

## Practical-2 Platform as a service using AWS.

SAP ID: 86092300024

### **Platform as a Service:**

Platform as a Service (PaaS) is a cloud computing service model that provides a ready-made platform allowing customers to develop, deploy, and manage applications without dealing with the complexities of underlying infrastructure. AWS PaaS solutions, such as Elastic Beanstalk and App Runner, automate infrastructure provisioning, scaling, and maintenance, allowing developers to focus on writing code without dealing with the complexities of managing servers. These services provide integrated development tools, support for various programming languages, scalability, and managed services like databases, all while following a pay-as-you-go pricing model.

### **Here are some key characteristics and components of PaaS in AWS:**

1. **Application Hosting Environment:**
  - a. PaaS provides a ready-made environment for hosting applications. AWS offers services like AWS Elastic Beanstalk, which allows developers to upload their application code, and the platform takes care of provisioning and managing the underlying infrastructure.
2. **Automated Deployment:**
  - a. PaaS offerings typically include automated deployment and scaling capabilities. AWS services like AWS Elastic Beanstalk and AWS App Runner automate the deployment process, making it easier to handle updates and changes to applications.
3. **Managed Services:**
  - a. AWS PaaS offerings often include managed services for databases, messaging, caching, and other common application components. For example, Amazon RDS (Relational Database Service) is a managed database service that can be leveraged within a PaaS environment.
4. **Development Frameworks:**
  - a. PaaS platforms often support multiple programming languages and frameworks, allowing developers to choose the tools they are most comfortable with. AWS Elastic Beanstalk supports various programming languages, including Java, Python, Node.js, and more.

5. **Scalability:**
  - a. PaaS solutions in AWS are designed to scale easily. They automatically handle the provisioning and scaling of resources based on application demand. This ensures that the application can handle varying levels of traffic without manual intervention.
6. **Integrated Development Tools:**
  - a. PaaS offerings typically come with integrated development tools that facilitate the development and debugging process. AWS provides tools like AWS CodeBuild, AWS CodePipeline, and AWS CodeDeploy for continuous integration and continuous deployment (CI/CD) workflows.
7. **Cost Management:**
  - a. PaaS services often follow a pay-as-you-go pricing model, allowing users to pay only for the resources they consume. This can lead to cost savings as users do not need to invest in and maintain their own infrastructure.
8. **Security and Compliance:**
  - a. AWS PaaS offerings incorporate security features and compliance measures to help protect applications and data. AWS manages the security of the underlying infrastructure, while users are responsible for securing their application code and data.

## Advantages of PAAS:

1. **Simplified Development:** PaaS abstracts the complexities of infrastructure management, allowing developers to focus on coding and application logic rather than worrying about hardware, networking, and other underlying components. This simplification accelerates the development process.
2. **Faster Time-to-Market:** PaaS provides pre-configured development frameworks, tools, and services, reducing the time needed to set up and manage infrastructure. This results in quicker deployment of applications, enabling faster time-to-market for new features or products.
3. **Scalability:** PaaS platforms often include automated scaling capabilities. This means applications can easily handle varying levels of traffic, and resources can be automatically provisioned or de-provisioned based on demand. It ensures optimal performance without manual intervention.
4. **Cost-Efficiency:** PaaS typically follows a pay-as-you-go pricing model, allowing users to pay only for the resources they consume. This can result in cost savings, as organizations do not need to invest heavily in infrastructure or worry about overprovisioning.
5. **Flexibility and Portability:** PaaS supports various programming languages and frameworks, offering flexibility to developers. It also provides a level of portability,

allowing applications to be easily moved between different cloud providers or environments.

## 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.

## Components of Amazon ElasticBeanStalk:

1. **Application:** Elastic Beanstalk directly takes in our project code. So the Elastic Beanstalk application is named the same as your project home directory.
2. **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.
3. **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.

Date:-23/01/2024

4. **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.
5. **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.
6. **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.
7. **Automatic Provisioning:** Elastic Beanstalk takes away the burden of choosing the right services and configuring their security groups to work together.
8. **Impossible to Outgrow:** AWS claims that since Elastic Beanstalk uses the Auto Scaling feature it can, in theory, handle any amount of internet traffic.

## **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.

**Date:-23/01/2024**

**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.

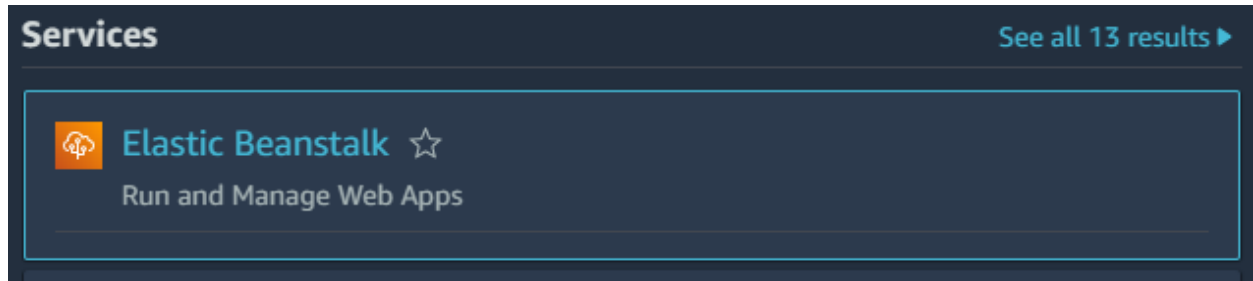
**Resource-Level Permissions:** IAM policies can define permissions not only at the service level but also at the resource level. This allows fine-grained control over access to specific AWS resources.

**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.

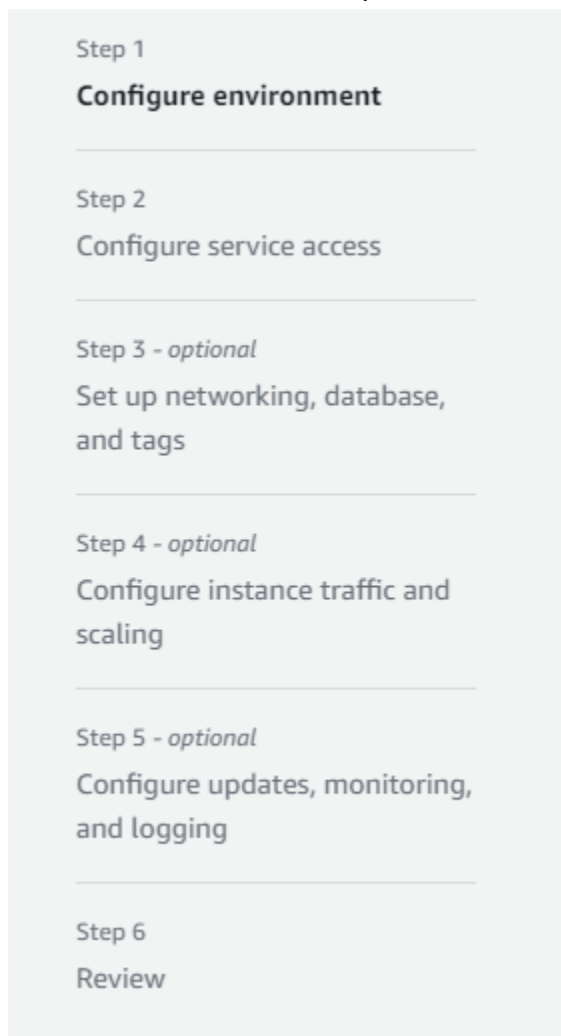
**Policy Conditions:** IAM policies can include conditions that must be met for the policy to be in effect. Conditions can be based on factors such as the time of day, the source IP address, or the use of MFA.

Date:-23/01/2024

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

Date:-23/01/2024

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)

Search IAM

Dashboard

Access management

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

Refresh

Delete

Create role

Search

Role name

Trusted entities

Last activity

Roles Anywhere

Info

Manage

Access AWS from your non AWS workloads

X.509 Standard

Temporary credentials

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

Filter by Type

All types

14 matches

	Policy name	Type	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSElasticBeanstalk</a>	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkCustomPlatformforEC2Role</a>	AWS managed	Provide the instance in your custom pl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkEnhancedHealth</a>	AWS managed	AWS Elastic Beanstalk Service policy f...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</a>	AWS managed	This policy is for the AWS Elastic Bean...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkMulticontainerDocker</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="#">AWSElasticBeanstalkRoleWorkerTier</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="#">AWSElasticBeanstalkWorkerTier</a>	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"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service	-
<input type="checkbox"/>	<a href="#">FirstTime_Role</a>	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

Date:-23/01/2024

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

<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](#) [↗](#)

##### 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.

Time

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!

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

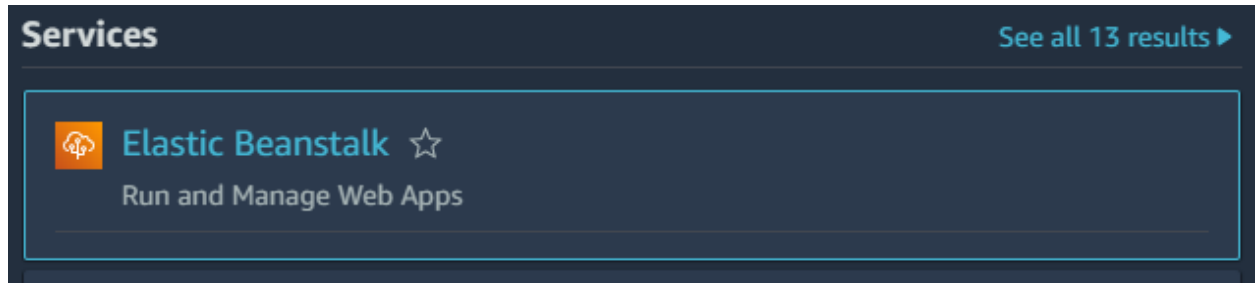
Deploy a Flask Application to AWS Elastic Beanstalk

Customizing and Configuring a Python Container

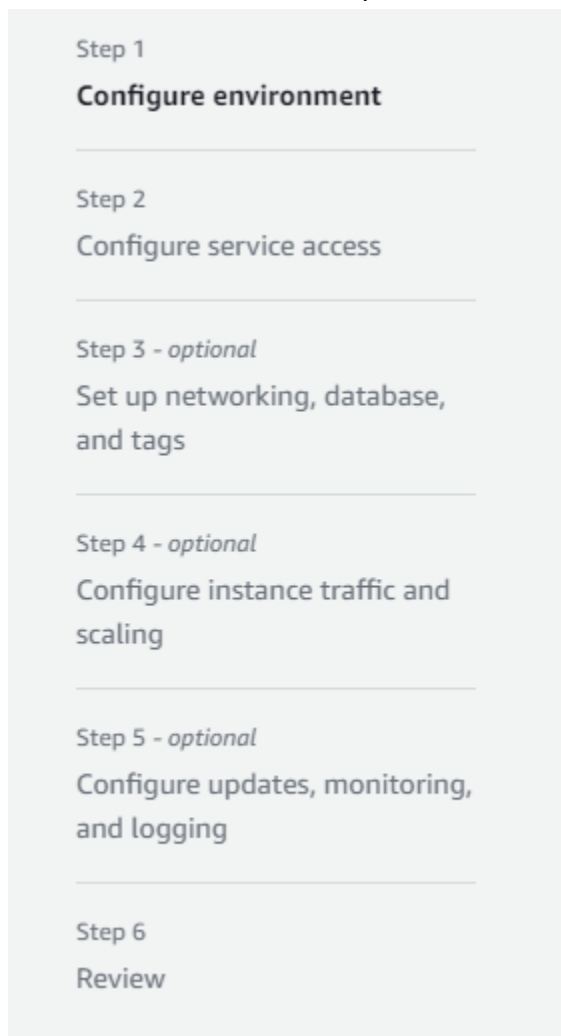
Working with Logs

Date:-23/01/2024

1. We'll be using Elastic Beanstalk - Java



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

AWS-Java

Maximum length of 100 characters.

► Application tags (optional)

Activate Wi

## 12. Select the platform

Date:-23/01/2024

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)

Search IAM

Dashboard

Access management

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

Refresh

Delete

Create role

Search

Role name

Trusted entities

Last activity

Roles Anywhere

Info

Manage

Access AWS from your non AWS workloads

X.509 Standard

Temporary credentials

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

Filter by Type

All types

14 matches

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSElasticBeanstalk</a>	AWS managed	Grants account administrative permissions...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkCustomPlatformforEC2Role</a>	AWS managed	Provide the instance in your custom platform...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkEnhancedHealth</a>	AWS managed	AWS Elastic Beanstalk Service policy for enhanced health...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</a>	AWS managed	This policy is for the AWS Elastic Beanstalk...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkMulticontainerDocker</a>	AWS managed	Provide the instances in your multicontainer Docker...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkReadOnly</a>	AWS managed	Grants read-only permissions. Explicitly denies...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCore</a>	AWS managed	AWSElasticBeanstalkRoleCore (Elastic Beanstalk...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCWL</a>	AWS managed	(Elastic Beanstalk operations role) Allow...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleECS</a>	AWS managed	(Elastic Beanstalk operations role) Allow...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleRDS</a>	AWS managed	(Elastic Beanstalk operations role) Allow...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleSNS</a>	AWS managed	(Elastic Beanstalk operations role) Allow...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleWorkerTier</a>	AWS managed	(Elastic Beanstalk operations role) Allow...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Provide the instances in your web tier...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWorkerTier</a>	AWS managed	Provide the instances in your worker tier...

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"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-
<input type="checkbox"/>	<a href="#">FirstTime_Role</a>	AWS Service: ec2	-

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

Date:-23/01/2024

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



Date:-23/01/2024

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

Environment ID

e-dmjt62hpnb

Domain

-

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

Environment ID

e-dmjt62hpnb

Domain

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

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!

Date:-23/01/2024

← → ↻ ⚠ Not secure aws-java-env.eba-3fup2hrb.ap-south-1.elasticbeanstalk.com ☆ 📱 W ⋮

# 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](#)

## Elastic Bean - Tomcat

### 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


.ap-south-1.elasticbeanstalk.com

[Check availability](#)

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

Tomcat ▼

### Platform branch

Tomcat 10 with Corretto 17 running on 64bit Amazon Linux 2023 ▼

### Platform version

5.1.3 (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.

### Version label


Unique name for this version of your application code.

*Version label*

Source code origin. Maximum size 500 MB

- ☒ **Local file**

### Upload application

 **Choose file**

✔ File name: **Calendar.war**

File must be less than 500MB max file size

- ☐ **Public S3 URL**

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

Elastic Beanstalk > Environments > Tomcatapp-env

Tomcatapp-env

Environment overview

Health

Unknown

Domain

-

Environment ID

e-kw65ahctwx

Application name

Tomcatapp

Platform

Tomcat 10 with Corretto 17 running on 64bit Amazon Linux 2023/5.1.3

Running version

-

Platform state

Supported

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Events (1)

Filter events by text, property or value

Time

January 30, 2024 07:39:02 (UTC+5:30)

Type

INFO

Details

createEnvironment is starting.

← → ↺ ⚠ Not secure tomcatapp-env.eba-sff59uwz.ap-south-1.elasticbeanstalk.com/Calendar.html

# GWT Calendar

Click on day to get date popup. Example Datepicker. Built with the tomcat war builder.  
<http://code.google.com/p/gwt-examples/>

< January >		< 2024 >				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			