



How To Deploy Sample NodeJS Application With AWS Elastic Beanstalk



Ujwal Pachghare 🌞 · Follow

5 min read · 4 days ago



The original version of this blog is aut0ps.blogspot.com

🤖 What Is Elastic Beanstalk?

AWS Elastic Beanstalk is a fully managed service that simplifies the deployment and management of applications in the AWS Cloud.


In simple way AWS Elastic Beanstalk is a powerful tool for deploying and managing web applications in the AWS Cloud.

What Are Key Benefits Of Using Beanstalk?

- **Easy to Use:** simply upload your application and Elastic Beanstalk automatically handles the provisioning, load balancing, scaling, and application health monitoring.
- **Support Multiple Languages:** Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby.
- **Customizable:** You can change the configuration as needed.
- **Cost:** There is no charge for using Elastic Beanstalk. You pay only for the AWS resources that your application is using.

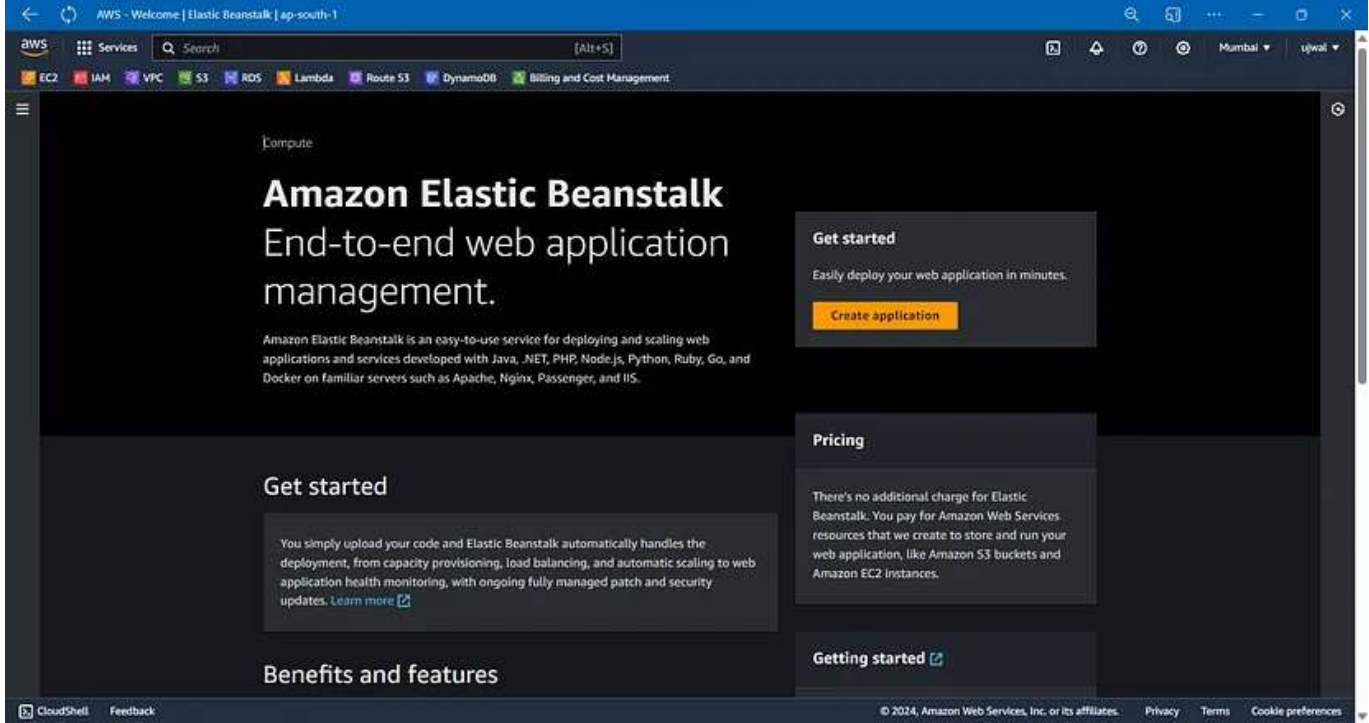
How To Deploy Application on Elastic Beanstalk?

First, it's mandatory to have access of AWS management console. If you have an AWS account so you are ready to start the journey of Elastic Beanstalk.

For deep understanding follow this AWS official documentation [here](#)  The following steps will guide through deploying a web application on AWS Elastic Beanstalk. Let's Start.....

1. Open the Elastic Beanstalk Page

Go to AWS Console → Search for the Elastic Beanstalk → Click on Elastic Beanstalk from dropdown. Great!! Now you will see following page.



2. Create Environment and Deploy our Application

In this demo we will just choose mandatory options to setup an environment for Sample Nodejs Application.

If I sign on the box title with ● that means the full box options, you could leave default. If I sign any option with ● means that option is mandatory or you must edit it according to your needs.

● Mandatory/ Edit it

● Leave Default

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

Environment description

● We chose **Web Server Environment** because we are deploying sample web application of python. This option can be used for serving HTTP request for websites, web applications and web APIs

● **Worker Application** this option can be used for creating separate worker environment tier.

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

Node.js

Platform branch

Node.js 20 running on 64bit Amazon Linux 2023

Platform version

6.1.0 (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

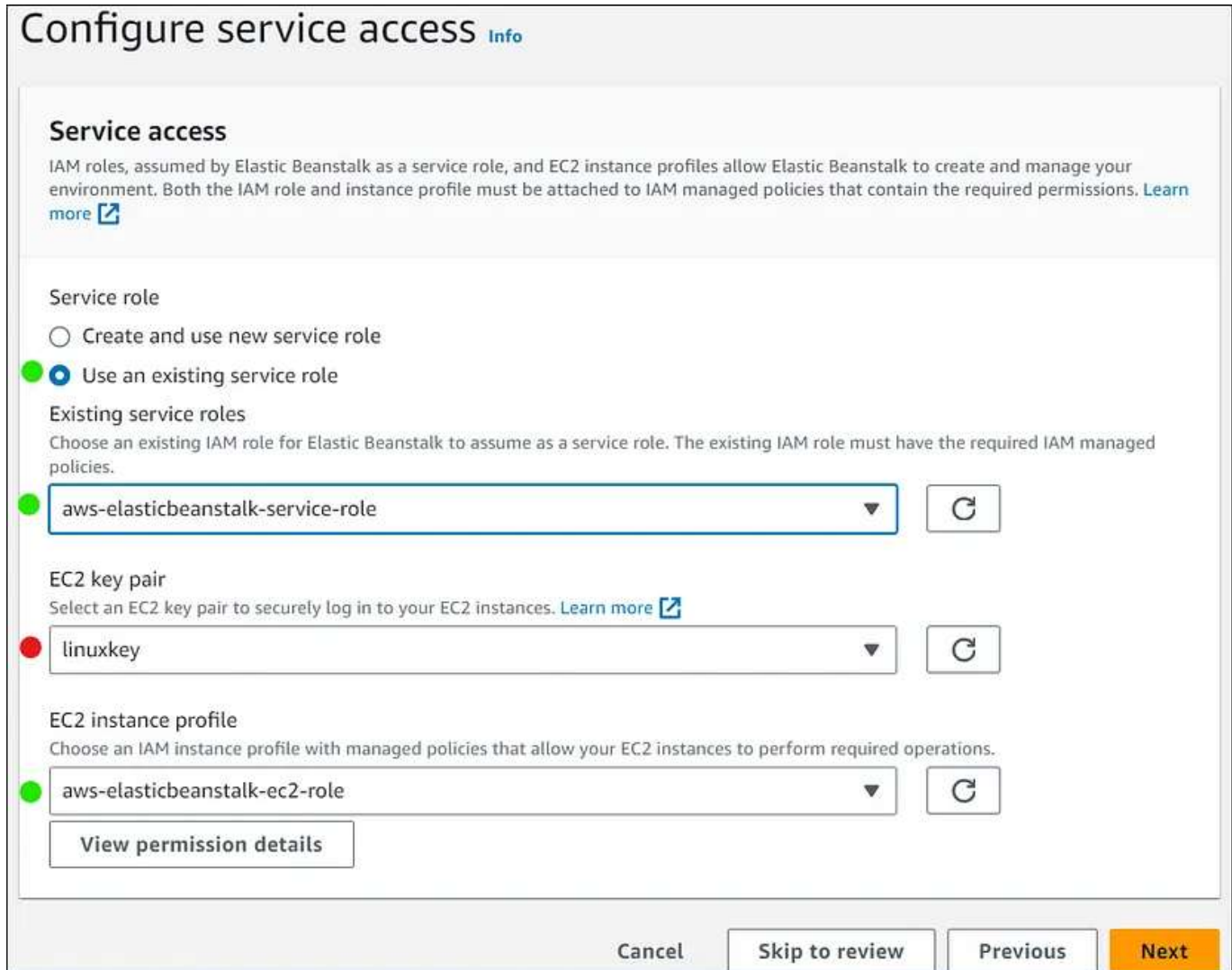
Cancel Next

● **Managed Platform** means it will be maintained and managed by AWS Elastic Beanstalk automatically.

● **Platform** means in which programming language runtime environment we want to run our application. we can also choose **Platform version**.

● **Application Code:** here we can upload our app code from local machine or can get from S3 bucket. we are using **sample app** code which will be managed AWS Elastic Beanstalk.

● **Presets:** here we can choose our instances with deferent configurations or can create our custom.



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

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

●

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.

●

● **Use an existing service role** means the IAM Role which is already present with all mandatory permissions. If there is now existing role then you can choose **Create and use new service role** options, it will create new role with all mandatory permissions.

● **Choose EC2 key pair** If you wanna SSh into instance in the future. its not mandatory.

● **EC2 Instance Profile** means a role with use case EC2 and which have all the mandatory permissions. choose existing/default role , in case there is no

any existing/default role. you can create role by following demo which is at the last of this blog.↓

That`s it, for Sample Application purpose these things are enough. now just click on **Skip and review** button at the last of the page. after clicking on button, you will be redirected to the following page review all the things and click on **Submit button** at the last of the page.

Review [Info](#)

Step 1: Configure environment

Edit

Environment information

Environment tier	Application name
Web server environment	NodeApp
Environment name	Application code
NodeApp-env	Sample application
Platform	
arn:aws:elasticbeanstalk:ap-south-1::platform/Node.js 20 running on 64bit Amazon Linux 2023/6.1.0	

Step 2: Configure service access

Edit

Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
--------------	--------------	----------------------

👉 Wait for 3–4 minutes. after 3–4 minutes you will see **Ok health status** and **domain name** of our application. now click on the **domain name**.

Elastic Beanstalk > Environments > NodeApp-env

NodeApp-env Info

Environment overview

Health

● Ok - View causes

Domain

● NodeApp-env.eba-knhhqua2.ap-south-1.elasticbeanstalk.com [🔗](#)

Environment ID

📄 e-xgddptrzcd

Application name

NodeApp

Platform

Change version

Platform

Node.js 20 running on 64bit Amazon Linux 2023/6.1.0

Running version

-

Platform state

✔ Supported

Events Health Logs Monitoring Alarms Managed updates Tags

Events (10) Info

Time

Type

Details

→ After click on domain name you will see following our application page in the browser. Congrats 🎉 you have deployed Sample NodeJS application with AWS Elastic Beanstalk.

← ↻ 🏠 ⚠ Not secure | nodeapp-env.eba-knhhqua2.ap-south-1.elasticbeanstalk.com

Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?


- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploying an Express Application to AWS Elastic Beanstalk](#)
- [Deploying an Express application with clustering to Elastic Beanstalk](#)
- [Customizing and Configuring a Node.js Container](#)
- [Working with Logs](#)

Cleaning Up

1. Go to AWS Elastic Beanstalk Console →: Click on **Application** from left side menu →: Select our Application by checking box → Click on **Action** menu → Click on **Delete Application** → you can see following box → type our app name → Click on **Delete**

Confirm application deletion ×

Permanently delete **NodeApp**? This action can't be undone

 If you proceed with this action, the following environments will be terminated:

- NodeApp-env

Enter the name of the application to confirm:

CancelDelete

? Elastic Beanstalk FAQs

1. Who should use AWS Elastic Beanstalk?

Those who want to deploy and manage their applications within minutes in the AWS Cloud.

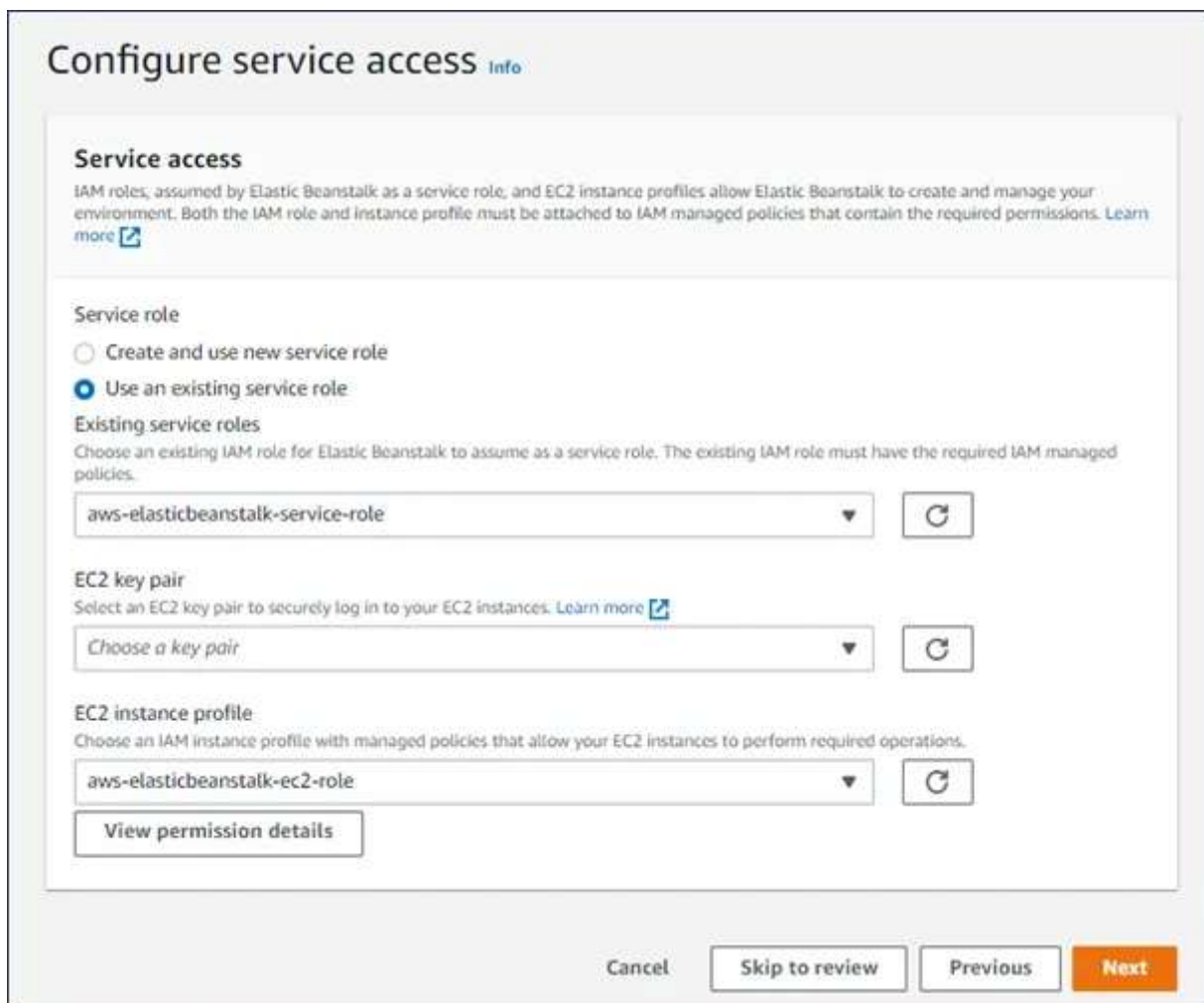
2. Which Operating Systems Does AWS Elastic Beanstalk Use?

AWS Elastic Beanstalk runs on the Amazon Linux AMI and the Windows Server AMI.

3. What Kinds Of Applications Are Supported By Aws Elastic Beanstalk?

AWS Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker, and it is best for web applications. and non-web applications can also be deployed using Elastic Beanstalk.

Create IAM Role For EC2 Instance Profile



The screenshot shows the 'Configure service access' page in the AWS IAM console. The page title is 'Configure service access' with an 'Info' link. Below the title is a section titled 'Service access' with a description: '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](#)'. There are three main sections: 'Service role', 'EC2 key pair', and 'EC2 instance profile'. The 'Service role' section has two radio buttons: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected). Below this is a dropdown menu for 'Existing service roles' with the text 'Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.' The dropdown is currently set to 'aws-elasticbeanstalk-service-role'. The 'EC2 key pair' section has a dropdown menu with the text 'Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)'. The dropdown is currently set to 'Choose a key pair'. The 'EC2 instance profile' section has a dropdown menu with the text 'Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.' The dropdown is currently set to 'aws-elasticbeanstalk-ec2-role'. There is a 'View permission details' button below the EC2 instance profile dropdown. At the bottom of the page are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Configure service access [Info](#)

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

Service role
☐ Create and use new service role
☒ Use an existing service role

Existing service roles
Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

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.

aws-elasticbeanstalk-ec2-role

[View permission details](#)

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

→ To create an IAM Role for EC2 instance profile selection

1. Go to IAM console

2. In the IAM console navigation pane, choose **Roles**.
3. Choose **Create role**.
4. Under **Trusted entity type**, choose **AWS service**.
5. Under **Use case**, choose **EC2**.
6. Choose **Next**.
7. Attach the appropriate managed policies. Scroll in the **View instance profile permissions** modal window to see the managed policies. The policies are also listed here:

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Permissions policies (3/947) [Info](#)

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

Search: Filter by Type: 14 matches

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permissions. Explicitly allows develo...
<input type="checkbox"/>	AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	Provide the instance in your custom platform builder environment ...
<input type="checkbox"/>	AWSElasticBeanstalkEnhancedHealth	AWS managed	AWS Elastic Beanstalk Service policy for Health Monitoring system
<input type="checkbox"/>	AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	This policy is for the AWS Elastic Beanstalk service role used to perf...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicontainer Docker environment ac...
<input type="checkbox"/>	AWSElasticBeanstalkReadOnly	AWS managed	Grants read-only permissions. Explicitly allows operators to gain dir...
<input type="checkbox"/>	AWSElasticBeanstalkRoleCore	AWS managed	AWSElasticBeanstalkRoleCore (Elastic Beanstalk operations role) All...
<input type="checkbox"/>	AWSElasticBeanstalkRoleCWL	AWS managed	(Elastic Beanstalk operations role) Allows an environment to manag...
<input type="checkbox"/>	AWSElasticBeanstalkRoleEC2	AWS managed	(Elastic Beanstalk operations role) Allows a multicontainer Docker e...
<input type="checkbox"/>	AWSElasticBeanstalkRoleRDS	AWS managed	(Elastic Beanstalk operations role) Allows an environment to integr...
<input type="checkbox"/>	AWSElasticBeanstalkRoleSNS	AWS managed	(Elastic Beanstalk operations role) Allows an environment to enable...
<input type="checkbox"/>	AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allows a worker environment tier...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server environment access to upl...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker environment access to upload l...

1. Choose **Next**.
2. Enter a name for the role.

3. (Optional) Add tags to the role.

4. Choose **Create role**.

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

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

Description
Add a short explanation for this role.

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


Step 1: Select trusted entities

Trust policy

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

Step 2: Add permissions

Permissions policy summary

Policy name 

[AWSElasticBeanstalkMulticontainerDocker](#)

[AWSElasticBeanstalkWebTier](#)

[AWSElasticBeanstalkWorkerTier](#)

AWS

Elastic Beanstalk Deploy

Web Applications

Nodejs Aws Deploy

Aws Elastic Beanstalk