AWS Elastic Beanstalk
End-toEnd Web application management

Cloud services ---> important services
IaaS ---> ex: EC2, S3, RDS, VPC
PaaS ---> Platform as a Service ex: Elastic Beanstalk
SaaS ---> Software as a Service (Gmail, Zoom, Teams)

Web Application ---> Deployment

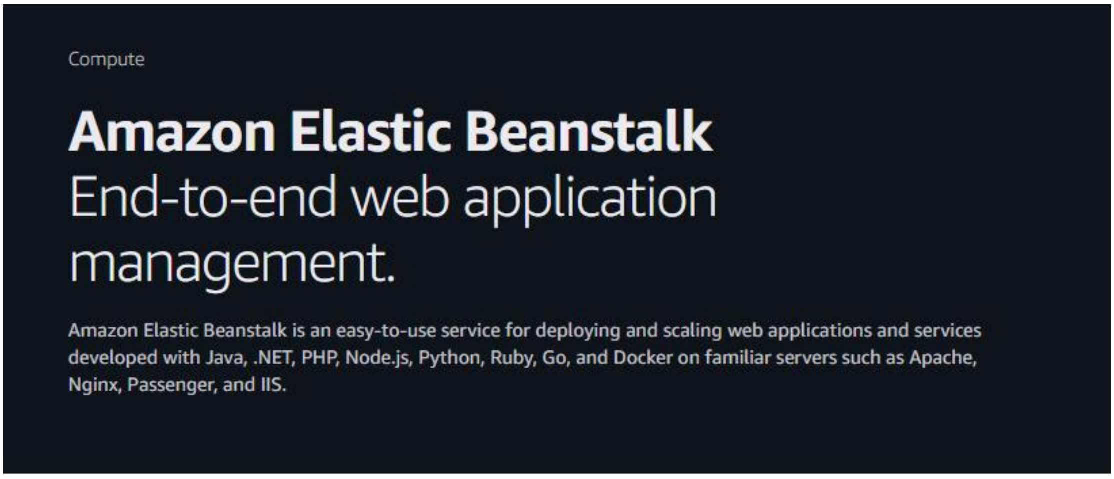
We need the following:

1. Create Network VPC
2. Create Security Groups
3. Enable Inbound Rules
4. Create EC2 instances
5. Install required Software/dependencies on EC2 to run our appn code --> Ex: Java, Tomcat, IIS
6. Create Load balancer
7. Setup Autoscaling groups for high availability
8. Deploy our application

If you use PaaS, you need not do the above steps manually, Elastic Beanstalk will take care of it
You only need to take care of Application code, other steps will be taken care by AWS itself

Elastic Beanstalk: it provides Platform as a Service ---> AWS will provide read-made platform to run our application

Note: Whenever we go with AWS Elastic Beanstalk, then the first 7 steps of given list will be taken care by Elastic Beanstalk and we will be taking care only Application deployment not everything. No specific or fixed charges/price for Elastic Beanstalk, it depends on user-case by use-case. Many resources are available like EC2 instances, Load balancers, AutoScaling Group etc



Compute

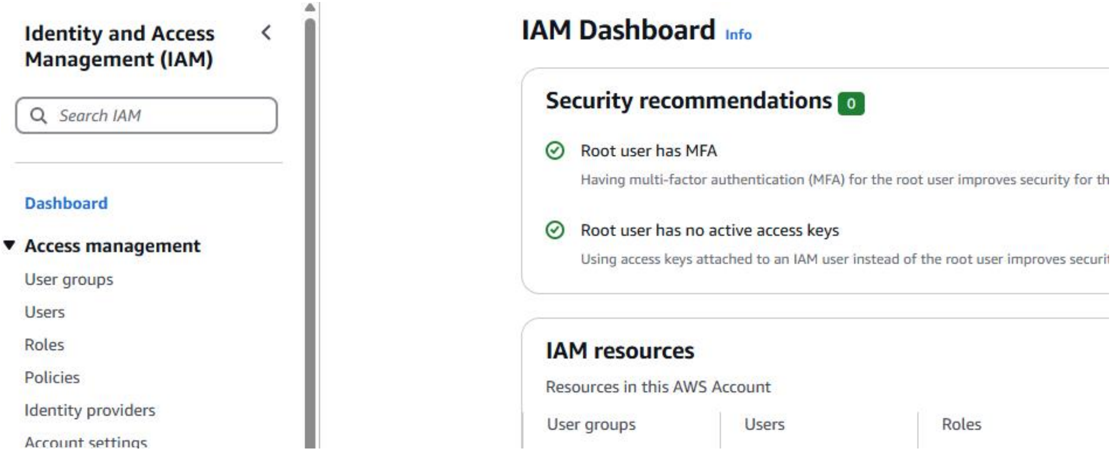
Amazon Elastic Beanstalk
End-to-end web application management.

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.

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.

If you want to use EBS, we need to create an IAM role. To access one service from another service, we need IAM role

Step 1 is IAM



Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

IAM Dashboard Info

Security recommendations 0

✓ Root user has MFA

Having multi-factor authentication (MFA) for the root user improves security for th

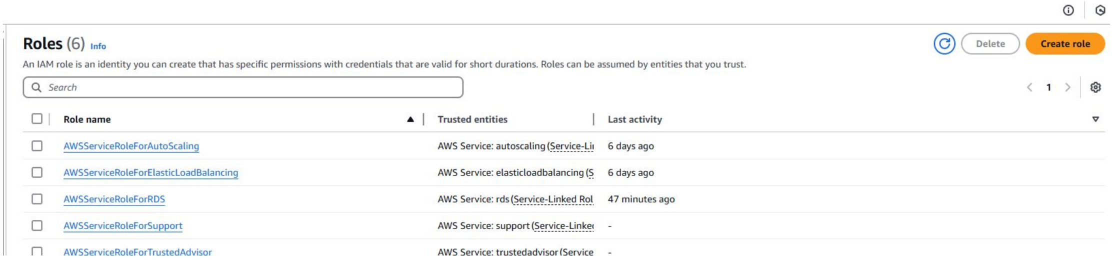
✓ Root user has no active access keys

Using access keys attached to an IAM user instead of the root user improves securi

IAM resources

Resources in this AWS Account

User groups	Users	Roles



Roles (6) Info

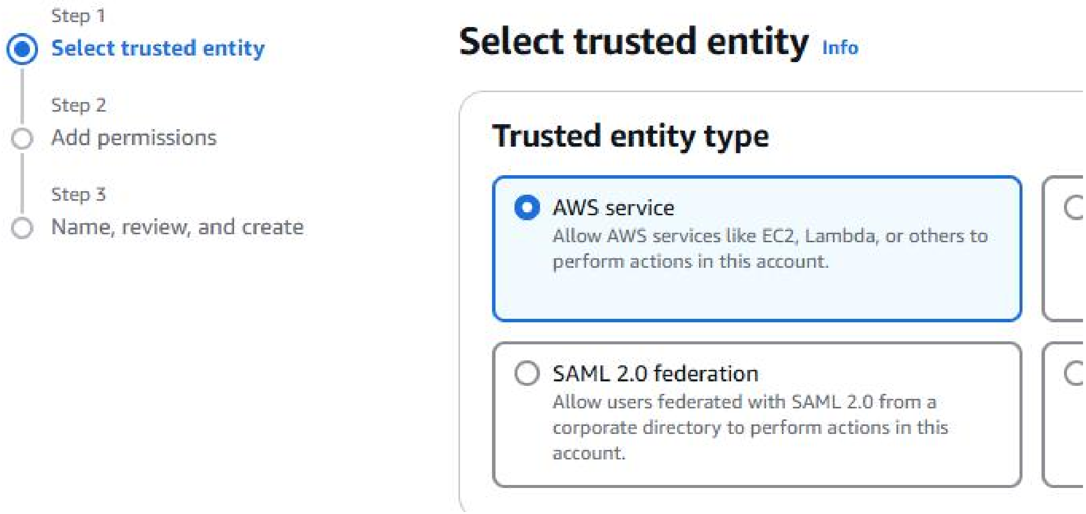
Delete | Create role

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.

Search

□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Li	6 days ago
□	AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S	6 days ago
□	AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Rol	47 minutes ago
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

Create Role



Step 1

● Select trusted entity

Step 2

○ Add permissions

Step 3

○ Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service

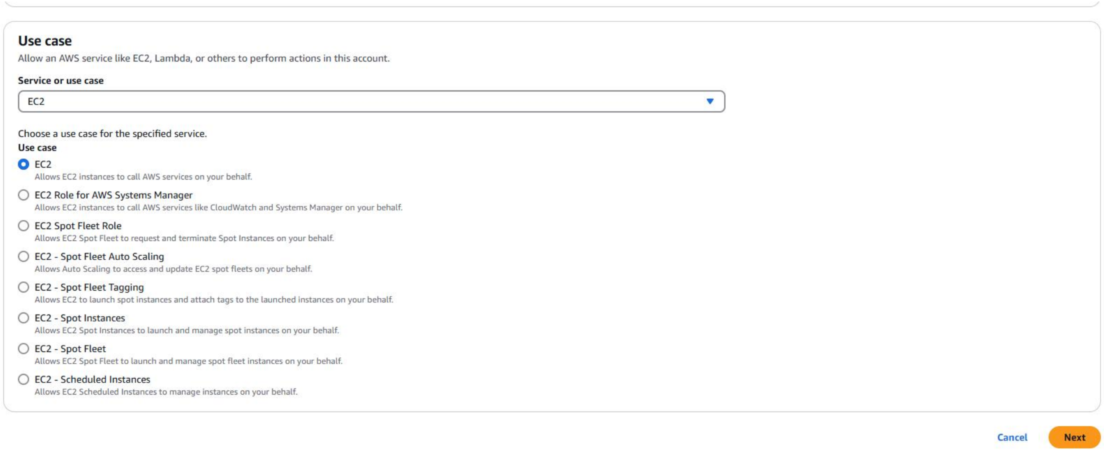
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ SAML 2.0 federation

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

Select AWS service

EC2, Click Next



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.

○ EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

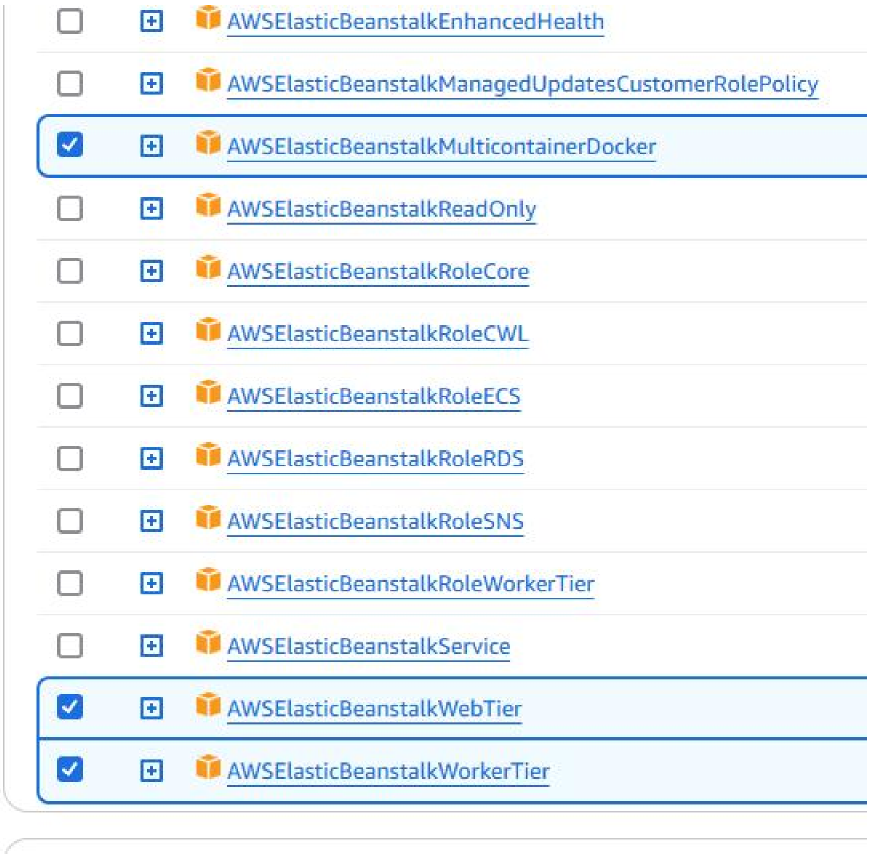
○ EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

○ 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.

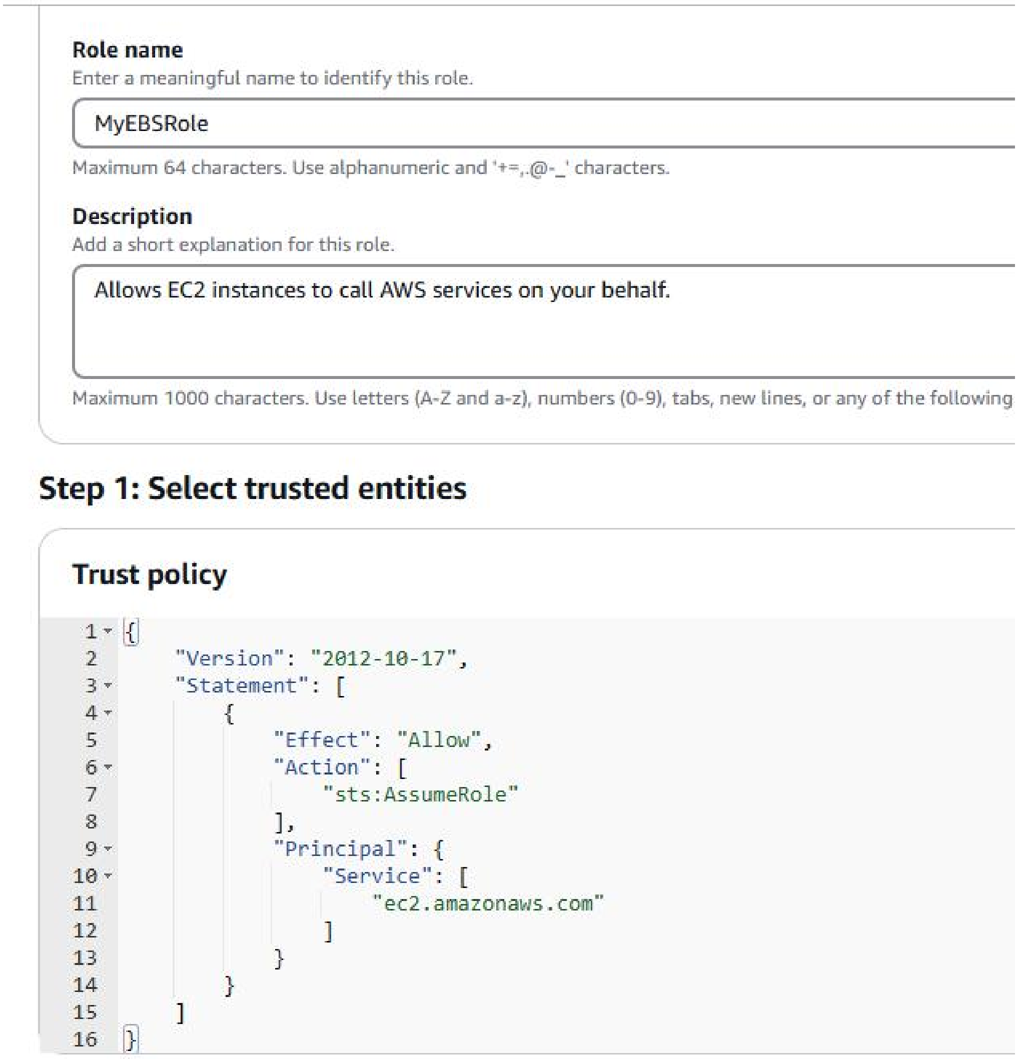
Note: this EC2 requires 3 minimum permissions

1. AWSElasticBeanstalkMulticontainerDocker
2. AWSElasticBeanstalkWebTier
3. AWSElasticBeanstalkWorkerTier



- □ AWSElasticBeanstalkEnhancedHealth
- □ AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy
- ☑ AWSElasticBeanstalkMulticontainerDocker
- □ AWSElasticBeanstalkReadOnly
- □ AWSElasticBeanstalkRoleCore
- □ AWSElasticBeanstalkRoleCWL
- □ AWSElasticBeanstalkRoleECS
- □ AWSElasticBeanstalkRoleRDS
- □ AWSElasticBeanstalkRoleSNS
- □ AWSElasticBeanstalkRoleWorkerTier
- □ AWSElasticBeanstalkService
- ☑ AWSElasticBeanstalkWebTier
- ☑ AWSElasticBeanstalkWorkerTier

Click Next



Role name

Enter a meaningful name to identify this role.

MyEBSRole

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.

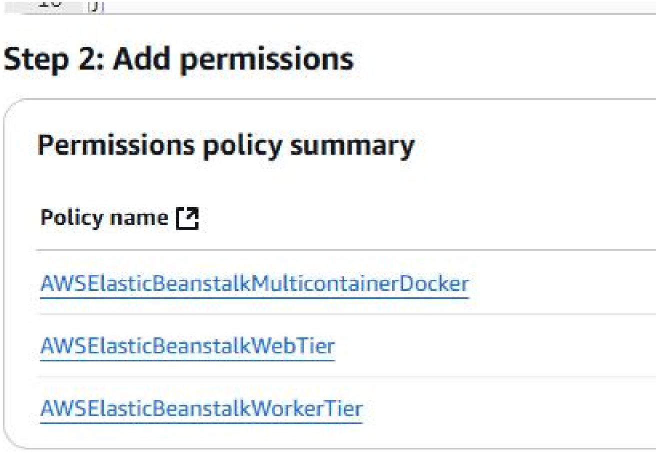
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    ]
}
```

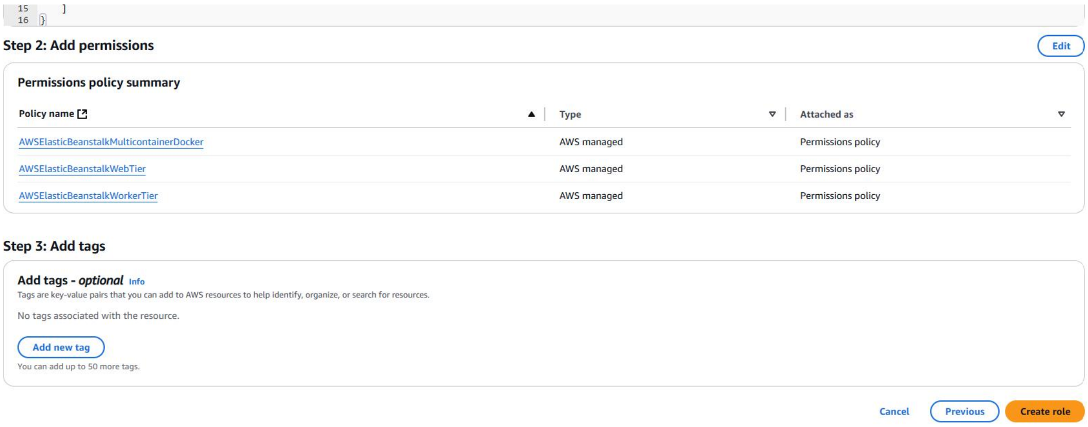
Double check the 3 policies discussed above



Step 2: Add permissions

Permissions policy summary

Policy name
AWSElasticBeanstalkMulticontainerDocker
AWSElasticBeanstalkWebTier
AWSElasticBeanstalkWorkerTier



```
    ]
}
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

Step 3: Add tags

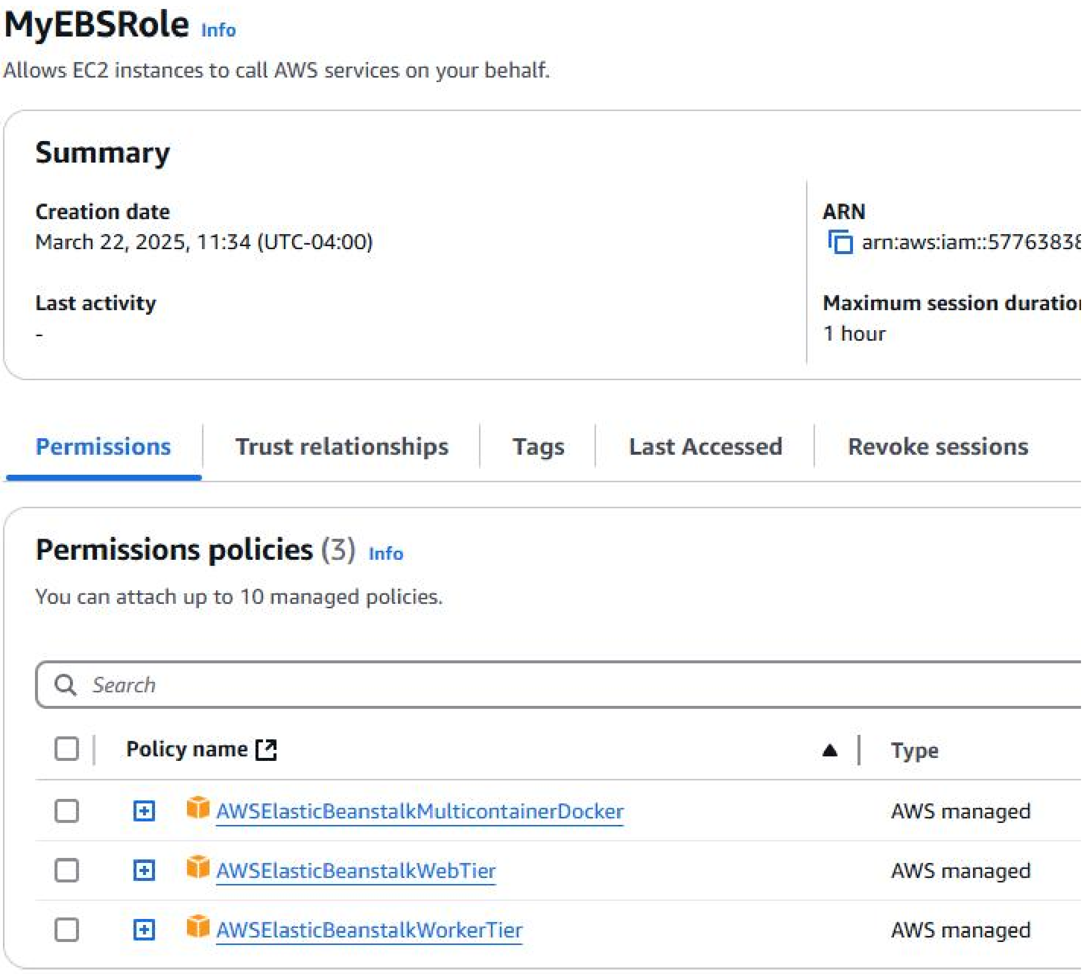
Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

You can add up to 50 more tags.

Click Create Role



MyEBSRole Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
March 22, 2025, 11:34 (UTC-04:00)

Last activity
-

ARN
arn:aws:iam::57763838

Maximum session duratio
1 hour

Permissions policies (3) Info

You can attach up to 10 managed policies.

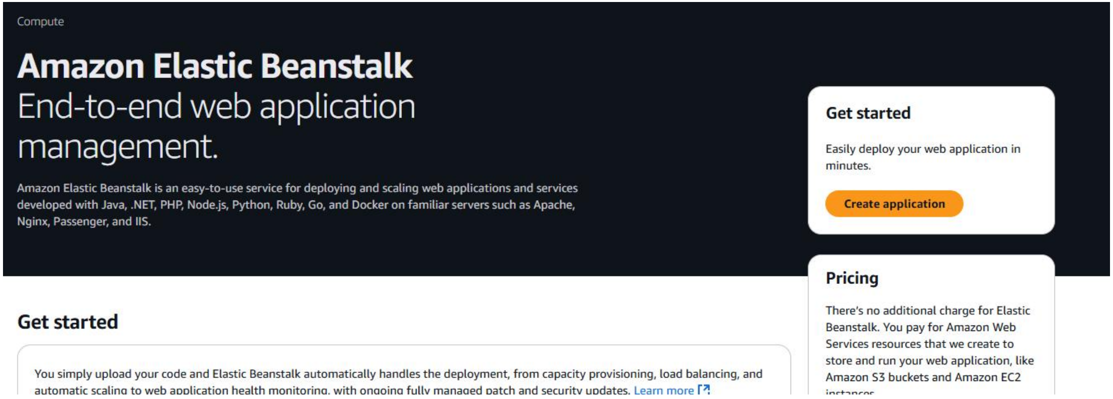
□	Policy name	Type
□	AWSElasticBeanstalkMulticontainerDocker	AWS managed
□	AWSElasticBeanstalkWebTier	AWS managed
□	AWSElasticBeanstalkWorkerTier	AWS managed

Lab/Practical Task on Elastic Beanstalk

1. Create an IAM Role with following policies:
    a) AWSElasticBeanstalkMulticontainerDocker
    b) AWSElasticBeanstalkWebTier
    c) AWSElasticBeanstalkWorkerTier
2. Create Application using Elastic Beanstalk
3. Create Environment for the application by choosing required Runtime

4. After the environment is created, it will generate DNS to access application

Go back to EBS page



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

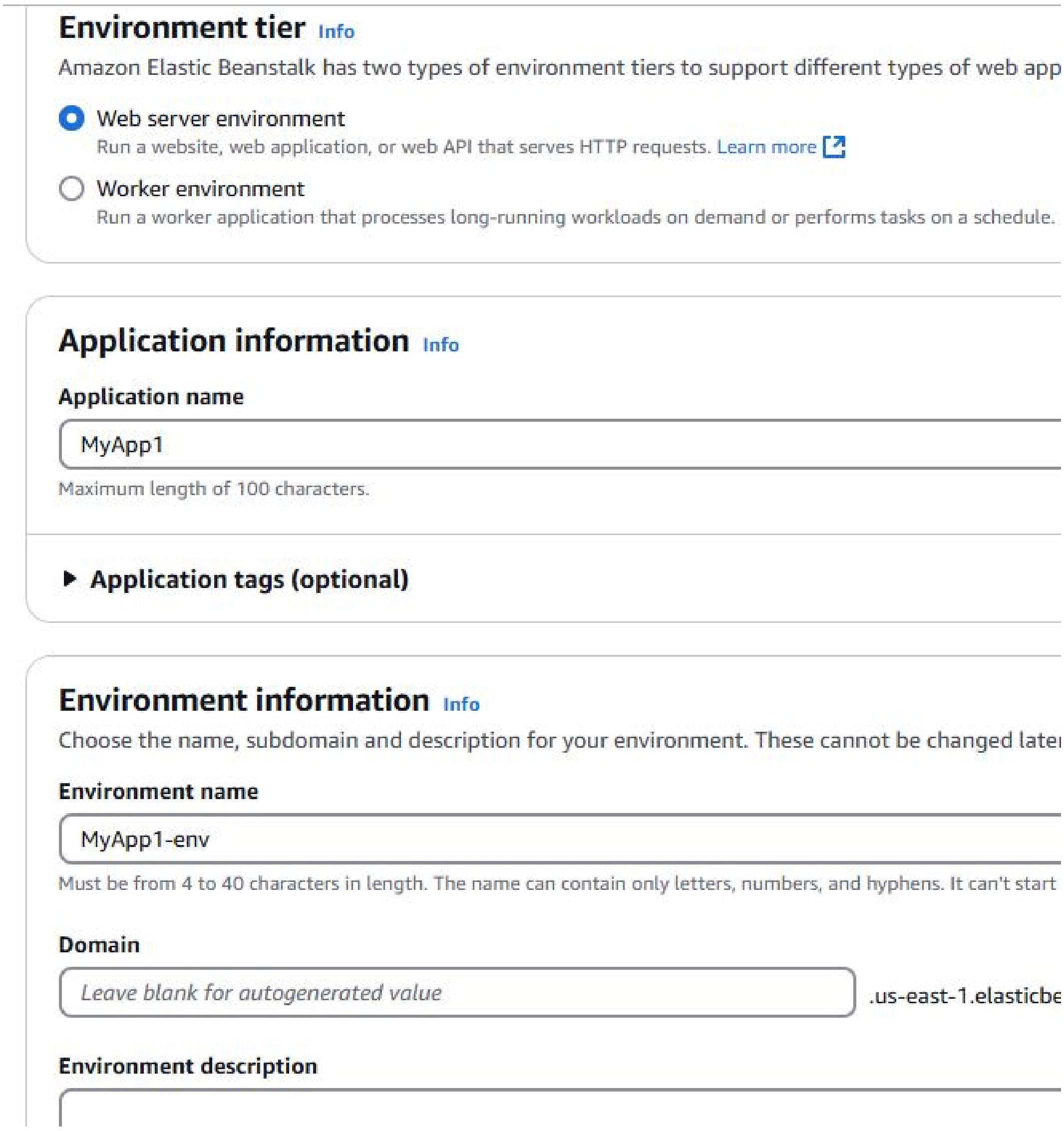
There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances

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

Create application

Click Web server environment



Environment tier Info

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

● 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.

Application information Info

Application name

MyApp1

Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed late

Environment name

MyApp1-env

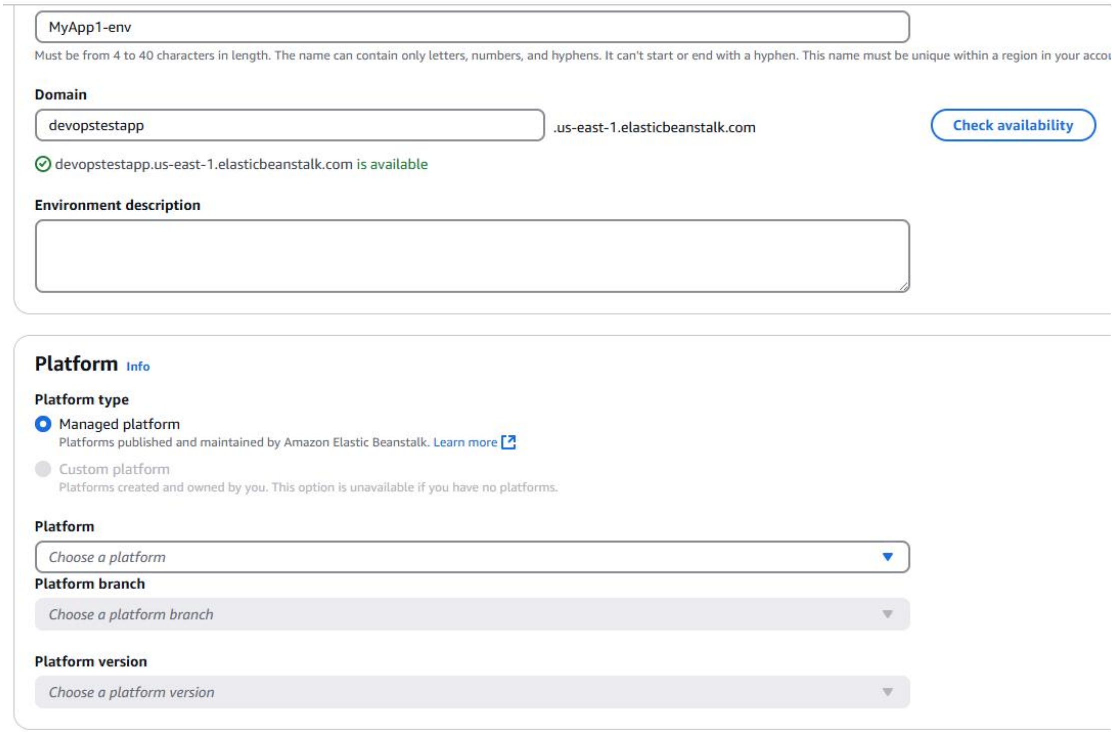
Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start

Domain

Leave blank for autogenerated value .us-east-1.elasticbe

Environment description

Enter Domain: devopstestapp



MyApp1-env

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 accou

Domain

devopstestapp .us-east-1.elasticbeanstalk.com

devopstestapp.us-east-1.elasticbeanstalk.com is available

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

Choose a platform

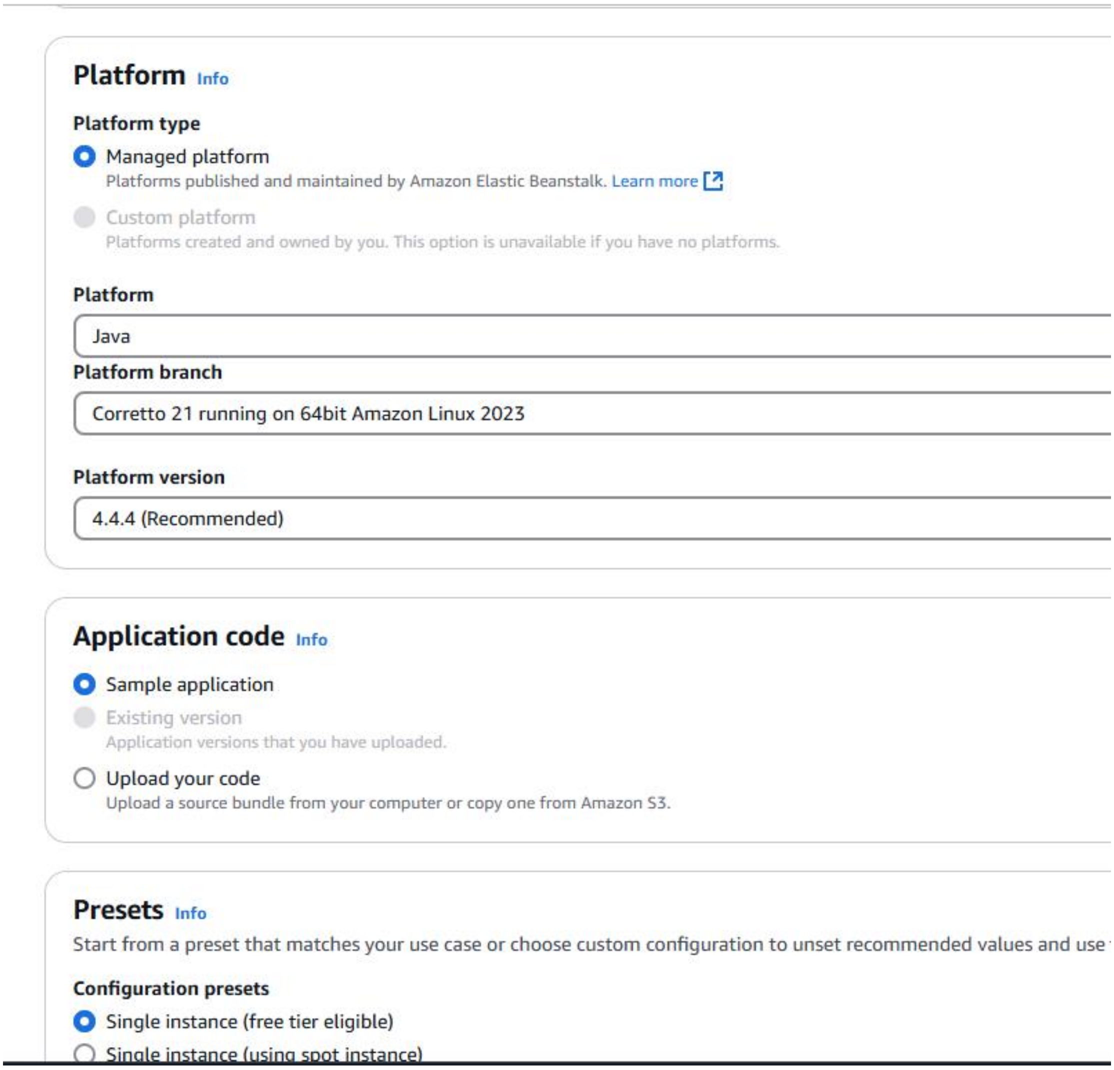
Platform branch

Choose a platform branch

Platform version

Choose a platform version

Select Platform Java for now



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

Java

Platform branch

Corretto 21 running on 64bit Amazon Linux 2023

Platform version

4.4.4 (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.

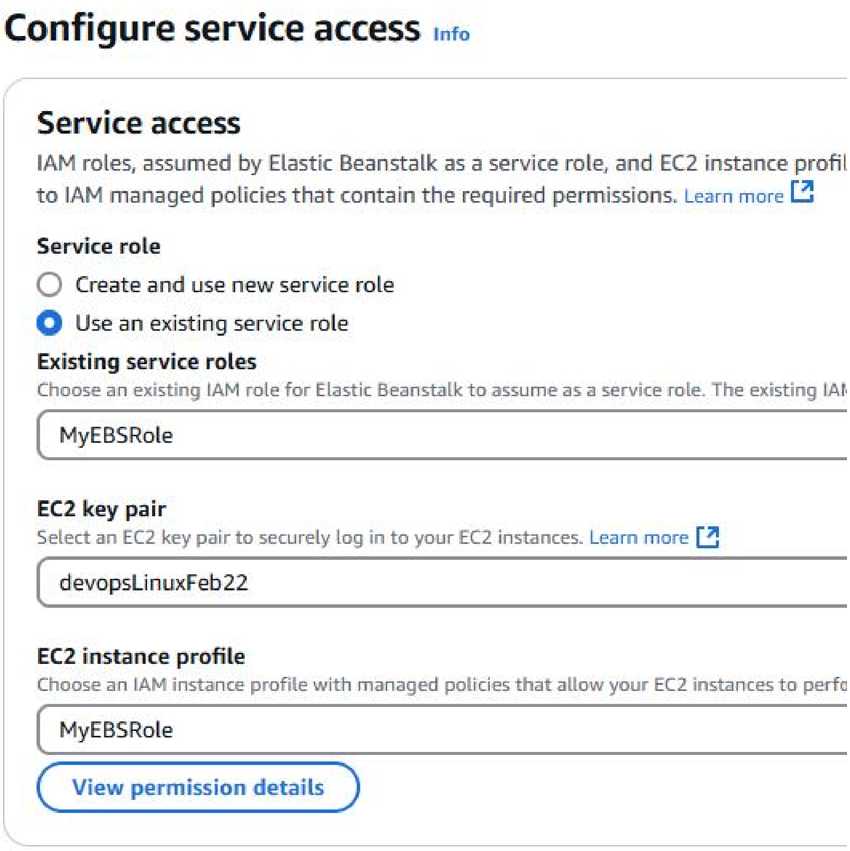
Presets Info

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use

Configuration presets

● Single instance (free tier eligible)

○ Single instance (using spot instance)



Configure service access Info

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profil
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 IAI

MyEBSRole

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

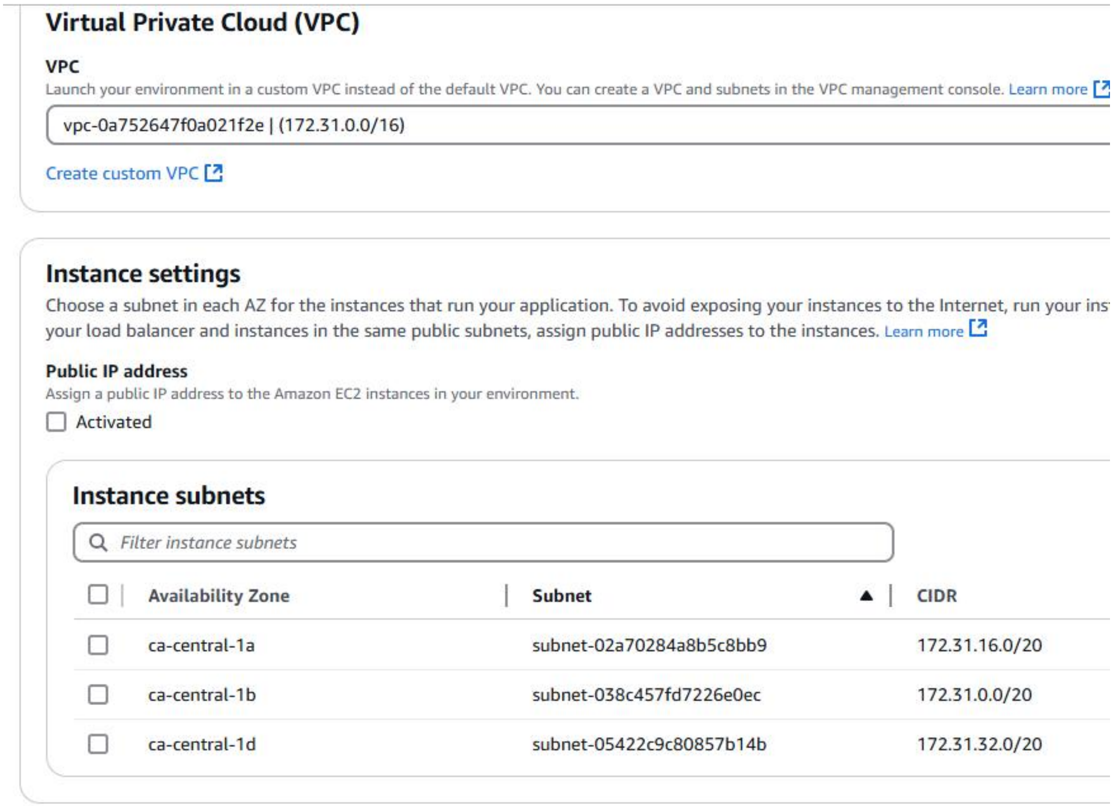
devopsLinuxFeb22

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perfo

MyEBSRole

Select default options only



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-0a752647f0a021f2e | (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 ins
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.

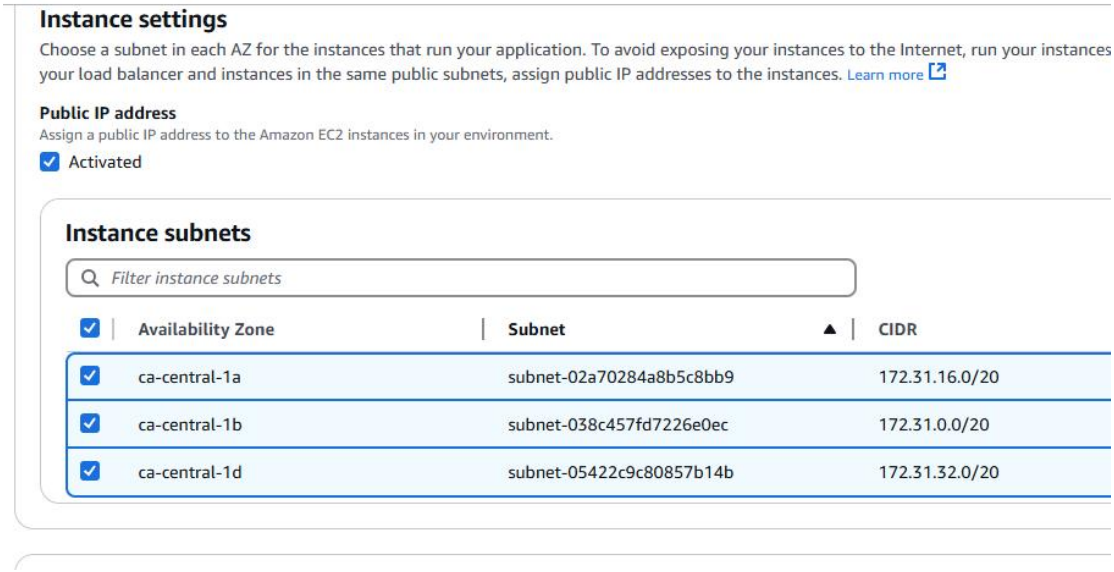
□ Activated

Instance subnets

Filter instance subnets

□	Availability Zone	Subnet ▲	CIDR
□	ca-central-1a	subnet-02a70284a8b5c8bb9	172.31.16.0/20
□	ca-central-1b	subnet-038c457fd7226e0ec	172.31.0.0/20
□	ca-central-1d	subnet-05422c9c80857b14b	172.31.32.0/20

Select Activated



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
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.

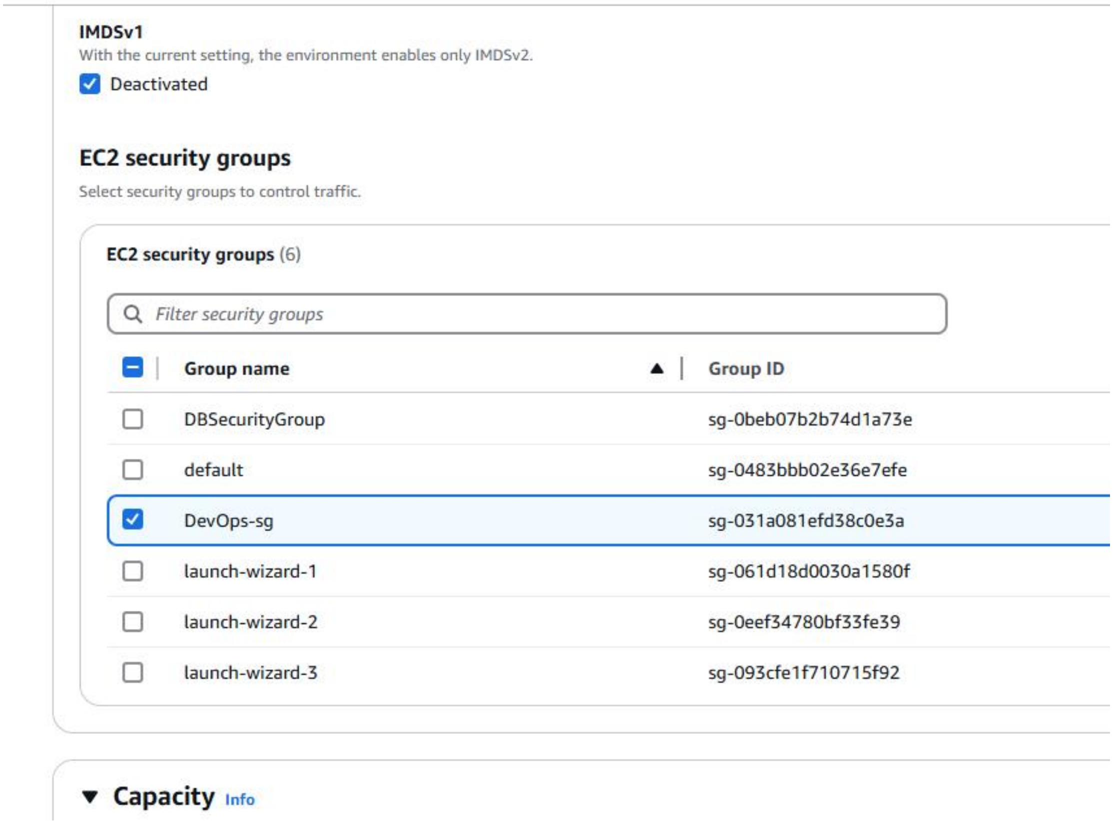
☑ Activated

Instance subnets

Filter instance subnets

☑	Availability Zone	Subnet ▲	CIDR
☑	ca-central-1a	subnet-02a70284a8b5c8bb9	172.31.16.0/20
☑	ca-central-1b	subnet-038c457fd7226e0ec	172.31.0.0/20
☑	ca-central-1d	subnet-05422c9c80857b14b	172.31.32.0/20

Select Security group we already created



IMDSv1

With the current setting, the environment enables only IMDSv2.

☑ Deactivated

EC2 security groups

Select security groups to control traffic.

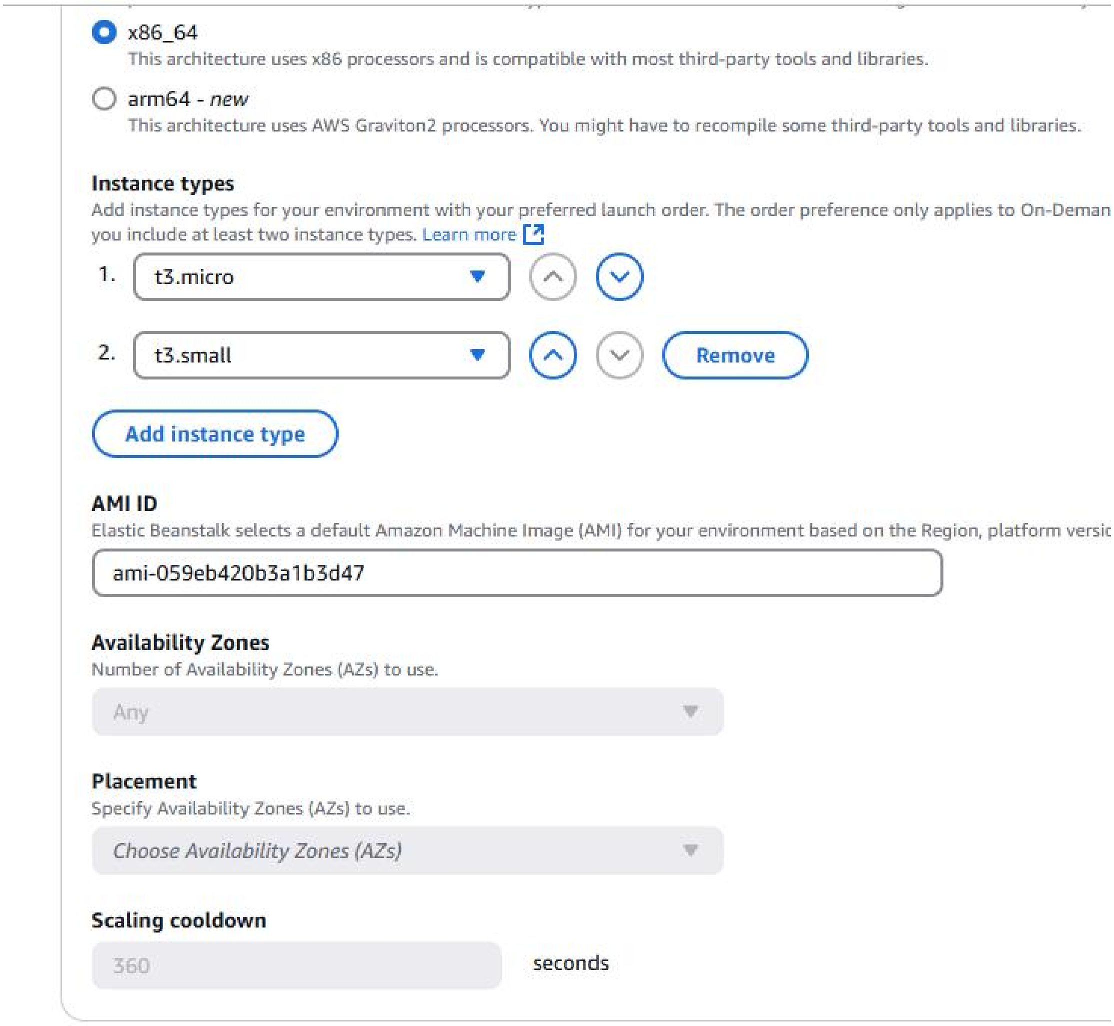
EC2 security groups (6)

Filter security groups

⊟	Group name ▲	Group ID
□	DBSecurityGroup	sg-0beb07b2b74d1a73e
□	default	sg-0483bbb02e36e7efe
☑	DevOps-sg	sg-031a081efd38c0e3a
□	launch-wizard-1	sg-061d18d0030a1580f
□	launch-wizard-2	sg-0eef34780bf33fe39
□	launch-wizard-3	sg-093cfe1f710715f92

▼ Capacity Info

Keep these things default



● 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 environment with your preferred launch order. The order preference only applies to On-Deman
you include at least two instance types. Learn more

1. t3.micro
2. t3.small

AMI ID
Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform versic

ami-059eb420b3a1b3d47

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

Any

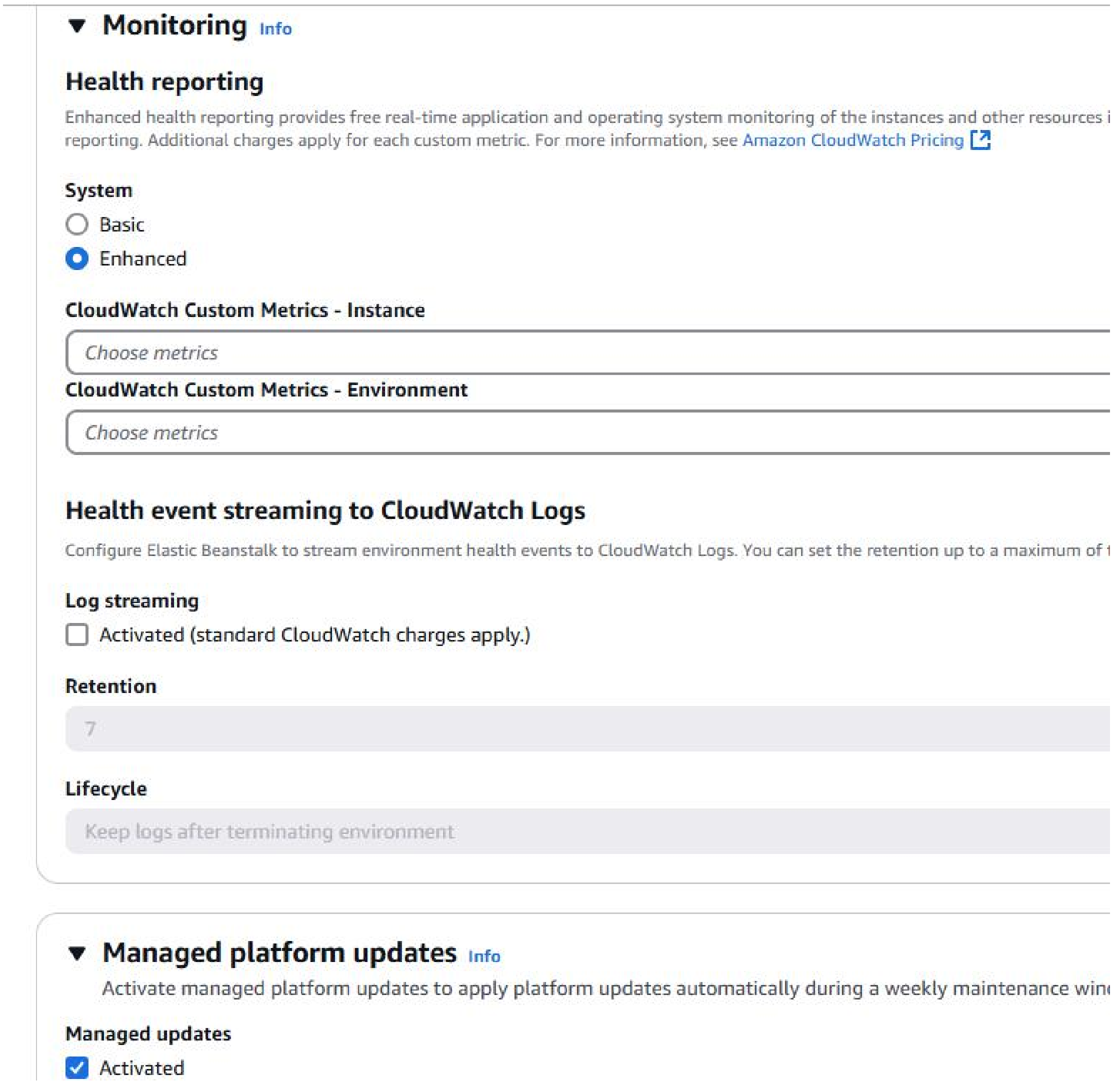
Placement
Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs)

Scaling cooldown

360 seconds

All default -> click Next



Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources i 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 1

Log streaming

□ Activated (standard CloudWatch charges apply.)

Retention

7

Lifecycle

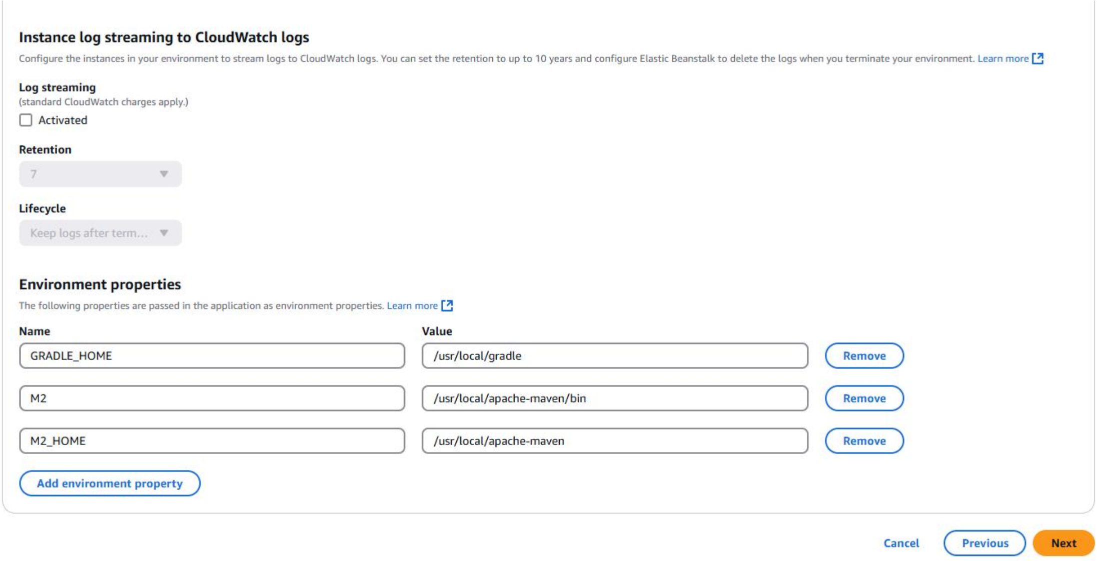
Keep logs after terminating environment

Managed platform updates Info

Activate managed platform updates to apply platform updates automatically during a weekly maintenance win

Managed updates

☑ Activated



Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more

Log streaming

(standard CloudWatch charges apply.)

□ Activated

Retention

7

Lifecycle

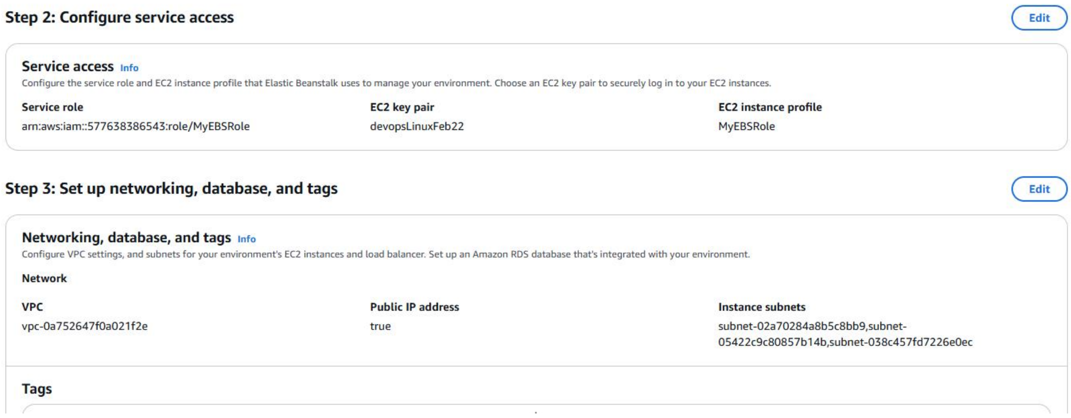
Keep logs after term...

Environment properties

The following properties are passed in the application as environment properties. Learn more

Name	Value	
GRADLE_HOME	/usr/local/gradle	Remove
M2	/usr/local/apache-maven/bin	Remove
M2_HOME	/usr/local/apache-maven	Remove

Review our configuration



Step 2: Configure service access

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
arn:aws:iam::577638386543:role/MyEBSRole	devopsLinuxFeb22	MyEBSRole

Step 3: Set up networking, database, and tags

Networking, database, and tags Info

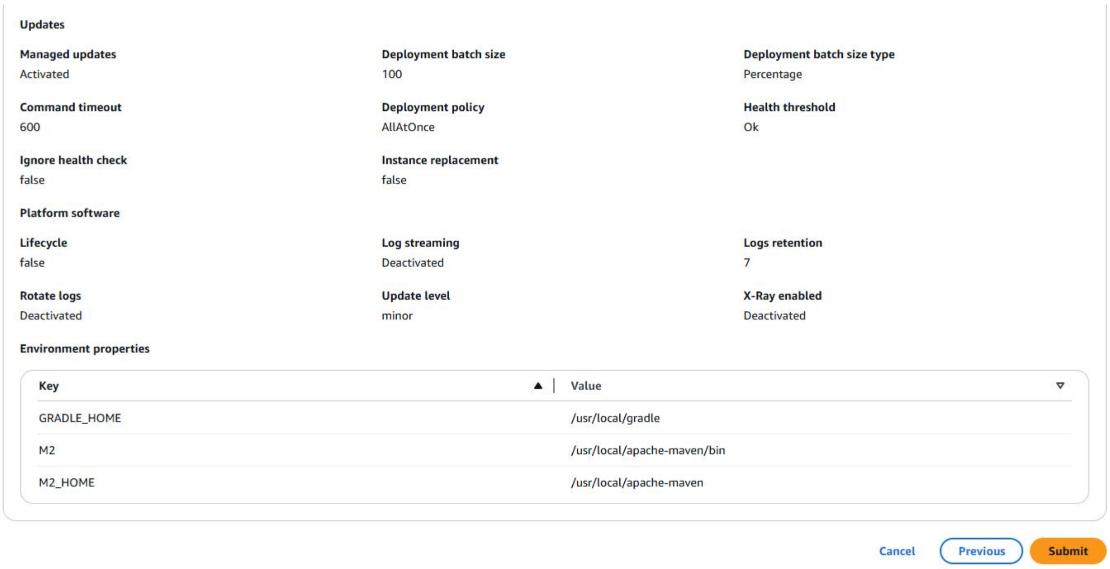
Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

VPC	Public IP address	Instance subnets
vpc-0a752647f0a021f2e	true	subnet-02a70284a8b5c8bb9,subnet-05422c9c80857b14b,subnet-038c457fd7226e0ec

Tags

Click Submit



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

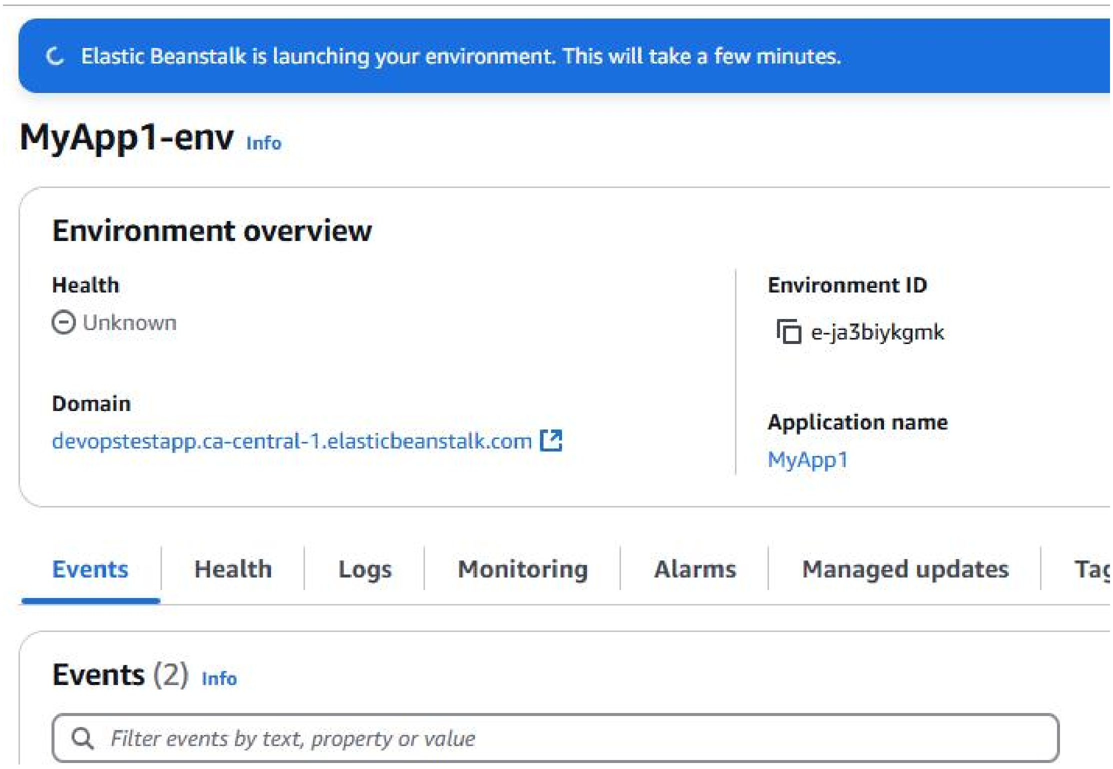
Lifecycle	Log streaming	Logs retention
false	Deactivated	7

Rotate logs	Update level	X-Ray enabled
Deactivated	minor	Deactivated

Environment properties

Key ▲	Value ▽
GRADLE_HOME	/usr/local/gradle
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven

My application will be available on this domain



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

MyApp1-env

Environment overview

Health
Unknown

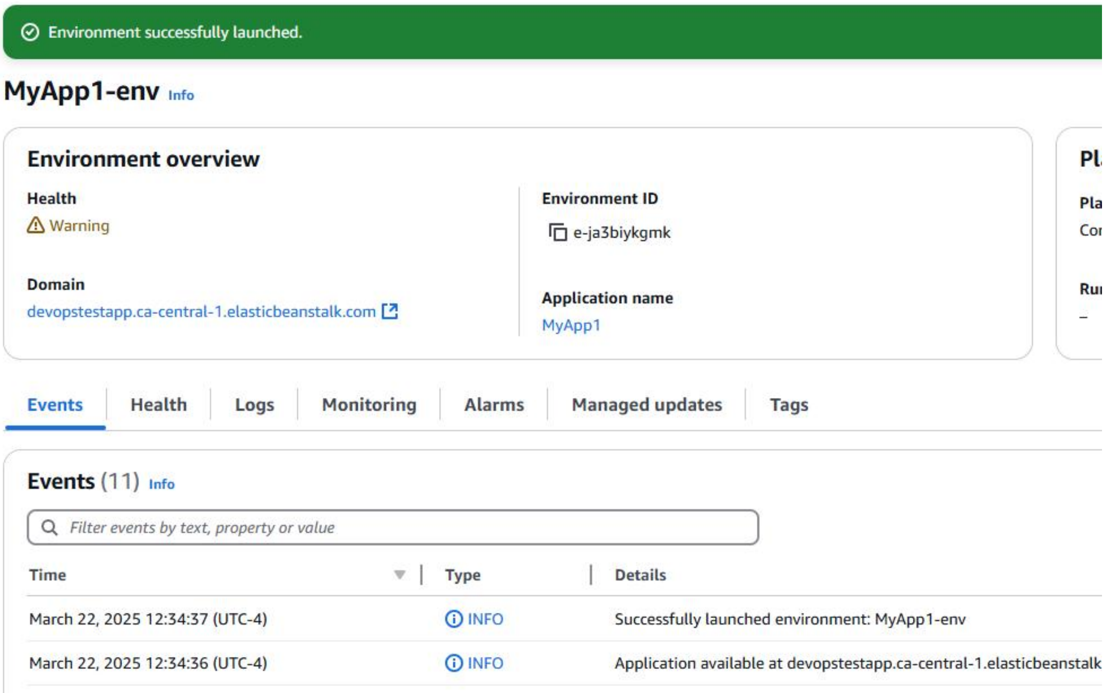
Domain
devopstestapp.ca-central-1.elasticbeanstalk.com

Environment ID
e-ja3biykgmk

Application name
MyApp1

Events (2)

Successfully launched



Environment successfully launched.

MyApp1-env

Environment overview

Health
Warning

Domain
devopstestapp.ca-central-1.elasticbeanstalk.com

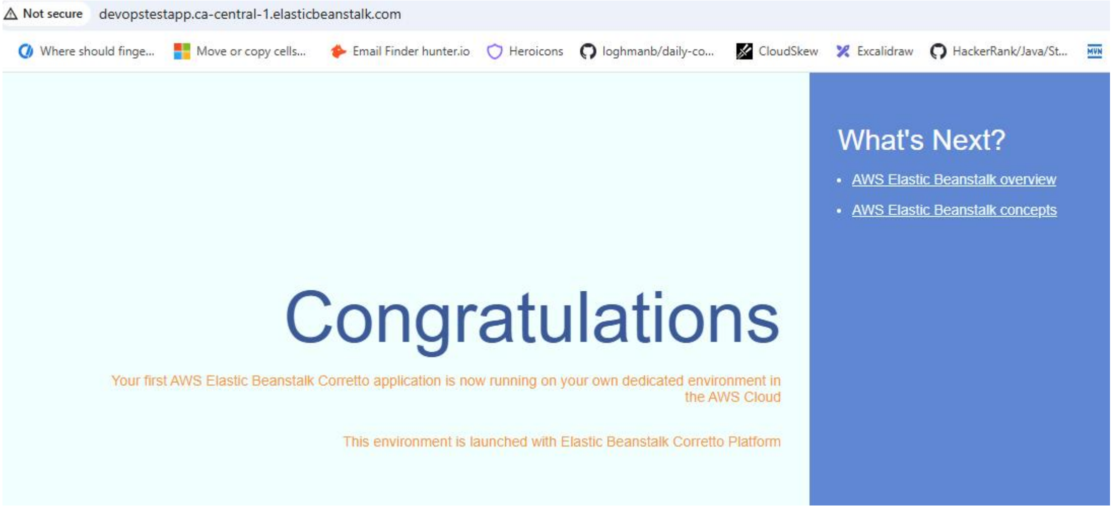
Environment ID
e-ja3biykgmk

Application name
MyApp1

Events (11)

Time	Type	Details
March 22, 2025 12:34:37 (UTC-4)	INFO	Successfully launched environment: MyApp1-env
March 22, 2025 12:34:36 (UTC-4)	INFO	Application available at devopstestapp.ca-central-1.elasticbeanstalk

If you click on the link



What's Next?

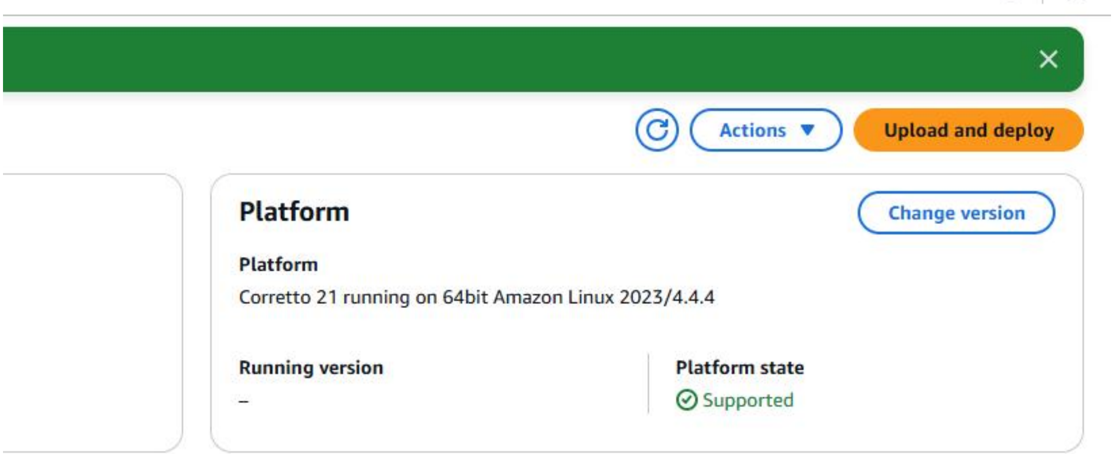
- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts

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

This is the Sample application, we can deploy our own application also ---> Click on Upload and deploy

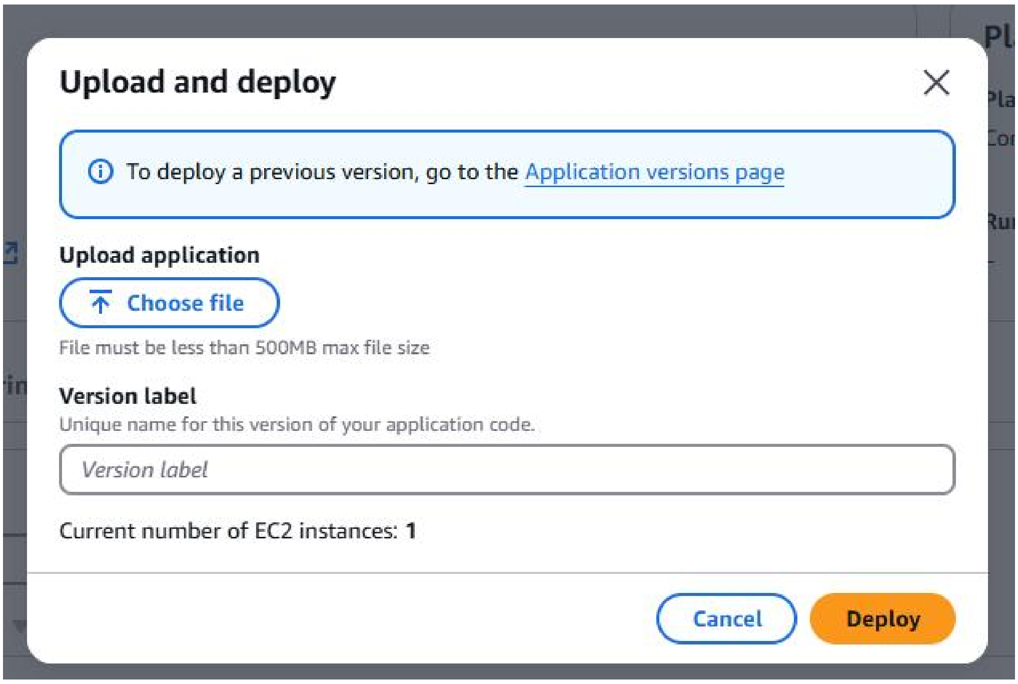


Platform

Platform
Corretto 21 running on 64bit Amazon Linux 2023/4.4.4

Running version
–

Platform state
✓ Supported



Upload and deploy

ⓘ To deploy a previous version, go to the Application versions page

Upload application

File must be less than 500MB max file size

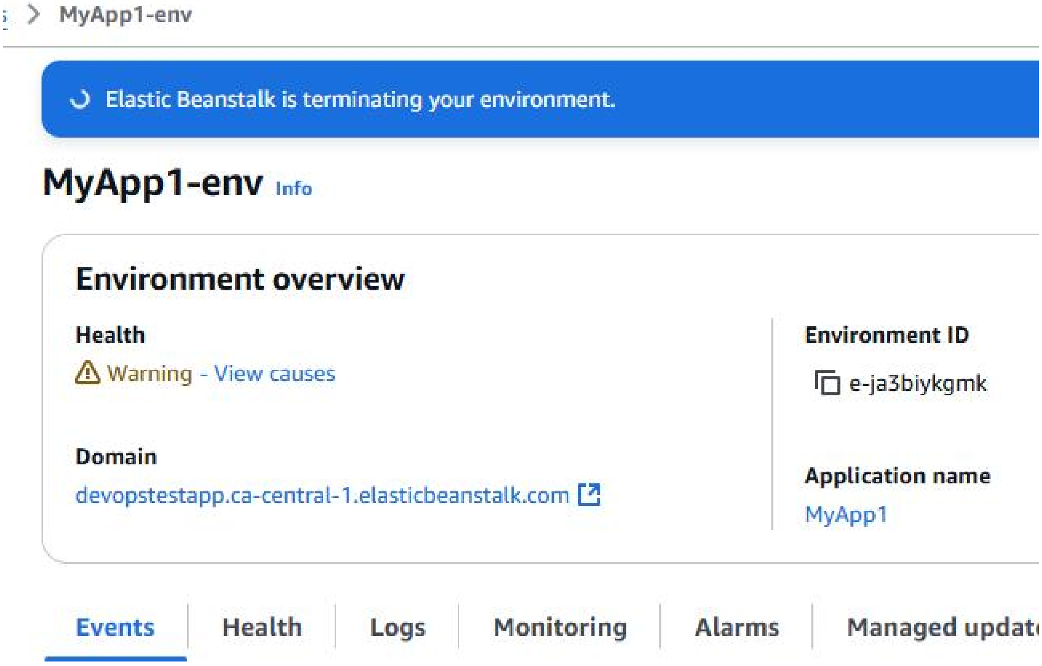
Version label
Unique name for this version of your application code.

Version label

Current number of EC2 instances: **1**

You can choose JAR or WAR file and upload to deploy your own application

Then Terminate the environment



Elastic Beanstalk is terminating your environment.

MyApp1-env Info

Environment overview

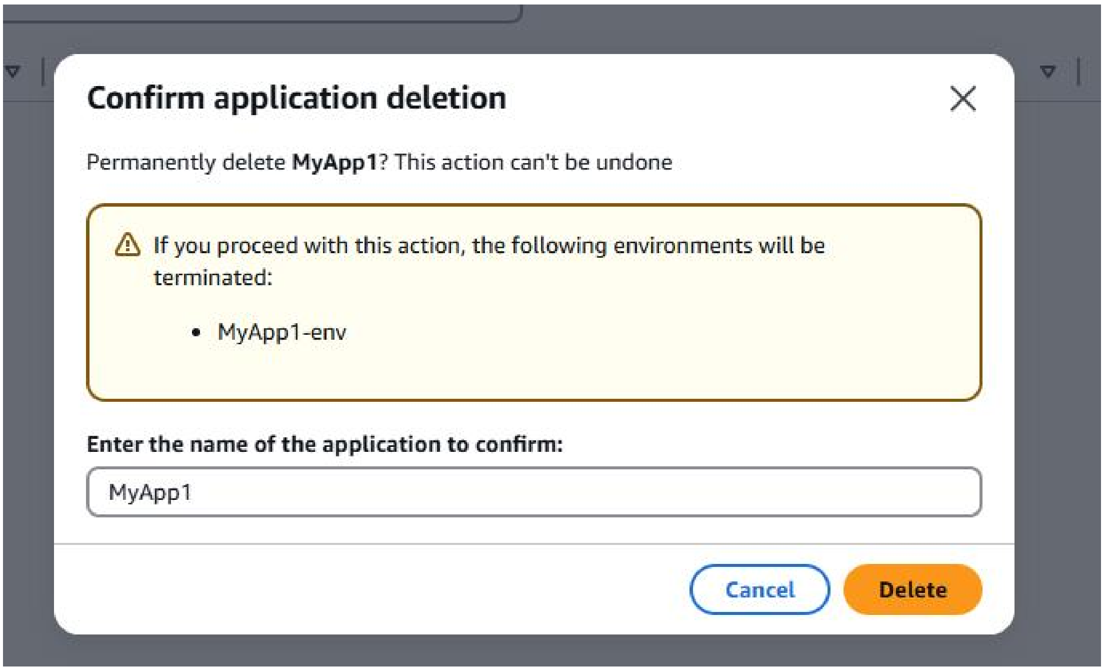
Health
⚠ Warning - View causes

Domain
devopstestapp.ca-central-1.elasticbeanstalk.com

Environment ID
e-ja3biykgmk

Application name
MyApp1

Go back to Application



Confirm application deletion

Permanently delete **MyApp1**? This action can't be undone

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

- MyApp1-env

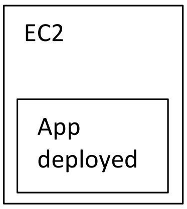
Enter the name of the application to confirm:

MyApp1

Elastic Beanstalk
PaaS ---> inside platform it will manage other resources and it is Pay-as-you-Go model

Inside platform, it will manage
1. EC2 instances
2. S3 buckets
3. LBR
4. Elastic IPs

EC2 VM ---> Hourly billing



EC2

App deployed

There is another concept ---> Pay as you use (you have deployed the application, but noone is making a request) . say if application is not getting executed, it will not be charged
If code is executed only then bill should be generated, till then bill will not be generated
If code I executed for 10 min, then bill should be generated for only 10 minutes

If you want this then we need ====> **Serverless computing comes into picture**

AWS Lambdas