

Practical-5 Platform as a service using AWS

Name: Ruchika G Karoo

Roll no: A029

1)Platform as a Service (PaaS)

Platform as a Service (PaaS) is a cloud computing model that provides a ready-to-use environment for developers to build, run, and manage applications without worrying about the underlying infrastructure. PaaS offers pre-configured tools, libraries, and frameworks that simplify the development and deployment process, helping developers focus on coding and application logic rather than server management and infrastructure maintenance. Examples of PaaS include AWS Elastic Beanstalk, Google App Engine, and Microsoft Azure App Service.

2)AWS Elastic Beanstalk

AWS Elastic Beanstalk is an easy-to-use PaaS solution that automates the deployment, scaling, and management of applications in the AWS Cloud. With Elastic Beanstalk, developers can deploy applications written in popular programming languages (like Python, Java, Node.js, PHP, Ruby, and more) by simply uploading code. Beanstalk handles tasks such as provisioning the necessary infrastructure, managing load balancing, auto-scaling, monitoring, and more. It's particularly useful for fast deployments and quick iterations while maintaining flexibility over the underlying AWS services.

3)Components of Elastic Beanstalk

Elastic Beanstalk comprises several key components that work together to create and manage the environment in which applications run. These include:

1. Application: A container that holds different environments. Each environment runs a version of the application, allowing for testing and production environments within the same application container.
2. Environment: The specific infrastructure resources and settings used to run an application version. Each environment is associated with a URL, load balancer, and other resources that Elastic Beanstalk configures based on the application's needs.

3. Environment Configuration: Settings applied to the environment, such as instance type, security groups, scaling policies, and software versions. These configurations can be updated as the application evolves.

4. Application Version: A deployable version of the code. Multiple versions can be stored and rolled back if necessary.

5. Environment Tier: The type of environment, either Web Server (used for web applications) or Worker (for asynchronous, background tasks).

6. Elastic Beanstalk CLI (EB CLI): A command-line tool that facilitates the management of applications and environments directly from the command line.

4) IAM (Identity and Access Management)

AWS Identity and Access Management (IAM) is a security service that enables users to securely control access to AWS services and resources. With IAM, users can manage permissions and access controls for various AWS resources, ensuring that only authorized individuals or applications can perform specific actions. Key IAM components include:

1. Users: Individual user accounts representing a person or service.

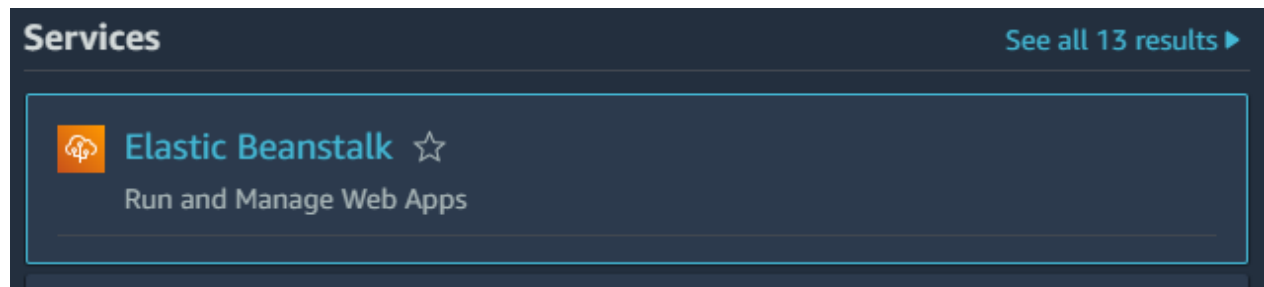
2. Groups: Collections of users to simplify permission management. Permissions applied to a group apply to all members.

3. Roles: Roles allow users or services to assume permissions temporarily, ideal for cross-account or temporary access.

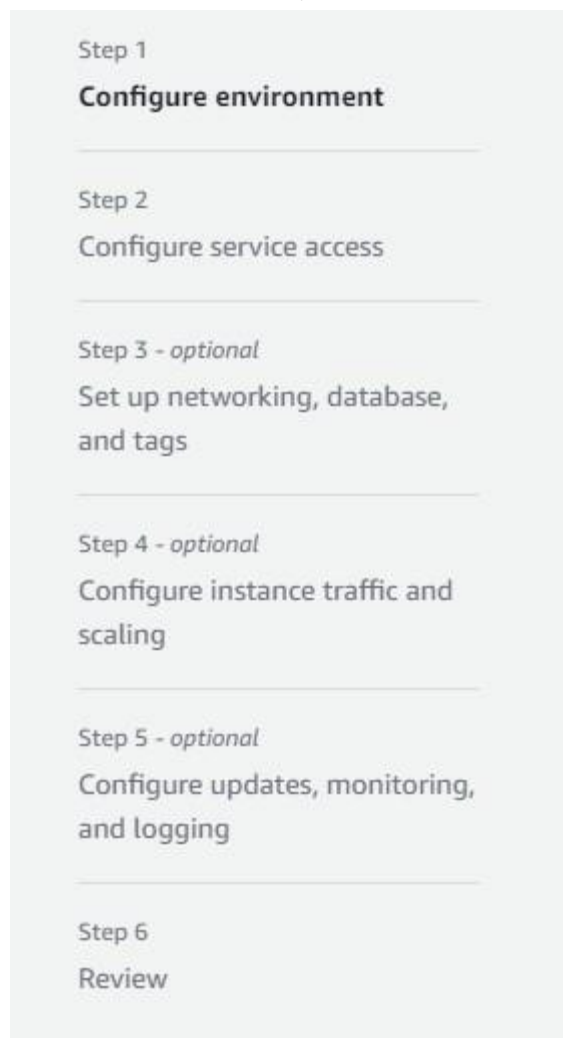
4. Policies: JSON-based documents that define permissions and can be attached to users, groups, or roles to control access.

IAM plays a crucial role in Elastic Beanstalk by securing access to application environments and associated resources, allowing developers to manage permissions granularly.

1. We'll be using Elastic Beanstalk - Python



2. There will be the steps



3. Fill in the details

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.

Domain

.eu-north-1.elasticbeanstalk.com

[Check availability](#)

Environment description

4. Select the 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
4.0.7 (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

5. Go to the IAM services -> Create role

Identity and Access Management (IAM)

Q Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

▼ Access reports

Access Analyzer

External access

Unused access

Analyzer settings

Credential report

Organization activity

Service control policies (SCPs)

Related consoles

[IAM Identity Center](#)

[AWS Organizations](#)

IAM > Roles

Roles 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.

Q Search

< 1 > ⚙

Role name

Trusted entities

Last activity

Roles Anywhere info

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

Manage

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

X.509 Standard
Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.

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

Trusted entity type

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

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

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

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

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

Use case

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

Service or use case

EC2 ▼

Choose a use case for the specified service.

Use case

- ☒ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☐ **EC2 Role for AWS Systems Manager**
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- ☐ **EC2 Spot Fleet Role**
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- ☐ **EC2 - Spot Fleet Auto Scaling**
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

Step 2
Add permissions

Step 3
Name, review, and create

Permissions policies (3/908) [Info](#)

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

Filter by Type

Q beans X All types 14 matches

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

Role FirstTime_Role created.

[IAM](#) > Roles

Roles (3) [Info](#)

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


Q Search

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linker)	-
<input type="checkbox"/>	FirstTime_Role	AWS Service: ec2	-

6. Create new service role, and select the EC instance profile

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](#) 



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](#)

7. Set up network and database

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-0a88c42ba8195a843 | (172.31.0.0/16) ▼

[Create custom VPC](#)

Instance settings

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

Public IP address

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

☐ Activated

Instance subnets

Filter instance subnets

<input type="checkbox"/>	Availability Zone	Subnet ▲	CIDR	Name
<input type="checkbox"/>	ap-south-1c	subnet-04ee19ce2...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-053798557...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-059670b7...	172.31.0.0/20	

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](#)

8. Keep the settings default

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

5 minute ▼

Instance metadata service (IMDS)

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

9. Keep the settings default

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](#) [\[E\]](#)

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

Lifecycle

Keep logs after terminating environment

10. Review and submit!

11. The environment will start launching

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

Elastic Beanstalk > Environments > Trap-env

Trap-env

Environment overview

Health

Unknown

Domain

-

Environment ID

e-d8xdksuqke

Application name

Trap

Platform

Python 3.11 running on 64bit Amazon Linux 2023/4.0.7

Running version

-

Platform state

Supported

Events (2)

Time

January 23, 2024 08:36:39 (UTC+5:30)

INFO

Using elasticbeanstalk-ap-south-1-730335184583 as Amazon S3 storage bucket for environment data.

January 23, 2024 08:36:38 (UTC+5:30)

INFO

createEnvironment is starting.

Environment overview

Health

Pending

Domain

Trap-env.eba-pxx2hqv2.ap-south-1.elasticbeanstalk.com

Environment ID

e-d8xdksuqke

Application name

Trap

The webapp is launched!

← → Not secure trap-env.eba-pxx2hqv2.ap-south-1.elasticbeanstalk.com

Congratulations

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

This environment is launched with Elastic Beanstalk Python Platform


What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy a Django Application to AWS Elastic Beanstalk
- Deploy a Flask Application to AWS Elastic Beanstalk
- Customizing and Configuring a Python Container
- Working with Logs

1. We'll be using Elastic Beanstalk - Java

Services

[See all 13 results ►](#)

 **Elastic Beanstalk** ☆
Run and Manage Web Apps

2. There will be the steps

Step 1

Configure environment

Step 2

Configure service access

Step 3 - *optional*

Set up networking, database,
and tags

Step 4 - *optional*

Configure instance traffic and
scaling

Step 5 - *optional*

Configure updates, monitoring,
and logging

Step 6

Review

Configure environment [Info](#)

Environment tier [Info](#)

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

☒ Web server environment

Run a website, web application, or web API that serves HTTP requests. [Learn more](#) 

☐ Worker environment

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#) 

Application information [Info](#)

Application name

AWS-Java

Maximum length of 100 characters.

► Application tags (optional)

Activate Wi

12. Select the 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

4.0.7 (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

13. Go to the IAM services -> Create role

Identity and Access Management (IAM)

Q Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

▼ Access reports

Access Analyzer

External access

Unused access

Analyzer settings

Credential report

Organization activity

Service control policies (SCPs)

Related consoles

[IAM Identity Center](#)

[AWS Organizations](#)

IAM > Roles

Roles 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.

Q Search

< 1 > ⚙

Role name

Trusted entities

Last activity

Roles Anywhere info

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

Manage

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

X.509 Standard
Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.

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

Trusted entity type

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

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

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

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

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

Use case

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

Service or use case

EC2 ▼

Choose a use case for the specified service.

Use case

- ☒ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☐ **EC2 Role for AWS Systems Manager**
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- ☐ **EC2 Spot Fleet Role**
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- ☐ **EC2 - Spot Fleet Auto Scaling**
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

Step 2

Add permissions

Step 3

Name, review, and create

Permissions policies (3/908) [Info](#)

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

Q beans

X

Filter by Type

All types

14 matches

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

Role FirstTime_Role created.

IAM > Roles

Roles (3) [Info](#)

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

Q Search

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linker...	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service...	-
<input type="checkbox"/>	FirstTime_Role	AWS Service: ec2	-

14. Create new service role, and select the EC instance profile

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](#)

Choose a key pair



EC2 instance profile

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

Java-Server



15. Set up network and database

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-0a88c42ba8195a843 | (172.31.0.0/16) ▼

[Create custom VPC](#)

Instance settings

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

Public IP address

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

☐ Activated

Instance subnets

Filter instance subnets

<input type="checkbox"/>	Availability Zone	Subnet ▲	CIDR	Name
<input type="checkbox"/>	ap-south-1c	subnet-04ee19ce2...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-053798557...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-059670b7...	172.31.0.0/20	

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](#)

16. Keep the settings default

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

5 minute ▼

Instance metadata service (IMDS)

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

17. Keep the settings default

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

Lifecycle

Keep logs after terminating environment

18. Review and submit!

19. The environment will start launching

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

Elastic Beanstalk > Environments > AWS-Java-env

AWS-Java-env

Info

⌂

Actions

Upload and deploy

Environment overview

Health

Unknown

Domain

-

Environment ID

e-dmjt62hpnb

Application name

AWS-Java

Platform

Change version

Platform

Corretto 21 running on 64bit Amazon Linux 2023/4.2.0

Running version

-

Platform state

Supported

Environment successfully launched.

Elastic Beanstalk > Environments > AWS-Java-env

AWS-Java-env

Info

⌂

Actions

Upload and deploy

Environment overview

Health

Ok

Domain

AWS-Java-env.eba-3fup2hrb.ap-south-1.elasticbeanstalk.com

Environment ID

e-dmjt62hpnb

Application name

AWS-Java

Platform

Change version

Platform

Corretto 21 running on 64bit Amazon Linux 2023/4.2.0

Running version

-

Platform state

Supported

The webapp is launched!

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](#)

