you want to access can be specified in IAM. IAM doesn't provide any replica or backup. IAM can be used for many purposes such as, if one wants to control access of individual and group access for your AWS resources. With IAM policies, managing permissions to your workforce and systems to ensure least-privilege permissions becomes easier. The AWS IAM is a global service.

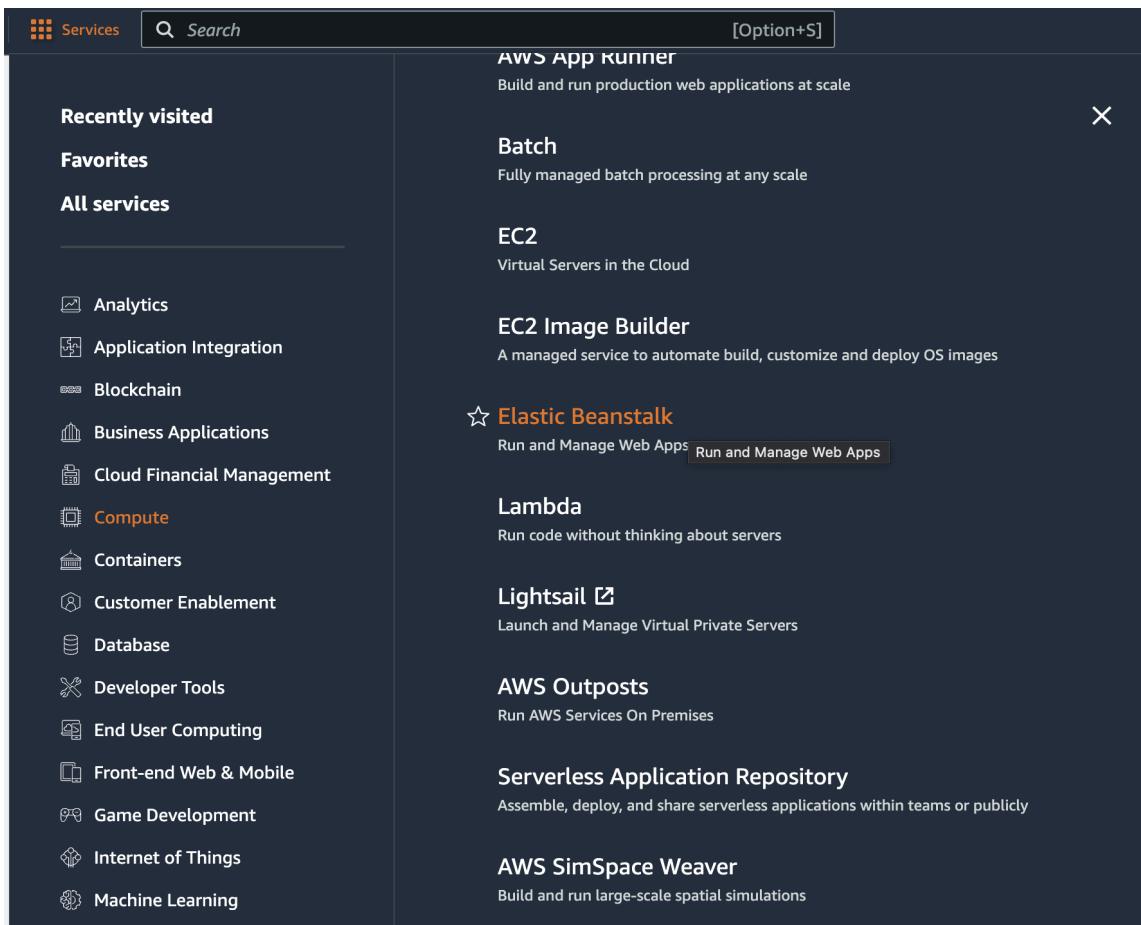
## Implement PaaS using elastic beanstalk for Python

1. Sign in to your AWS account.

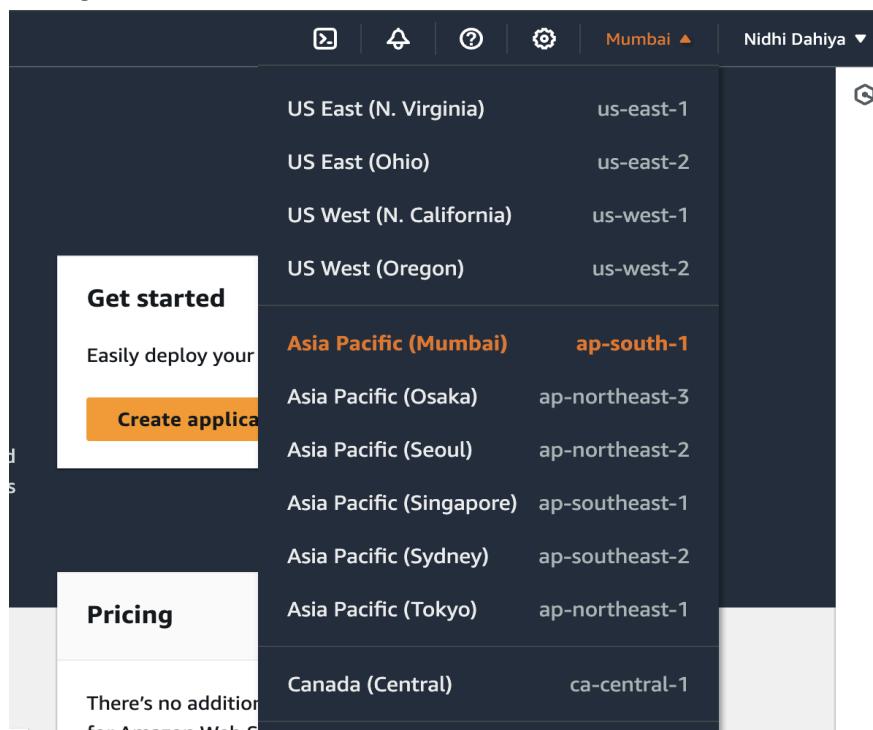
2. Select All Services.



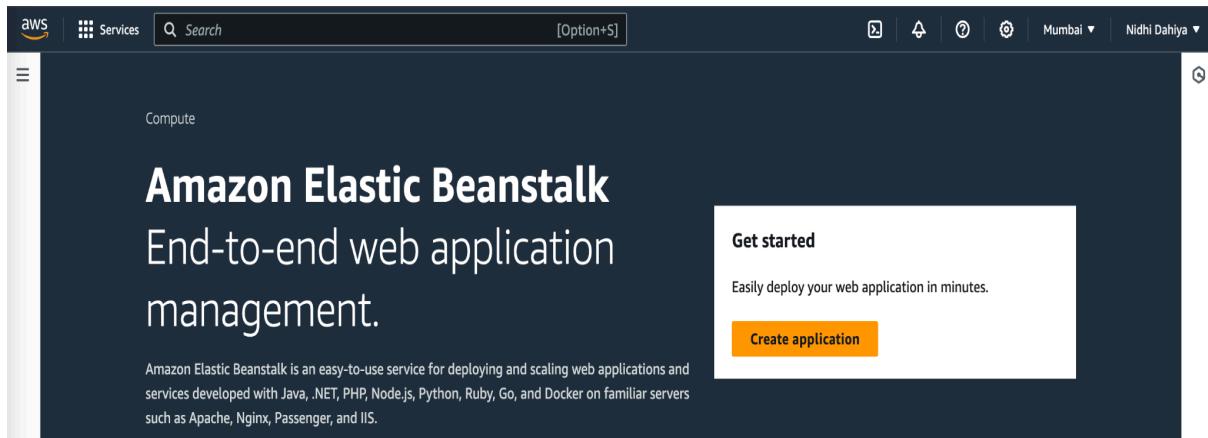
### 3. Select Elastic Beanstalk.



### 4. Select region Mumbai.



5. Click on create application.



6. Click on the web server environment.

A screenshot of the 'Configure environment' step in the AWS Elastic Beanstalk wizard. The title 'Configure environment' is at the top, followed by a link 'Info'. Below it, a section titled 'Environment tier' has a 'Info' link. It states that Amazon Elastic Beanstalk has two types of environment tiers: 'Web server environment' and 'Worker environment'. The 'Web server environment' is selected, indicated by a blue radio button. A description follows: 'Run a website, web application, or web API that serves HTTP requests.' There is a 'Learn more' link with a help icon. The 'Worker environment' option is also listed with its description: 'Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.' Another 'Learn more' link is provided for this option.

7. Give the application name and description. (give the domain or else it'll be given by default)

**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.

Domain

.ap-south-1.elasticbeanstalk.com
 Check availability

Environment description

My sample application to demonstrate PaaS.

8. Choose the platform i.e, Python.

**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

4.0.8 (Recommended)

9. Select sample application (no code to upload), and single instance.

**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**

10. Click on next.

11. Click on create a new service role.

**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

**Service role name**

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

**Cancel** **Next Step**

12. For creating a new role, Go to IAM in ALL SERVICES. (use a new tab)

The screenshot shows the AWS Services dashboard. On the left, there's a sidebar with 'Recently visited' (Amazon Honeycode), 'Favorites' (empty), and 'All services'. Below that is a list of various AWS services with icons: Analytics, Application Integration, Blockchain, Business Applications, Cloud Financial Management, Compute, Containers, Customer Enablement, Database, Developer Tools, End User Computing, Front-end Web & Mobile, and Game Development. On the right, the 'IAM' service is highlighted with a star icon and the text 'Manage access to AWS resources'. Other services listed include Amazon Honeycode, IAM Identity Center, Incident Manager, Amazon Inspector, Amazon Interactive Video Service, and IoT 1-Click.

13. Click on roles and then create role.

The screenshot shows the 'Identity and Access Management (IAM)' service page. The left sidebar has sections for 'Access management' (User groups, Users, Roles), 'Roles' (Policies, Identity providers, Account settings), and 'Dashboard'. The main area is titled 'IAM > Roles' and shows a table of roles. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'. Two roles are listed: 'AWSServiceRoleForSupport' (AWS Service: support) and 'AWSServiceRoleForTrustedAdvisor' (AWS Service: trustedadvisor). A 'Create role' button is located at the top right of the table. Below the table is a section titled 'Roles Anywhere' with a 'Manage' button.

## 14. Click on AWS service.

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

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**  
Allows 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.

## 15. Select EC2 and then next.

**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**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**  
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- EC2 - Scheduled Instances**  
Allows EC2 Scheduled Instances to manage instances on your behalf.

**Cancel** **Next**

16. Search for beanstalk rules, and select following 3 and next.

Filter by Type			
<input type="text"/> beanstalk	<input type="button"/>	All types	14 matches
	Policy name	Type	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSEI...</a>	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkCustom...</a>	AWS managed	Provide the instance in your custom pl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkEnhanc...</a>	AWS managed	AWS Elastic Beanstalk Service policy f...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkManag...</a>	AWS managed	This policy is for the AWS Elastic Bean...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkMultico...</a>	AWS managed	Provide the instances in your multicon...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkReadOnly</a>	AWS managed	Grants read-only permissions. Explicitl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCore</a>	AWS managed	AWSElasticBeanstalkRoleCore (Elastic ...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCWL</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleECS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleRDS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleSNS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleWo...</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Provide the instances in your web serv...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWorker...</a>	AWS managed	Provide the instances in your worker e...

17. Name the role.

### Name, review, and create

<b>Role details</b>
<p><b>Role name</b> Enter a meaningful name to identify this role. <input type="text" value="beanstalk-role"/> Maximum 64 characters. Use alphanumeric and '+,-,@-' characters.</p> <p><b>Description</b> Add a short explanation for this role. <input type="text" value="Allows EC2 instances to call AWS services on your behalf."/> Maximum 1000 characters. Use alphanumeric and '+,-,@-' characters.</p>
<b>Step 1: Select trusted entities</b>
<p><b>Trust policy</b></p> <pre> 1  "Version": "2012-10-17", 2  "Statement": [ 3    { 4      "Effect": "Allow", 5      "Action": [ 6        ... </pre>

18. Click on create role.

The screenshot shows the 'Step 2: Add permissions' section of the AWS IAM 'Create New Role' wizard. At the top, there's a table titled 'Permissions policy summary' with three rows:

Policy name	Type	Attached as
<a href="#">AWSElasticBeanstalkMulticontainerDocker</a>	AWS managed	Permissions policy
<a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Permissions policy

Below the table, there's a section titled 'Step 3: Add tags' with a note: 'Add tags - optional' and a message: 'Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.' It shows 'No tags associated with the resource.' and a button 'Add new tag'. A note below says 'You can add up to 50 more tags.' At the bottom right are buttons for 'Cancel', 'Previous', and 'Create role' (which is highlighted).

19. Role created.

The screenshot shows the 'Roles' list screen in the AWS IAM console. At the top, a green banner says 'Role beanstalk-role created.' with buttons for 'View role' and 'X'. Below it, the 'IAM > Roles' navigation path is shown. The main table has columns: 'Role name', 'Trusted entities', and 'Last activity'. It lists three roles:

Role name	Trusted entities	Last activity
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linked)	-
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linked)	-
<a href="#">beanstalk-role</a>	AWS Service: ec2	-

20. Go to EC2, click on the key pairs and create a key pair.

The screenshot shows the AWS Management Console navigation bar on the left, with 'Key pairs' selected under 'Network & Security'. The main area is titled 'Key pairs' with an 'Info' link. It features a search bar labeled 'Find Key Pair by attribute or tag'. Below the search bar is a table header with columns: Name, Type, Created, Fingerprint, and ID. A message 'No key pairs to display' is centered below the table. At the top right of the main area are 'Actions' and 'Create key pair' buttons.

21. Create a key pair.

The screenshot shows the 'Create key pair' wizard. The first step, 'Key pair', is completed. The second step, 'Name', has 'sample-key-pair' entered. The third step, 'Key pair type', has 'RSA' selected. The fourth step, 'Private key file format', has '.ppk' selected. The fifth step, 'Tags - optional', shows a button 'Add new tag' and a note about adding up to 50 more tags. The final step, 'Create key pair', is at the bottom right.

22. Go to the previous tab, and refresh and select the role and key pair and click on 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

**Service role name**  
 Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.  
 aws-elasticbeanstalk-service-role

[View permission details](#)

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

sample-key-pair ▼

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

beanstalk-role ▼

[View permission details](#)

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

23. Select VPC.

### 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-025f7c4e1bc4e0cba | (172.31.0.0/16) ▼

[Create custom VPC](#)

24. Select the same IP address in instance as of VPC (172.31.0.0).

The screenshot shows the AWS VPC configuration interface. At the top, there is a dropdown menu with the option "vpc-025f7c4e1bc4e0cba | (172.31.0.0/16)". Below it is a link "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**

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1c	subnet-018470deb...	172.31.16.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-040070104...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0dd23cf18...	172.31.32.0/20	

25. Select the same IP address in the database as of VPC.

**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-1c	subnet-018470deb...	172.31.16.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-040070104...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0dd23cf18...	172.31.32.0/20	

Enable database

**Restore a snapshot - optional**

Restore an existing snapshot from a previously used database.

Snapshot

26. Click on next.

27. Set as default, no changes and 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

##### 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

 ▾

28. Set as default, no changes and 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](#)

##### CloudWatch Custom Metrics - Environment

[Choose metrics](#)

##### Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

##### Log streaming

- Activated (standard CloudWatch charges apply.)

##### Retention

7

29. Review.

### Review Info

#### Step 1: Configure environment

[Edit](#)

##### Environment information

Environment tier	Application name
Web server environment	test-web-application
Environment name	Application code
Test-web-application-env	Sample application

Platform  
arn:aws:elasticbeanstalk: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 key pair	EC2 instance profile
arn:aws:iam::544606390347:role/service-role/aws-elasticbeanst	sample-key-pair	beanstalk-role

### 30. Submit.

Command timeout	Deployment policy	Health threshold				
600	AllAtOnce	Ok				
Ignore health check	Instance replacement					
false	false					
<b>Platform software</b>						
Lifecycle	Log streaming	NumProcesses				
false	Deactivated	1				
NumThreads	WSGIPath	Proxy server				
15	application	nginx				
Logs retention	Rotate logs	Update level				
7	Deactivated	minor				
X-Ray enabled						
Deactivated						
<b>Environment properties</b>						
<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>PYTHONPATH</td> <td>/var/app/venv/staging-LQM1lest/bin</td> </tr> </tbody> </table>			Key	Value	PYTHONPATH	/var/app/venv/staging-LQM1lest/bin
Key	Value					
PYTHONPATH	/var/app/venv/staging-LQM1lest/bin					
<a href="#">Cancel</a> <a href="#">Previous</a> <a href="#" style="background-color: orange; color: white; border: 1px solid orange;">Submit</a>						

### 31. Launching.

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

[Elastic Beanstalk](#) > [Environments](#) > Test-web-application-env

**Test-web-application-env** [Info](#) [C](#) [Actions ▾](#) [Upload and deploy](#)

<b>Environment overview</b>	<b>Platform</b> <a href="#">Change version</a>
Health Unknown	Platform Python 3.11 running on 64bit Amazon Linux 2023/4.0.8
Domain -	Running version -
Application name test-web-application	Platform state Supported

[Events](#) [Health](#) [Logs](#) [Monitoring](#) [Alarms](#) [Managed updates](#) [Tags](#)

### 32. Environment is created.

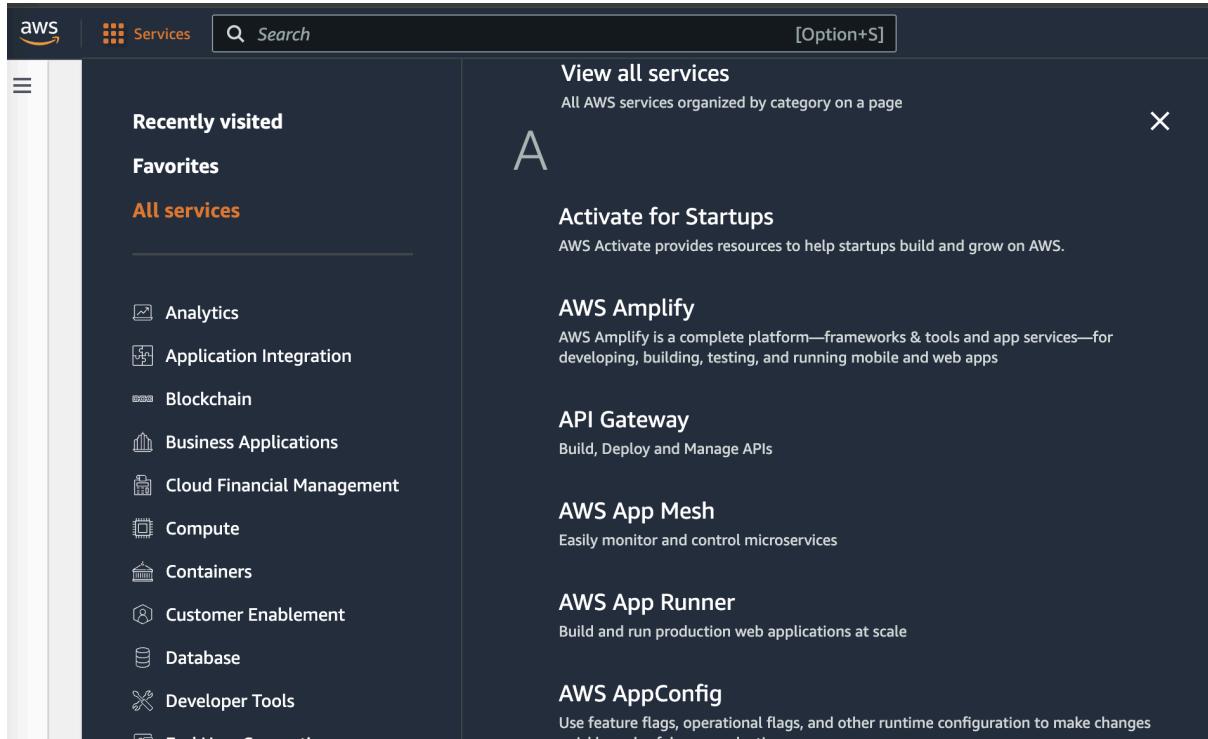
The screenshot shows the AWS Elastic Beanstalk interface. On the left, a sidebar lists 'Applications', 'Environments' (which is selected), and 'Change history'. Below this is a 'Recent environments' section containing 'Test-web-application-env'. The main content area has a green header bar with the text 'Environment successfully launched.' and a circular icon. Below this is a breadcrumb navigation bar with 'Elastic Beanstalk > Environments'. The main content is titled 'Environments (1) Info' with a search bar labeled 'Filter environments'. A table lists one environment: 'Test-web-application-env' with status 'Ok', application 'test-web...', platform 'Python 3...', and domain 'Test-web-application-env.eba...'. There are 'Actions' and 'Create environment' buttons at the top right of the table.

The screenshot shows a 'Congratulations' message on the left side of the screen. It says: 'Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud' and 'This environment is launched with Elastic Beanstalk Python Platform'. To the right, under the heading 'What's Next?', there is a list of 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'.

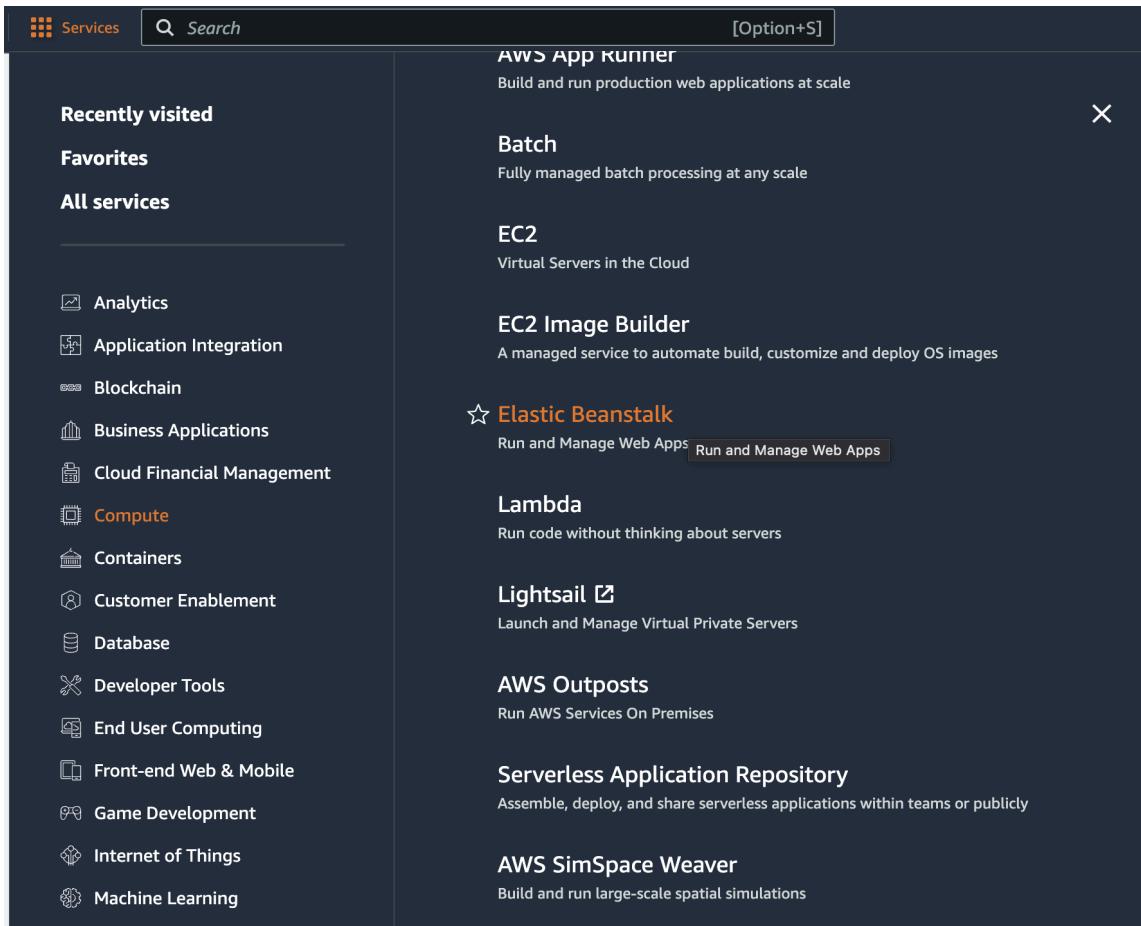
## Implement PaaS using elastic beanstalk for Java

1. Sign in to your AWS account.

2. Select All Services.



### 3. Select Elastic Beanstalk.



### 4. Click on create application.

The screenshot shows the AWS Elastic Beanstalk 'Applications' page. At the top, there's a breadcrumb navigation: 'Elastic Beanstalk > Applications'. Below that is a header with 'Applications (1) Info' and a 'Create application' button. There's a search bar labeled 'Filter results matching the display value'. The main table displays one application entry:

Application name	Environments	Date created	Last modified
<a href="#">test-web-application</a>	<a href="#">Test-web-application-env</a>	January 29, 2024 20:01:40 (...)	January 29, 2024 20:01:40 (...)

## 5. Name the application.

**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**

## 6. Give the description.

**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.

**Domain**  
 .ap-south-1.elasticbeanstalk.com

**Environment description**

**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**

## 7. Choose the platform.

**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. Choose Upload your code and download a java code war file and upload the same.

**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.

Version label

Unique name for this version of your application code.

sample-java

Source code origin. Maximum size 500 MB

Local file

Upload application

 Choose file

 File name: **SampleWebApp.war**

File must be less than 500MB max file size

Public S3 URL

**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)

9. Click on next.

10. Click on create a new service role.

## Configure service access Info

### Service access

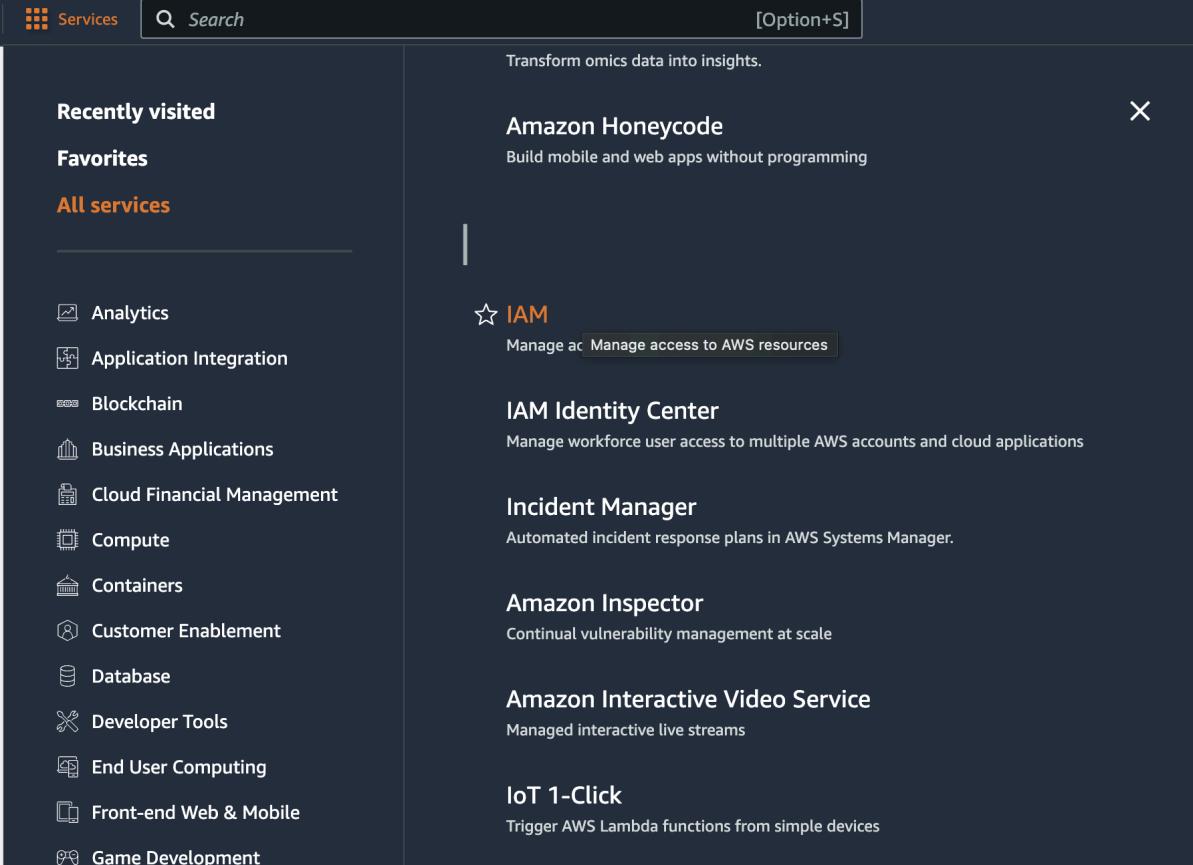
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

#### Service role

- Create and use new service role
- Use an existing service role

[Create role now](#)

11. For creating a new role, Go to IAM in ALL SERVICES. (use a new tab)



The screenshot shows the AWS Services menu interface. On the left, there's a sidebar with 'Recently visited' (Analytics, Application Integration, Blockchain, Business Applications, Cloud Financial Management, Compute, Containers, Customer Enablement, Database, Developer Tools, End User Computing, Front-end Web & Mobile, Game Development), 'Favorites' (Amazon Honeycode, IAM Identity Center, Incident Manager, Amazon Inspector, Amazon Interactive Video Service, IoT 1-Click), and 'All services'. The 'Services' tab is selected. In the center, the 'IAM' service is highlighted with a star icon and the text 'Manage access to AWS resources'. Other services listed include Amazon Honeycode, IAM Identity Center, Incident Manager, Amazon Inspector, Amazon Interactive Video Service, and IoT 1-Click.

12. Click on roles and then create role.

The screenshot shows the AWS IAM 'Roles' page. At the top, there's a breadcrumb navigation from 'IAM' to 'Roles'. Below it, a header says 'Roles (5) Info' with a 'Delete' button and an orange 'Create role' button. A search bar and a page navigation area with a single item are also present. The main content area lists five roles, each with a small icon, a name, and a 'View details' link.

13. Select EC2 and next.

The screenshot shows the 'Create New Role' wizard, Step 1: Set permissions. It has a 'Use case' section with a note about allowing AWS services like EC2, Lambda, or others to perform actions. Below is a 'Service or use case' dropdown set to 'EC2'. The 'Use case' section contains several options, each with a description:

- 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**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**  
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- EC2 - Scheduled Instances**  
Allows EC2 Scheduled Instances to manage instances on your behalf.

At the bottom right are 'Cancel' and 'Next' buttons.

14. Select the following and next.

Filter by Type			
	<input type="text" value="beanstalk"/>	<input type="button" value="X"/>	All types
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSE...</a>	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkCustom...</a>	AWS managed	Provide the instance in your custom pl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkEnhanc...</a>	AWS managed	AWS Elastic Beanstalk Service policy f...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkManag...</a>	AWS managed	This policy is for the AWS Elastic Bean...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkMultico...</a>	AWS managed	Provide the instances in your multicon...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkReadOnly</a>	AWS managed	Grants read-only permissions. Explicitl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCore</a>	AWS managed	AWSElasticBeanstalkRoleCore (Elastic ...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCWL</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleECS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleRDS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleSNS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleWo...</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Provide the instances in your web serv...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWorker...</a>	AWS managed	Provide the instances in your worker e...

15. Name the role and click on create role.

### Name, review, and create

#### Role details

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

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

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

Maximum 1000 characters. Use alphanumeric and '+,-,\_' characters.

#### Step 1: Select trusted entities

##### Trust policy

```

1 - [
2 -   "Version": "2012-10-17",
3 -   "Statement": [
4 -     {
5 -       "Effect": "Allow",
6 -       "Principal": "*"
6 -     }
6 -   ]
6 - ]

```

## 16. Role created.

The screenshot shows the AWS IAM Roles list. At the top, a green header bar displays the message "Role java-role created." and includes "View role" and "X" buttons. Below the header, the navigation path "IAM > Roles" is shown. The main area has a title "Roles (6) Info" with a "C" icon, "Delete" button, and "Create role" button. A search bar and pagination controls are at the top right. A table lists six roles:

Role name	Trusted entities	Last activity
<a href="#">aws-elasticbeanstalk-service-role</a>	AWS Service: elasticbeanstalk	18 minutes ago
<a href="#">AWSServiceRoleForAutoScaling</a>	AWS Service: autoscaling (Service-Linker)	26 minutes ago
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	-
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-
<a href="#">beanstalk-role</a>	AWS Service: ec2	16 minutes ago
<a href="#">java-role</a>	AWS Service: ec2	-

## 17. Go to EC2, click on the key pairs and create a key pair.

The screenshot shows the AWS Key Pairs list. At the top, a green header bar displays the message "Key pairs (1) Info" and includes "Actions" and "Create key pair" buttons. Below the header, the navigation path "Key pairs" is shown. The main area has a title "Key pairs (1) Info" with a "C" icon, "Actions" button, and "Create key pair" button. A search bar and pagination controls are at the top right. A table lists one key pair:

Name	Type	Created	Fingerprint	ID
sample-key-pair	rsa	2024/01/29 20:32 GMT+5:30	42:f0:5a:d4:ec:2d:f1:d0:9e:e6:1...	key-0e769a5a5cd33fee7

## 18. Create a key pair.

The screenshot shows the "Create key pair" wizard. At the top, a green header bar displays the message "Create key pair Info" and includes "Cancel" and "Create key pair" buttons. Below the header, the navigation path "Key pairs" is shown. The main area has a title "Key pair" with a "Info" link. It contains the following fields:

- Name:** say-java-key (input field)
- Key pair type:** RSA (radio button selected)
- Private key file format:** .ppk (radio button selected)
- Tags - optional:** No tags associated with the resource.
- Add new tag:** (button)
- Notes:** You can add up to 50 more tags.

19. Created.

Key pairs (2) <a href="#">Info</a>					
<input type="checkbox"/>	Name	Type	Created	Fingerprint	ID
<input type="checkbox"/>	say-java-key	rsa	2024/01/29 22:22 GMT+5:30	7c:24:57:24:51:73:8d:81:37:7b:6d:ba:2...	key-06cd0...
<input type="checkbox"/>	sample-key-pair	rsa	2024/01/29 20:32 GMT+5:30	42:f0:5a:d4:ec:2d:f1:d0:9e:e6:18:11:e2:...	key-0e769...

20. Go to the previous tab, and refresh and select the role and key pair and click on 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

**Service role name**  
Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

[View permission details](#)

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

[C](#)

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.  
 [C](#)

[View permission details](#)

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

21. Select VPC.

## 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-025f7c4e1bc4e0cba | (172.31.0.0/16) 

[Create custom VPC](#) 

22. Select the same IP address in instance as of VPC (172.31.0.0).

### 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-1c	subnet-018470deb...	172.31.16.0/20	
<input checked="" type="checkbox"/> ap-south-1b	subnet-040070104...	172.31.0.0/20	
<input type="checkbox"/> ap-south-1a	subnet-0dd23cf18...	172.31.32.0/20	

23. Select the same IP address in the database as of VPC.

The screenshot shows the 'Database Info' configuration step in the AWS Elastic Beanstalk console. It displays a table titled 'Choose database subnets (3)' with three rows of subnet information. The columns are 'Availability Zone', 'Subnet', 'CIDR', and 'Name'. The first row (ap-south-1c) has an unchecked checkbox. The second row (ap-south-1b) has a checked checkbox, indicating it is selected. The third row (ap-south-1a) has an unchecked checkbox. Below the table is a 'Enable database' toggle switch, which is turned off. Under the heading 'Restore a snapshot - optional', there is a note about restoring from a previous snapshot and a dropdown menu set to 'None'. At the bottom, under 'Database settings', there is a note about choosing an engine and instance type, and a dropdown menu currently set to 'Engine'.

Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1c	subnet-018470deb...	172.31.16.0/20
<input checked="" type="checkbox"/>	ap-south-1b	subnet-040070104...	172.31.0.0/20
<input type="checkbox"/>	ap-south-1a	subnet-0dd23cf18...	172.31.32.0/20

Enable database

**Restore a snapshot - optional**  
Restore an existing snapshot from a previously used database.

**Snapshot**  
None

**Database settings**  
Choose an engine and instance type for your environment's database.

**Engine**

24. Click on next.

25. Set as default, no changes and 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

MiB/s

#### Amazon CloudWatch monitoring

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

##### Monitoring interval

26. Set as default, no changes and 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* 

##### CloudWatch Custom Metrics - Environment

*Choose metrics* 

#### Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

##### Log streaming

- Activated (standard CloudWatch charges apply.)

##### Retention

7 

## 27. Review.

**Review [Info](#)**

**Step 1: Configure environment** [Edit](#)

Environment information	
Environment tier	Application name
Web server environment	this-java
Environment name	Application code
This-java-env	SampleWebApp.war
Platform	
arn:aws:elasticbeanstalk:ap-south-1::platform/Corretto 21 running on 64bit Amazon Linux 2023/4.2.0	

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

Service access <a href="#">Info</a>		
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 key pair	EC2 instance profile
arn:aws:iam::544606390347:role/service-role/aws-elasticbeanstalk-	say-java-key	java-role

## 28. Submit.

Rotate logs	Update level	X-Ray enabled								
Deactivated	minor	Deactivated								
<b>Environment properties</b>										
<table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>GRADLE_HOME</td> <td>/usr/local/gradle</td> </tr> <tr> <td>M2</td> <td>/usr/local/apache-maven/bin</td> </tr> <tr> <td>M2_HOME</td> <td>/usr/local/apache-maven</td> </tr> </tbody> </table>			Key	Value	GRADLE_HOME	/usr/local/gradle	M2	/usr/local/apache-maven/bin	M2_HOME	/usr/local/apache-maven
Key	Value									
GRADLE_HOME	/usr/local/gradle									
M2	/usr/local/apache-maven/bin									
M2_HOME	/usr/local/apache-maven									

[Cancel](#) [Previous](#) **Submit**

## 29. Launching.

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

[Elastic Beanstalk](#) > [Environments](#) > This-java-env

### This-java-env [Info](#)

[C](#) [Actions ▾](#) [Upload and deploy](#)

Environment overview		Platform	<a href="#">Change version</a>
Health	Environment ID	Platform	
(⌚ Pending)	e-pifn2ksbp7	Corretto 21 running on 64bit Amazon Linux 2023/4.2.0	
Domain	Application name	Running version	-
-	this-java	Platform state	<span style="color: green;">Supported</span>

[Events](#) [Health](#) [Logs](#) [Monitoring](#) [Alarms](#) [Managed updates](#) [Tags](#)

## 30. Environment is created.

[Applications \(2\) \[Info\]\(#\)](#)

[C](#) [Actions ▾](#) [Create application](#)

Filter results matching the display value

Application name	Environments	Date created	Last modified
<a href="#">test-web-application</a>	<a href="#">Test-web-application-env</a>	January 29, 2024 20:01:40 (...)	January 29, 2024 20:01:40 (...)
<a href="#">this-java</a>	<a href="#">This-java-env</a>	January 29, 2024 22:04:47 (...)	January 29, 2024 22:04:47 (...)