

# Platform as a Service

Writeup:-

- Platform as a service

AWS Elastic Beanstalk is the PaaS offering from AWS that makes it easy to deploy and manage applications in the cloud without worrying about the infrastructure. Developers can simply upload their code and Elastic Beanstalk automatically handles provisioning, load balancing, auto-scaling and monitoring.

Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

- Elastic Beanstalk

Elastic Beanstalk is a PaaS service offered by AWS to deploy and scale web applications quickly without worrying about the infrastructure. It automatically handles capacity provisioning, load balancing, scaling and application health monitoring. Developers just have to upload their code and Elastic Beanstalk will deploy it on AWS infrastructure like EC2, auto scale it and monitor it.

Why Elastic BeanStalk

- i. It supports multiple languages like Java, Python, Go etc. and platforms like Docker.
- ii. Beanstalk integrates well with other AWS services like EC2, S3, RDS etc.
- iii. The main benefits are fast and automated application deployment and management, multiple environments, auto scaling, and cost efficiency.

- Components of beanstalk

- i. Application: This is the actual web application code packaged into a zip and uploaded to Elastic Beanstalk.
- ii. Application Version: Each deployment of code is an application version. Rollbacks can be done to previous versions.
- iii. Environment: This is a version of the application running on AWS resources. We can create multiple environments like dev, test, prod etc from the same application.
- iv. Configuration Templates: These allow customization of the AWS resources powering an environment like EC2 instance type, autoscaling settings etc.
- v. Events: Important lifecycle events like deployments, scaling etc are logged for debugging.

- IAM

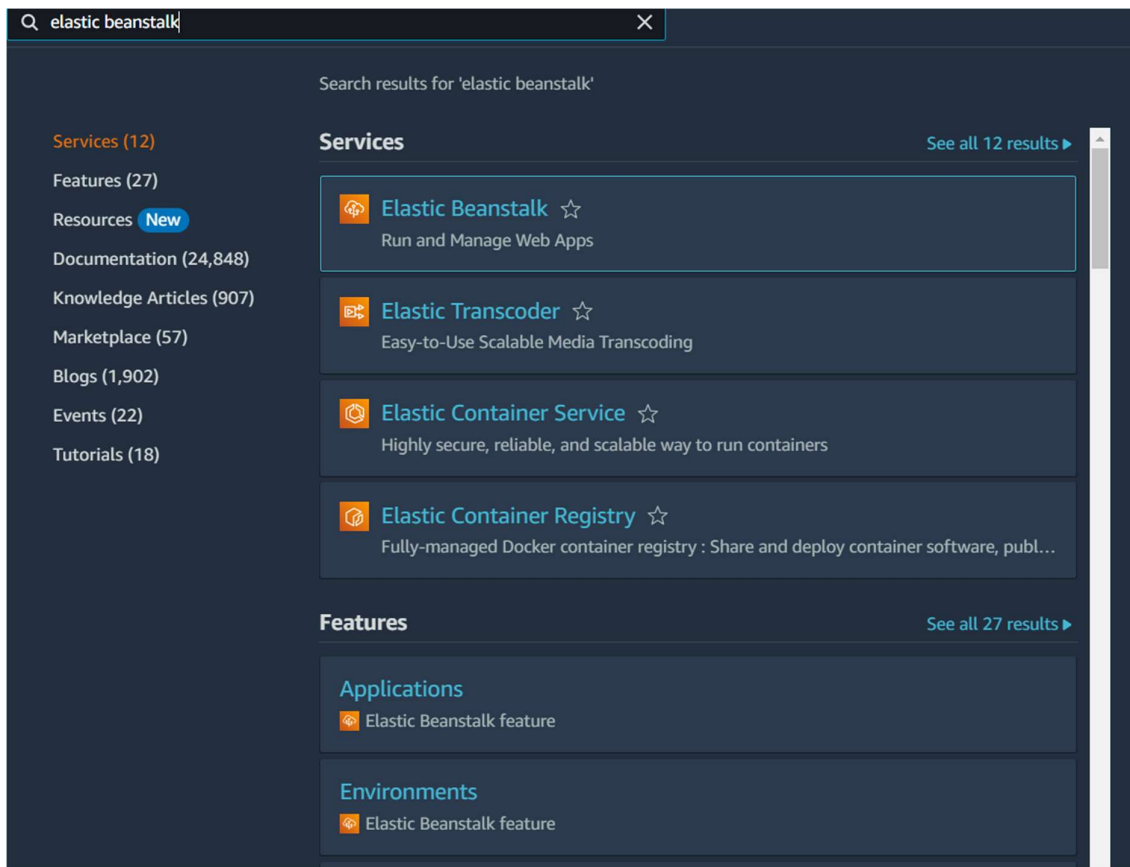
- i. IAM allows managing users, roles and permissions to access AWS services and resources.
- ii. Users can be created and assigned granular permissions policies.
- iii. Roles can be created with permissions and then assigned to AWS resources like EC2 instances.
- iv. Policies define the permissions like which AWS actions can be performed on which resources.
- v. IAM is important for security, access control and compliance in AWS.

v. • Implement PAAS using elastic beanstalk for the following.

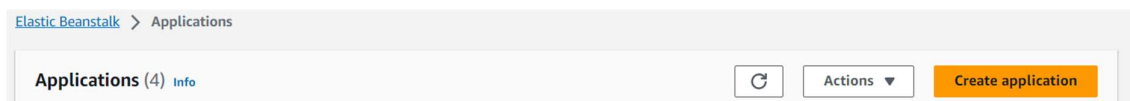
1. Server
2. Java
3. Python
4. Node.js

## For Python

Step 1: Search for elastic beanstalk



Step 2: Create a new application



### Step 3: Create a new environment

## 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

py1

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

Pv1-env

### Step 4: Choose python as 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

Python

Platform branch

Python 3.11 running on 64bit Amazon Linux 2023

Platform version

4.0.8 (Recommended)

## Step 5: Search IAM and create a new role

[IAM](#) > Roles

**Roles (3)** [Info](#) Refresh Delete Create role

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

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">AWSServiceRoleForAutoScaling</a>	AWS Service: autoscaling (Service-Linker)	34 minutes ago
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-

## Step 6: Select EC2 as use case

### 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**

## Step 7: Select the following permissions and then create the role

<input checked="" type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Multi-Container Docker Engine</a>	AWS managed	Provide the instances in your multicon...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Read-Only</a>	AWS managed	Grants read-only permissions. Explicit...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Role Core</a>	AWS managed	AWS Elastic Beanstalk Role Core (Elastic ...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Role CWL</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Role ECS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Role RDS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Role SNS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Role Worker Tier</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Web Tier</a>	AWS managed	Provide the instances in your web serv...
<input checked="" type="checkbox"/>		<a href="#">AWS Elastic Beanstalk Worker Tier</a>	AWS managed	Provide the instances in your worker e...

## Step 8: Configure service access

### 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](#)

Choose a key pair

EC2 instance profile

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

py\_role

View permission details

Cancel

Skip to review

Previous

Next

Step 9: Configure VPC

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-046ef1fe00d0da77e | (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				
<div>Filter instance subnets</div>				
	Availability Zone	Subnet ▲	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-00ea1c17b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-083e52afb...	172.31.0.0/20	

## Step 10: Skip the next step without making changes

☒ **x86\_64**  
This architecture uses x86 processors and is compatible with most third-party tools and libraries.

☐ **arm64 - new**  
This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

**Instance types**  
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose x86 instance types

t3.micro × t3.small ×

**AMI ID**  
Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

ami-032f92d0dc3947eae

**Availability Zones**  
Number of Availability Zones (AZs) to use.

Any

**Placement**  
Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs)

**Scaling cooldown**

360

seconds

Cancel

Skip to review

Previous

Next

## Step 11: Skip the next step too

**Environment properties**  
The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value	
<div>PYTHONPATH</div>	<div>/var/app/venv/staging-LQM1lest/bin</div>	<div>Remove</div>

Add environment property

Cancel

Previous

Next

## Step 12: Proceed and submit

**Platform software**

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		

**Environment properties**

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

Cancel Previous Submit

## Step 13: Wait and let the environment launch

Environment successfully launched.

[Elastic Beanstalk](#) > [Environments](#) > Py1-env

### Py1-env

Info

Refresh Actions Upload and deploy

**Environment overview**

**Health**  
Ok

**Domain**  
[Py1-env.eba-anmfgz33.ap-south-1.elasticbeanstalk.com](#)

**Environment ID**  
e-kwm2wttrdg

**Application name**  
py1

**Platform** Change version

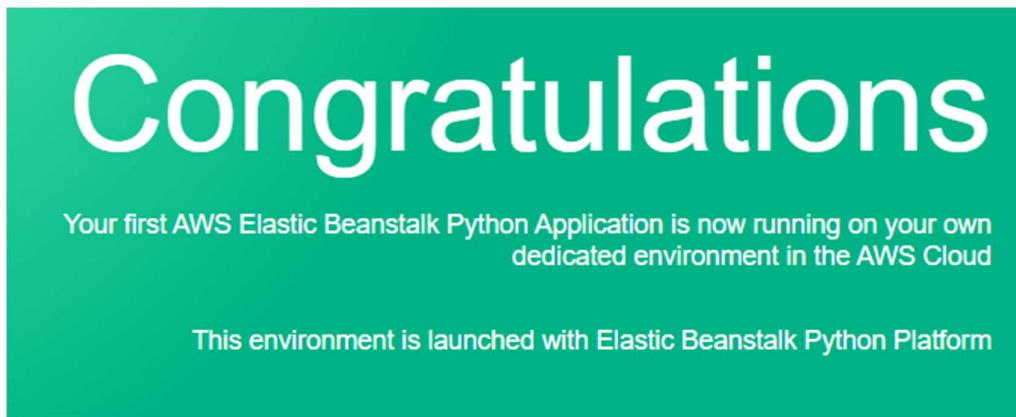
**Platform**  
Python 3.11 running on 64bit Amazon Linux 2023/4.0.8

**Running version**  
-

**Platform state**  
Supported



Step 14: Click on the domain link to check whether the environment is launched properly



## For Java

From Step 1 to Step 8 we repeat the process and from Step 4 we select Java and continue the steps

Step 9: Select database subnet

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

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

**Choose database subnets (3)**  

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	ap-south-1a	subnet-00ea1c17b...	172.31.32.0/20	
<input checked="" type="checkbox"/>	ap-south-1b	subnet-083e52afb...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-0ea51be5d...	172.31.16.0/20	

Step 10: Skip the steps as done previously and submit

### Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value	
<input type="text" value="GRADLE_HOME"/>	<input type="text" value="/usr/local/gradle"/>	<input type="button" value="Remove"/>
<input type="text" value="M2"/>	<input type="text" value="/usr/local/apache-maven/bin"/>	<input type="button" value="Remove"/>
<input type="text" value="M2_HOME"/>	<input type="text" value="/usr/local/apache-maven"/>	<input type="button" value="Remove"/>
<input type="button" value="Add environment property"/>		



Step 11: Wait and let the environment launch

✔ Environment successfully launched.

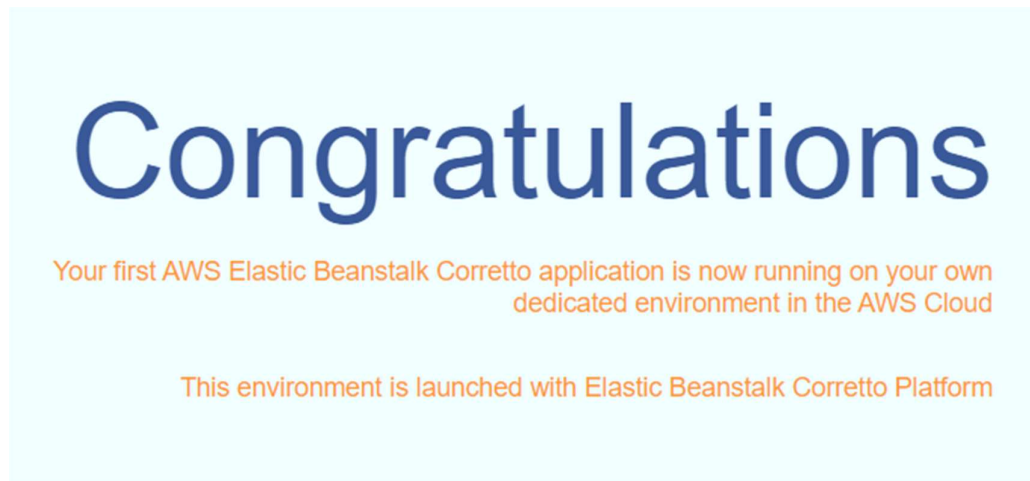
[Elastic Beanstalk](#) > [Environments](#) > Java1-env

## Java1-env [Info](#)

### Environment overview

Health  <b>Warning</b>	Environment ID  e-pqsypt2ph3
Domain <a href="#">Java1-env.eba-7jwyijzv.ap-south-1.elasticbeanstalk.com</a>	Application name <a href="#">java1</a>

Step 12: Click on domain link to check whether it is getting launched properly



## For Tomcat

Repeat the steps by selecting tomcat as platform

Step1: Upload calendar file in application code

**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.

Source code origin. Maximum size 500 MB

☒ Local file

Upload application

☒ File name: **Calendar.war**  
File must be less than 500MB max file size

☐ Public S3 URL

## Step 2: Skip the next steps again and submit

**Platform software**

Lifecycle	Log streaming	Initial JVM heap size (Xms)
false	Deactivated	256m
Max JVM heap size (Xmx)	JVM options	Proxy server
256m	–	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		

**Environment properties**

Key	Value
JDBC_CONNECTION_STRING	

Cancel Previous Submit

## Step 3: Wait for the environment to launch




✔ Environment successfully launched.

[Elastic Beanstalk](#) > [Environments](#) > Tom1-env

# Tom1-env

[Info](#)

**Environment overview**

Health	Environment ID
 Warning	 e-x6kvazkmk
Domain	Application name
<a href="https://Tom1-env.eba-pfv2p3ru.ap-south-1.elasticbeanstalk.com">Tom1-env.eba-pfv2p3ru.ap-south-1.elasticbeanstalk.com</a> 	<a href="#">tom1</a>

#### Step 4: Open the domain link to check the application

## GWT Calendar

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

< February >				< 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		