



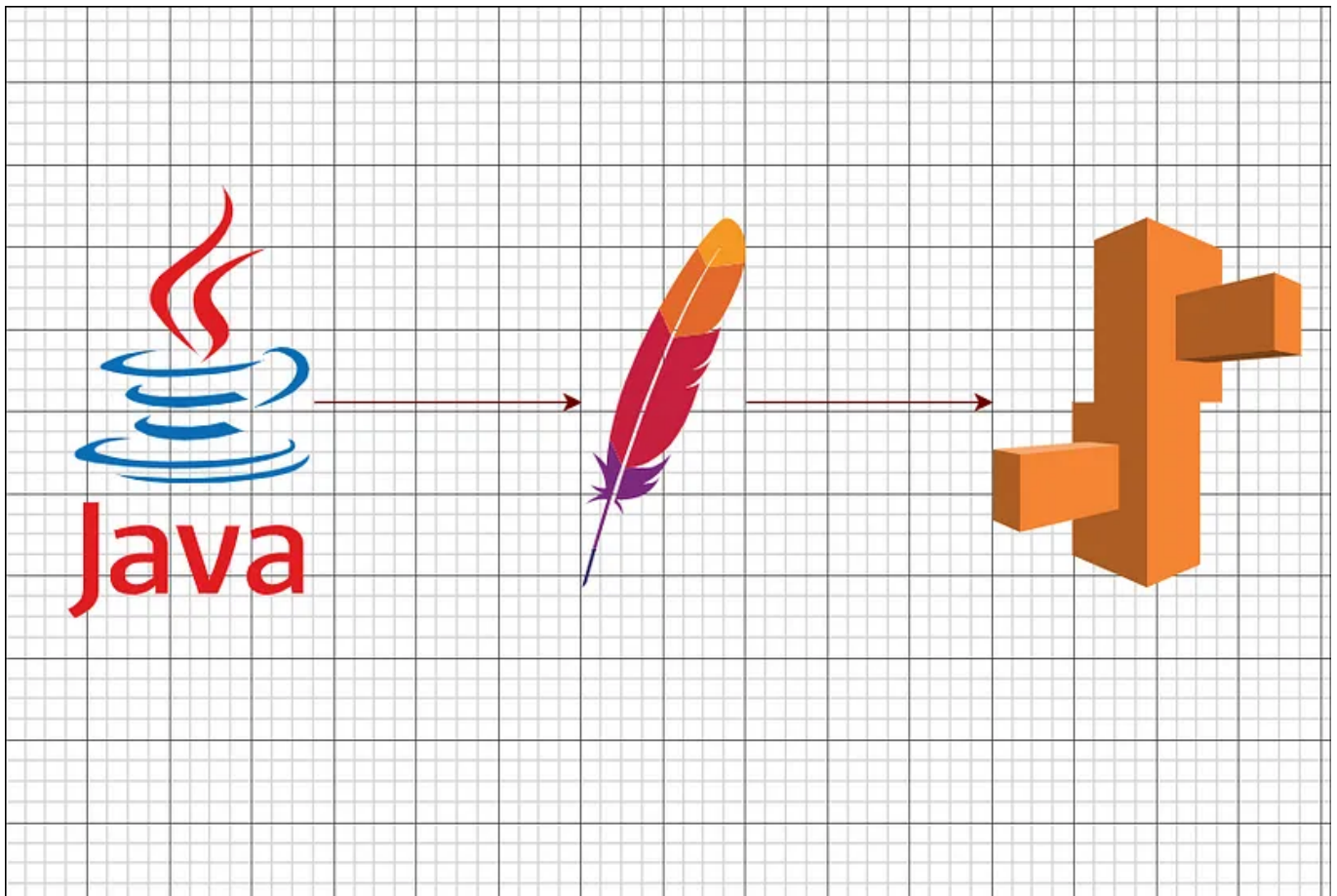
How to Set up a Java application built with Maven on AWS Elastic Beanstalk.

Java, Maven, Elastic Beanstalk, IAM, S3, EC2



Ujwal Pachghare ☀️ · Follow

5 min read · 21 hours ago



Hello friends, In this tutorial we will deploy Java Application which we build with Maven build tool and then deploy on AWS Elastic Beanstalk.

Step-1: Delete Unsupported IAM Roles

Step-2: Create new aws-elasticbeanstalk-ec2-role

Step-3: Create S3 Bucket For Storing Artifact

Step-4: Create EC2 Instance for Building Artifact

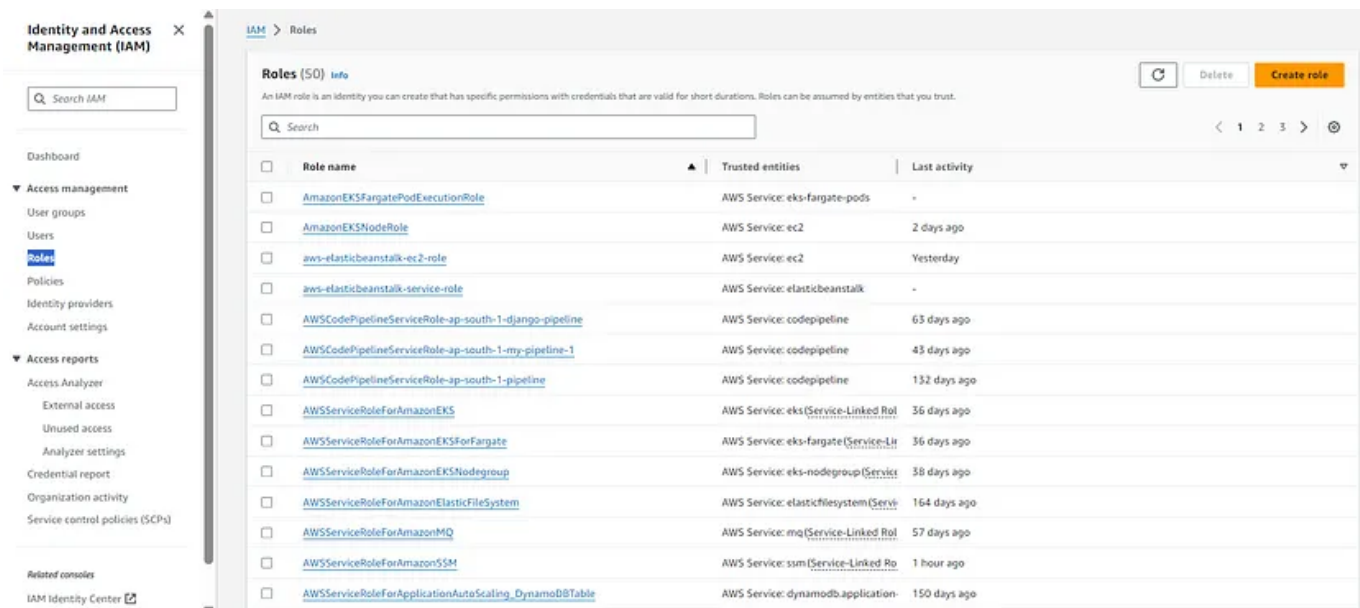
Step-5: Create Application in Elastic Beanstalk

Step 6: clean up

Note:- AWS Elastic Beanstalk has been facing issue related to IAM EC2 Role and to resolve that issue we have to follow respective instruction that are provided by AWS experts.

Step-1: Delete Unsupported IAM Roles

1. Navigate to IAM console → Select Role from left menu



2. search for aws-elasticbeanstalk-service-role in the search bar → Click on Delete button → Delete it

IAM > Roles

Roles (1/50) [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 aws-elasticbea X 2 matches < 1 > ⌂

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	aws-elasticbeanstalk-ec2-role	AWS Service: ec2	Yesterday
<input checked="" type="checkbox"/>	aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-

Roles Anywhere [Info](#) [Manage](#)

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

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.

3. Search for **aws-elasticbeanstalk-ec2-role** → click on **Delete** button → Delete it also

Roles (50) [Info](#)

An IAM role is an identity you can create that has specific permissions with credentials that an

Q beans

<input type="checkbox"/>	Role name
<input type="checkbox"/>	aws-elasticbeanstalk-ec2-role
<input type="checkbox"/>	aws-elasticbeanstalk-service-role
<input type="checkbox"/>	AWSServiceRoleForElasticBeanstalk

Step-2: Create new aws-elasticbeanstalk-ec2-role

1. Click on Create role → Select **AWS Service** → in the use case choose **EC2**

☒ **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

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.

2. Select necessary policies for our IAM Role → Search and check the boxes for policies that are in the image → click Next

Permissions policies (4/943) [Info](#)

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

Filter by Type

All types

14 matches

<input checked="" type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permis...
<input checked="" 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 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"/>	AWSElasticBeanstalkRoleCW	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	AWSElasticBeanstalkRoleEC2	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	AWSElasticBeanstalkRoleRD	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" 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 type="checkbox"/>	AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker e...

3. Give name to our new Role → Click on Create Role (button at the last)

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

aws-elasticbeanstalk-ec2-role

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

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and "+=,._@-_" characters.

Step 1: Select trusted entities

Trust policy

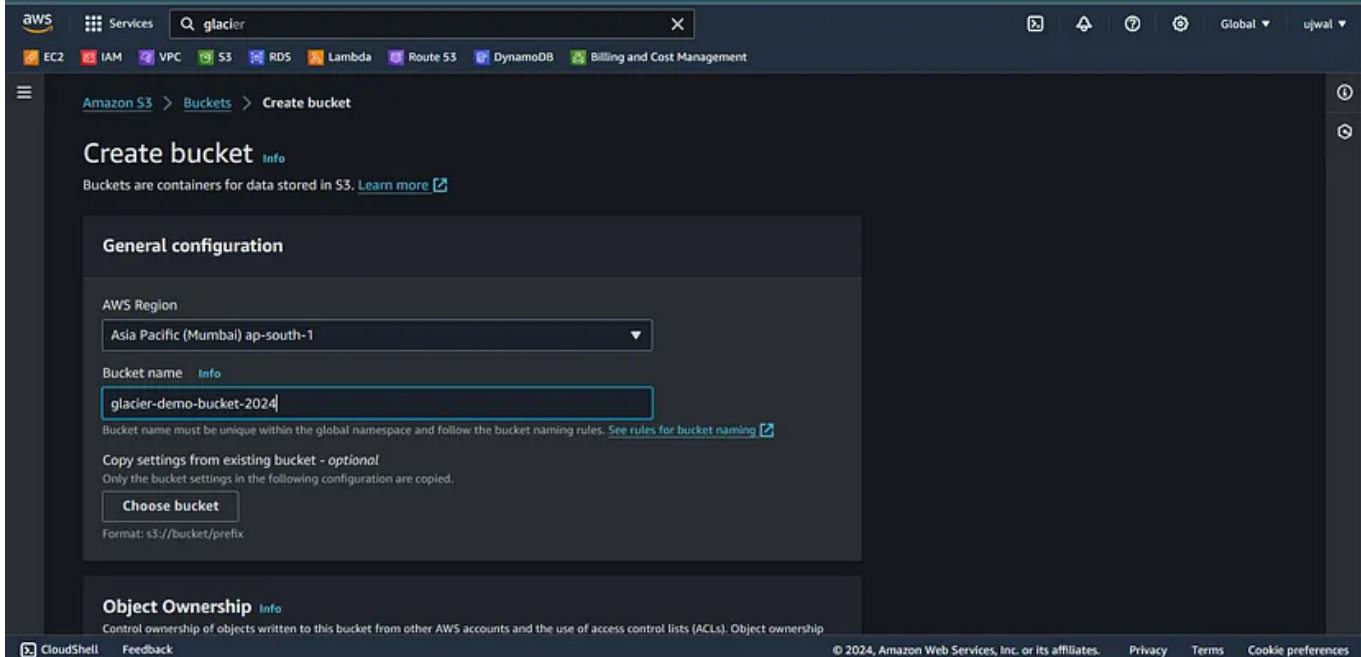
```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "sts:AssumeRole"  
8       ],  
9       "Principal": {  
10        "Service": [  
11          "ec2.amazonaws.com"  
12        ]  
13      }  
14    }  
15  ]  
16 }
```

Step-3: Create S3 Bucket For Storing Artifact

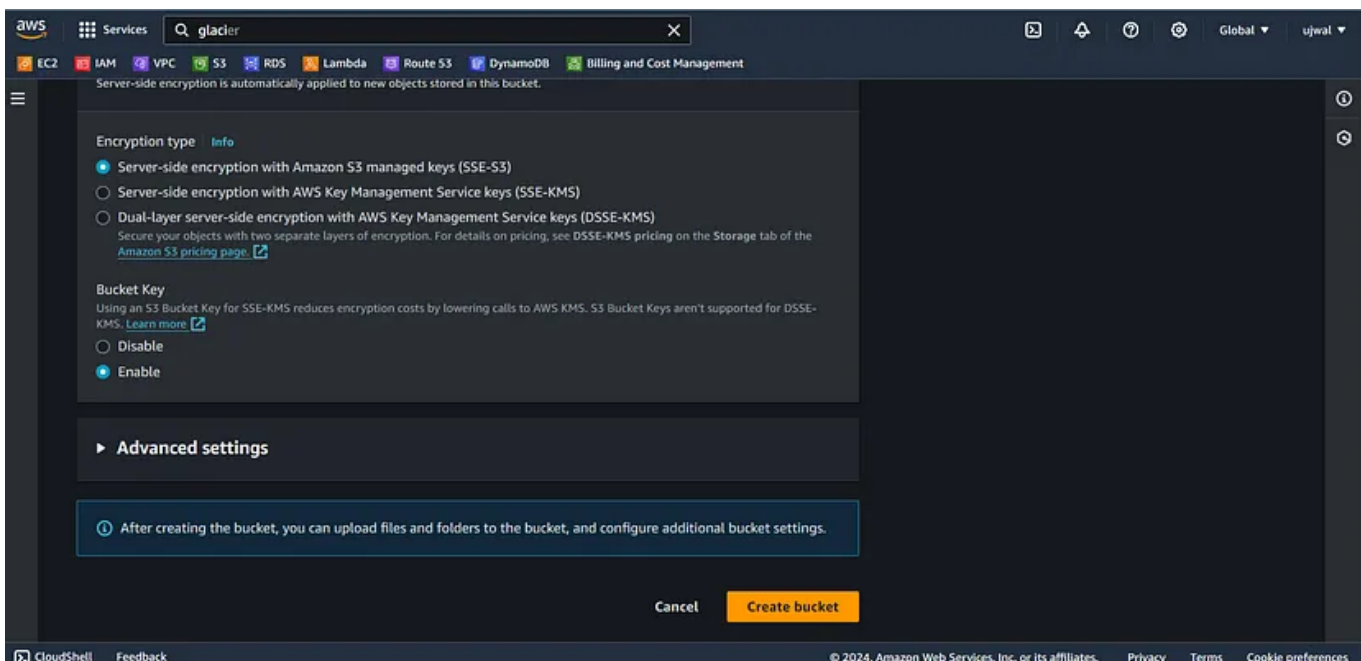
1. Go to S3 Dashboard and Click on Create Bucket

The screenshot shows the Amazon S3 console dashboard. On the left is a navigation sidebar with options like Buckets, Access Grants, and Storage Lens. The main content area has a header 'Storage' and a large title 'Amazon S3' with the tagline 'Store and retrieve any amount of data from anywhere'. Below this is a brief description of the service. To the right, there's a 'Create a bucket' section with a 'Create bucket' button. Further down, there's a 'Pricing' section explaining that there are no minimum fees and a link to the 'Simple Monthly Calculator'. At the bottom, there's a 'How it works' section with a video player titled 'Introduction to Amazon S3'. The footer contains copyright information for 2024, links for Privacy, Terms, and Cookie preferences, and a CloudShell button.

2. Giving unique name to the bucket

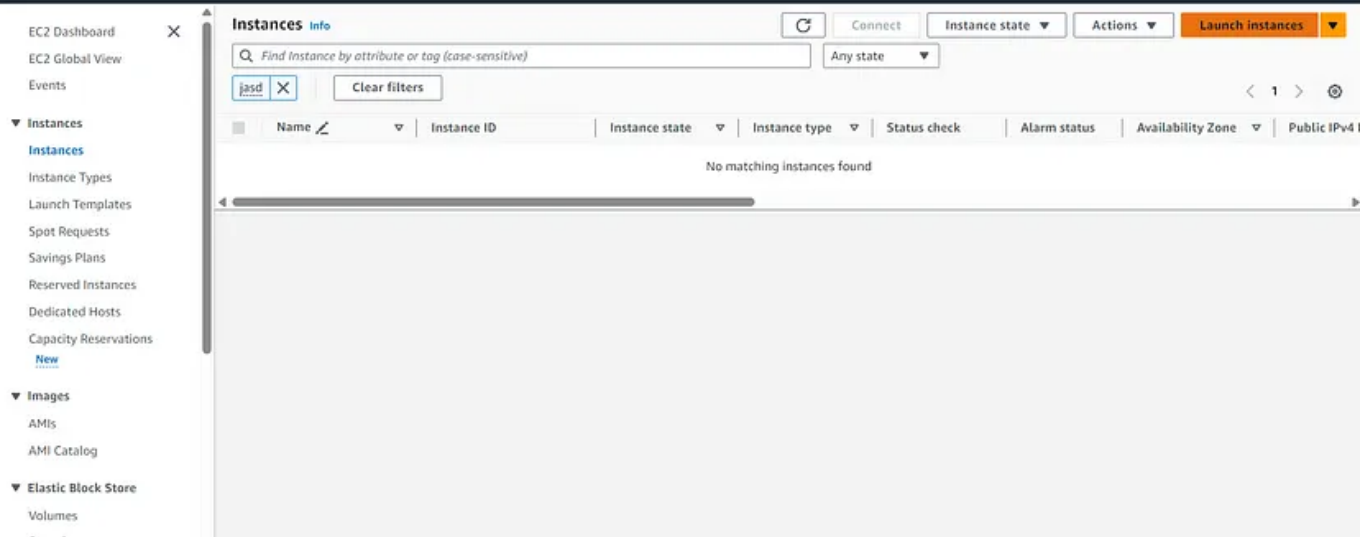


3. let all the options default and click on **Create Bucket** at last.

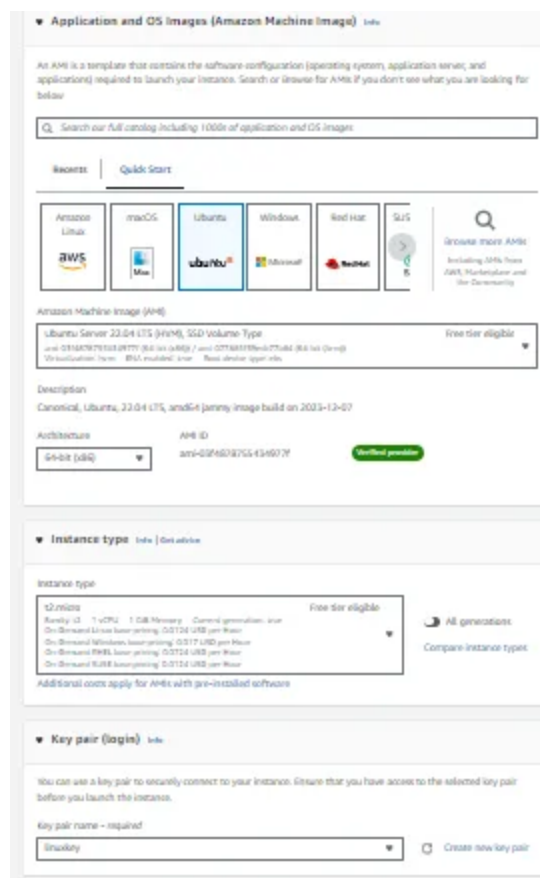


Step-4: Create EC2 Instance For Building Artifact

1. Navigate EC2 Console → Click on **Launch Instance**



2. Give name to EC2 Instance → Select Ubuntu AMI → Select type t2.micro → select existing key-pair → let other things default → Click on launch instance



3. Wait for Status check pass → Select our instance → click on Action button → click on Security → click on modify IAM Role → Select Role that has permissions to put objects in S3 Bucket or create new one with S3FullAccess → at last click on Update IAM Role

Instances (1/4) Info

Find Instance by attribute or tag (case-sensitive) Any state

Instance state: **running** X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/> vprofile-java-app	i-01330f42f9fd67f26	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a
<input type="checkbox"/> S3-deployment-env	i-0170ed466db1c412f	Running	t3.micro	2/2 checks passed	View alarms	
<input type="checkbox"/> Vprofile-app-env	i-09c19ed548ca8aff5	Running	t3.micro	2/2 checks passed	View alarms	
<input type="checkbox"/> S3-demo-with-ec2role-env	i-03738ea180fcdd9c4	Running	t3.micro	2/2 checks passed	View alarms	

Actions: Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates, Monitor and troubleshoot

4. SSH into Instance from browser → Select instance → Click on **Connect**

Instances (1/4) Info

Find Instance by attribute or tag (case-sensitive) Any state

Instance state: **running** X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/> vprofile-java-app	i-01330f42f9fd67f26	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a
<input type="checkbox"/> S3-deployment-env	i-0170ed466db1c412f	Running	t3.micro	2/2 checks passed	View alarms +	ap-south-1a
<input type="checkbox"/> Vprofile-app-env	i-09c19ed548ca8aff5	Running	t3.micro	2/2 checks passed	View alarms +	ap-south-1a
<input type="checkbox"/> S3-demo-with-ec2r...	i-03738ea180fcdd9c4	Running	t3.micro	2/2 checks passed	View alarms +	ap-south-1a

4. Connect to instance by clicking on **Connect**

Connect to instance [Info](#)

Connect to your instance i-01330f42f9fd67f26 (vprofile-java-app) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

 i-01330f42f9fd67f26 (vprofile-java-app)

Connection Type



Connect using EC2 Instance Connect


Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.



Connect using EC2 Instance Connect Endpoint

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

 3.109.184.241

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.



ubuntu



Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect

5. Run following Script in that instance to **build** and **copy** artifact from EC2 to S3 Bucket

```
sudo apt update
sudo apt install maven awscli -y
sudo git clone https://github.com/hkhcoder/vprofile-project.git
cd vprofile-project
mvn install
aws s3 cp target/vprofile-v2.war s3://<your-s3-bucket-name>
```

6. Navigate to S3 Console → go inside our new bucket → click on Copy URL

my-elastic-beackstalk-bucket-2024 [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (1) [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input checked="" type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input checked="" type="checkbox"/>	vprofile-v2.war	war	February 4, 2024, 15:53:21 (UTC+05:30)	52.1 MB	Standard

Step-5: Create Application in Elastic Beanstalk

1. Navigate to AWS Elastic Beanstalk Dashboard → Click on Create Application

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

[Create application](#)

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

Getting started

[Launch a web application](#)

More resources

[Documentation](#)

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. [Learn more](#)

Benefits and features

Easy to get started

Elastic Beanstalk is the simplest way to deploy and run your web application on Amazon Web Services. Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, automatic scaling, and web application health monitoring.

Complete resource control

You have the freedom to select the Amazon Web Services resources, such as Amazon EC2 instance types, that are optimal for your web application. Additionally, Elastic Beanstalk lets you manage and retain full control over the Amazon Web Services resources powering your web application.

2. Select **Web Server Environment** for deploying web application. Give name to our application and it will automatically take the environment name according to our application name.

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

3. Select platform for deploying our app. Our app is written in JAVA, so we will choose

Platform: Tomcat Server

Platform branch: Tomcat 9 with corretto 11 running on 64bit Amazon Linux 2023 (because our app is test and build in Java 11 runtime environment)

Platform version: 5.1.3(Recommended)

Check the circle of **Upload your code** → Give **version label** whatever you want
→ then **paste our S3 bucket URL** that we copied in step 4.5

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 9 with Corretto 11 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-1

Source code origin. Maximum size 500 MB

☐ Local file

☒ **Public S3 URL**

s://my-elastic-beackstalk-bucket-2024.s3.ap-south-1.amazonaws.com/vprofile-v2.wa

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

5. Check **first circle button** to create new role → choose **existing key-pair** → in **EC2 instance profile** choose the which we have created previously.

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](#)

linuxkey

EC2 instance profile

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

My-elastic-beanstalk-ec2-role

View permission details

Cancel

Skip to review

Previous

Next

6. If you want to configure other things like **Load Balancer and Autoscaling groups** so click on **Next** or just take all default values of Elastic Beanstalk by clicking on **Skip to review button** and after that **review the changes** then click on **submit button**.

Step 5: Configure updates, monitoring, and logging

[Edit](#)

Updates, monitoring, and logging [Info](#)

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System enhanced	Cloudwatch custom metrics - instance —	Cloudwatch custom metrics - environment —
Log streaming	Retention	Lifecycle
Deactivated	7	false

Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	
false	false	

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](#)

7. Keep in mind that it will take 4–5 minutes for deployment of our application

Elastic Beanstalk > Environments > S3-demo-with-ec2role-env

S3-demo-with-ec2role-env [Info](#)

Actions [Upload and deploy](#)

Environment overview

Health Ok

Domain s3-demo-with-ec2role-env.eba-k5txw3km.ap-south-1.elasticbeanstalk.com

Environment ID [e-msbibifm4n](#)

Application name [s3-demo-with-ec2role](#)

Platform

[Change version](#)

Platform Tomcat 9 with Corretto 11 running on 64bit Amazon Linux 2023/5.1.3

Running version version-1

Platform state Supported

Events (11) [Info](#)

Filter events by text, property or value


Time	Type	Details
February 4, 2024 16:11:22 (UTC+5:30)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 64 seconds ago and took 2 minutes.
February 4, 2024 16:10:41 (UTC+5:30)	INFO	Successfully launched environment: S3-demo-with-ec2role-env
February 4, 2024 16:10:40 (UTC+5:30)	INFO	Application available at S3-demo-with-ec2role-env.eba-k5txw3km.ap-south-1.elasticbeanstalk.com.
February 4, 2024 16:10:22 (UTC+5:30)	INFO	Added instance [i-03738ea180fcd9c4] to your environment.
February 4, 2024 16:10:09 (UTC+5:30)	INFO	Instance deployment completed successfully.
February 4, 2024 16:09:24 (UTC+5:30)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.

8. After deployment you can see **Ok status** that means now you can access the application so **click on Domain** then you will see following application login page

Not secure | s3-demo-with-ec2role-env.eba-k5txw3km.ap-south-1.elasticbeanstalk.com/login

HKH Infotech TECHNOLOGIES ABOUT BLOG LOGIN SIGN UP

WELCOME!

 **HKH Infotech**

Username

Password

[LOGIN](#)

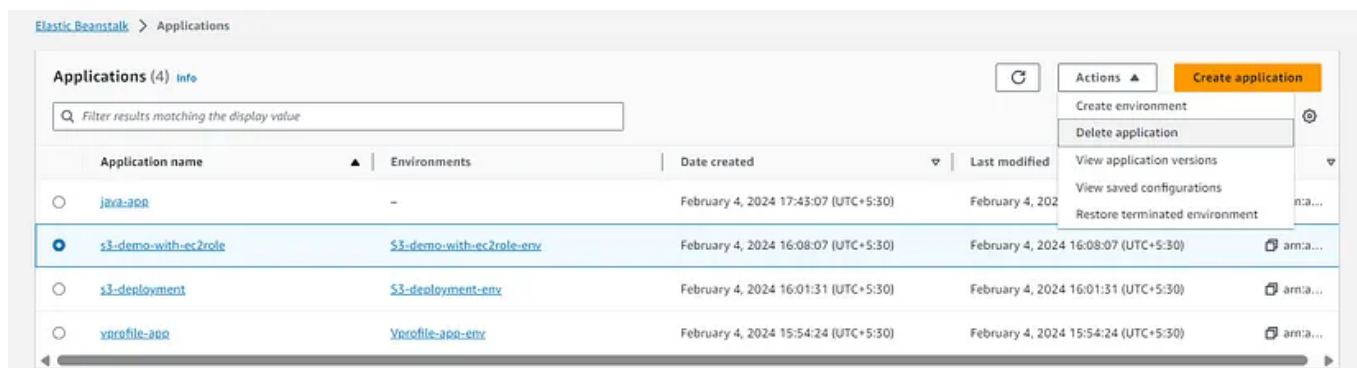
[SIGN UP](#)

Wow you have successfully deployed JAVA Application on AWS Elastic Beanstalk.

Step 6: clean up

Now it's time to clean all the resources. so first click on **application** from left navigation panel in **elastic beanstalk console**.

Select the app to delete → Actions → Delete application



Finally confirm to delete application by typing our application name in the box and click on **Delete**.

