SVKM'S NMIM'S Nilkamal School of Mathematics, Applied Statistics & Analytics
Master of Science (Data Science)

Practical-2 Platform as a service using AWS.

Date:-23/01/2024 **Submission Date:- 30/01/2024**

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.

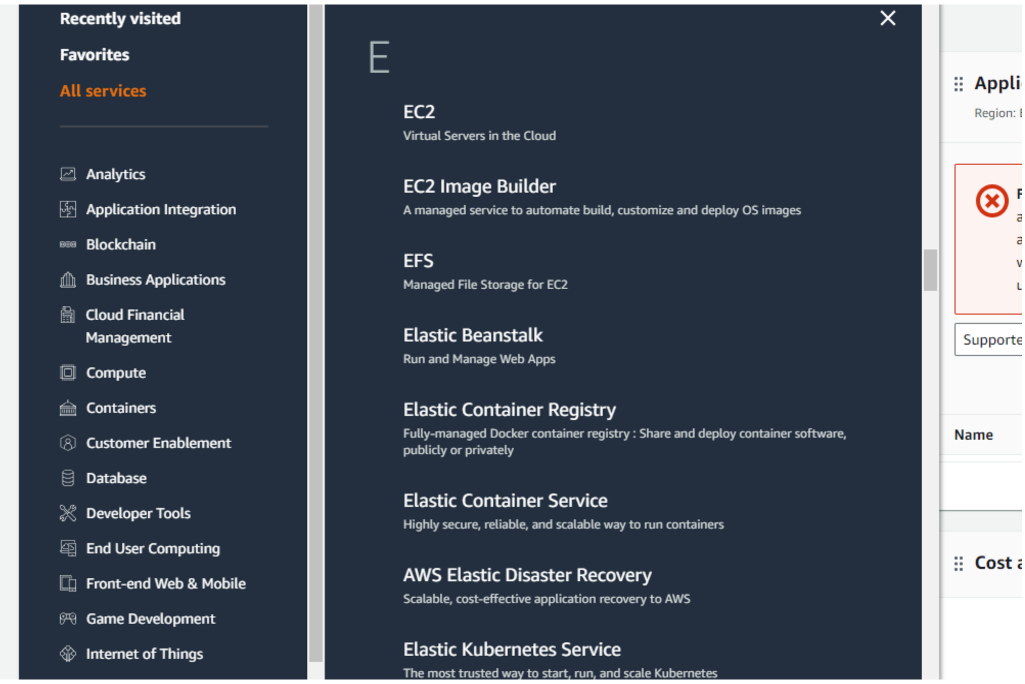
- **IAM**

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.

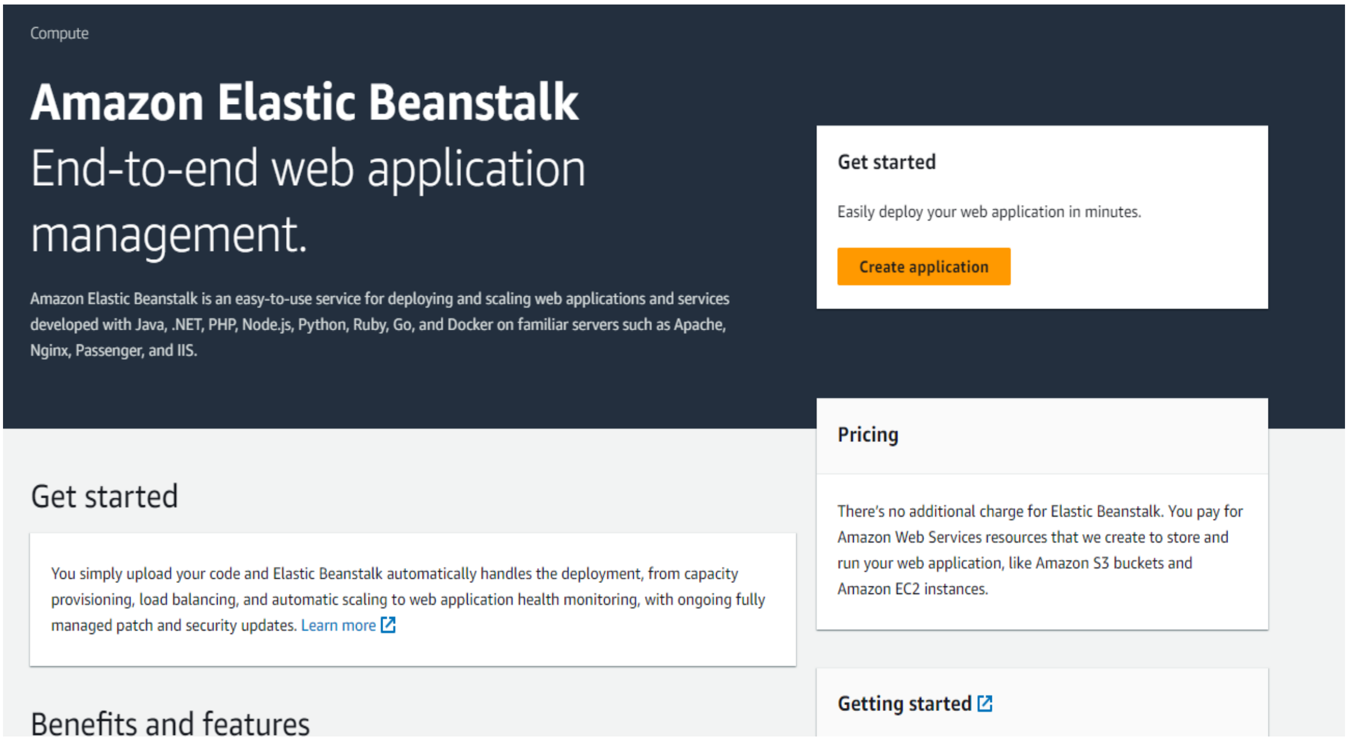
- Implement PAAS using elastic beanstalk for the following.
  1. Server
  2. Java
  3. Python
  4. Node.js

For Python

Step 1:- Select Elastic BeanStalk


Recently visited
Favorites
All services
Analytics
Application Integration
Blockchain
Business Applications
Cloud Financial Management
Compute
Containers
Customer Enablement
Database
Developer Tools
End User Computing
Front-end Web & Mobile
Game Development
Internet of Things
E
EC2
Virtual Servers in the Cloud
EC2 Image Builder
A managed service to automate build, customize and deploy OS images
EFS
Managed File Storage for EC2
Elastic Beanstalk
Run and Manage Web Apps
Elastic Container Registry
Fully-managed Docker container registry : Share and deploy container software, publicly or privately
Elastic Container Service
Highly secure, reliable, and scalable way to run containers
AWS Elastic Disaster Recovery
Scalable, cost-effective application recovery to AWS
Elastic Kubernetes Service
The most trusted way to start, run, and scale Kubernetes
Appli
Region:
Supporte
Name
Cost a

Step 2- Create a new 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.

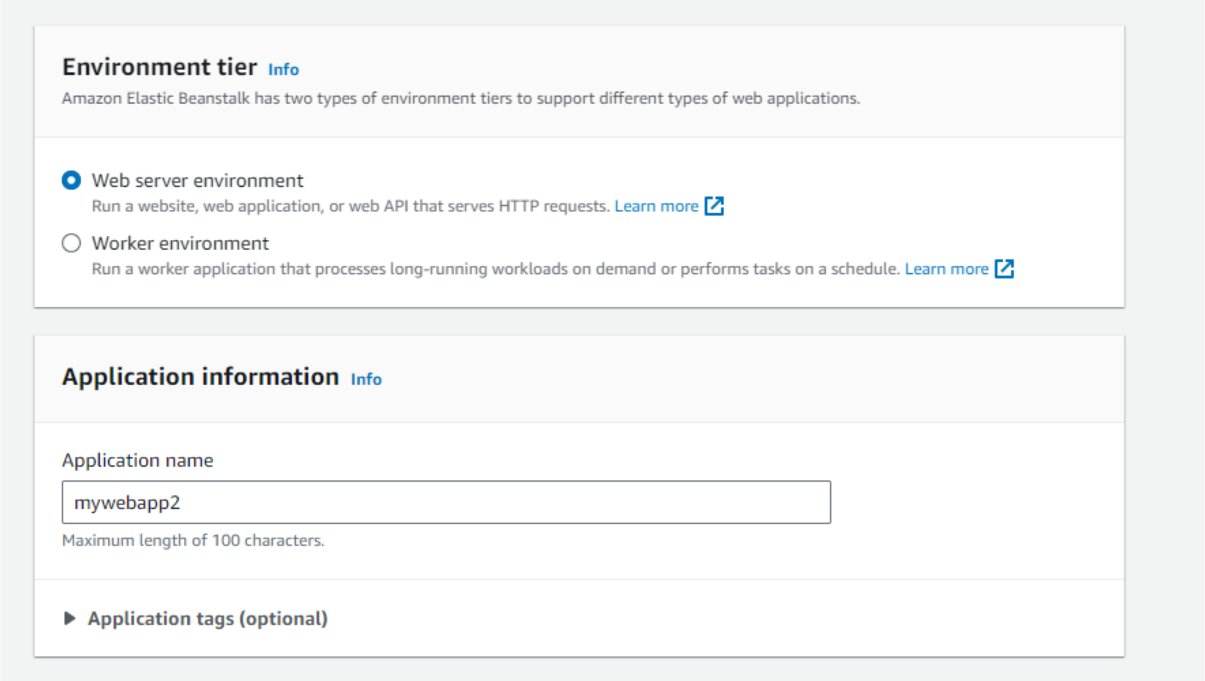
Get started

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

Getting started

Benefits and features

Step 3- Provide a new Name for the Application



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

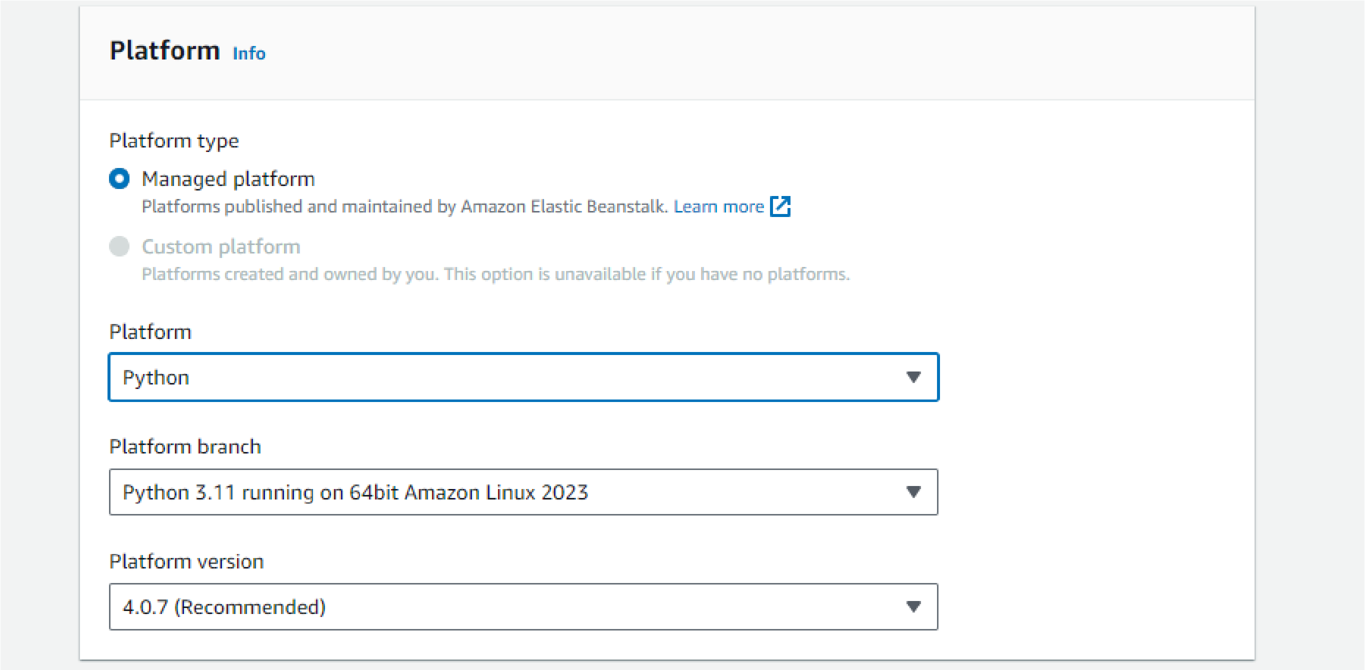
Application information Info

Application name: mywebapp2

Maximum length of 100 characters.

▶ Application tags (optional)

Step 4- Choose the Platform as Python



Platform

Platform type

● Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

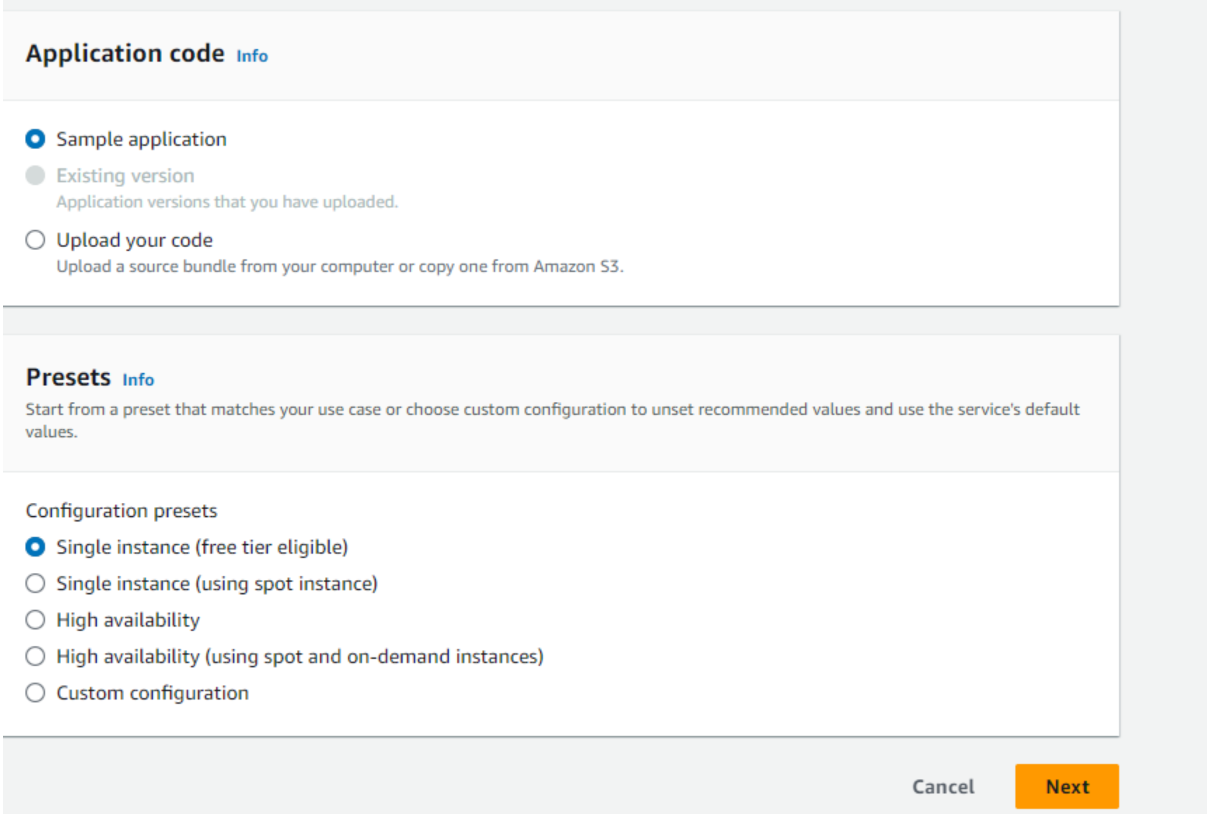
○ Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform: Python

Platform branch: Python 3.11 running on 64bit Amazon Linux 2023

Platform version: 4.0.7 (Recommended)

Step 5- Keep it as Single Instance



Application code

● 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.

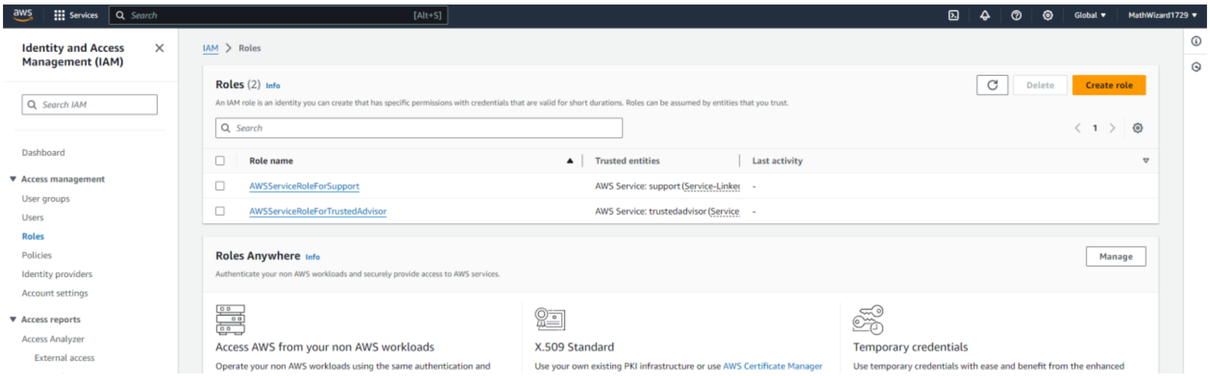
Presets

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

Step 6- Create a new Role under IAM(Identity Access Management)



Roles (2) 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.

	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

Roles Anywhere Info

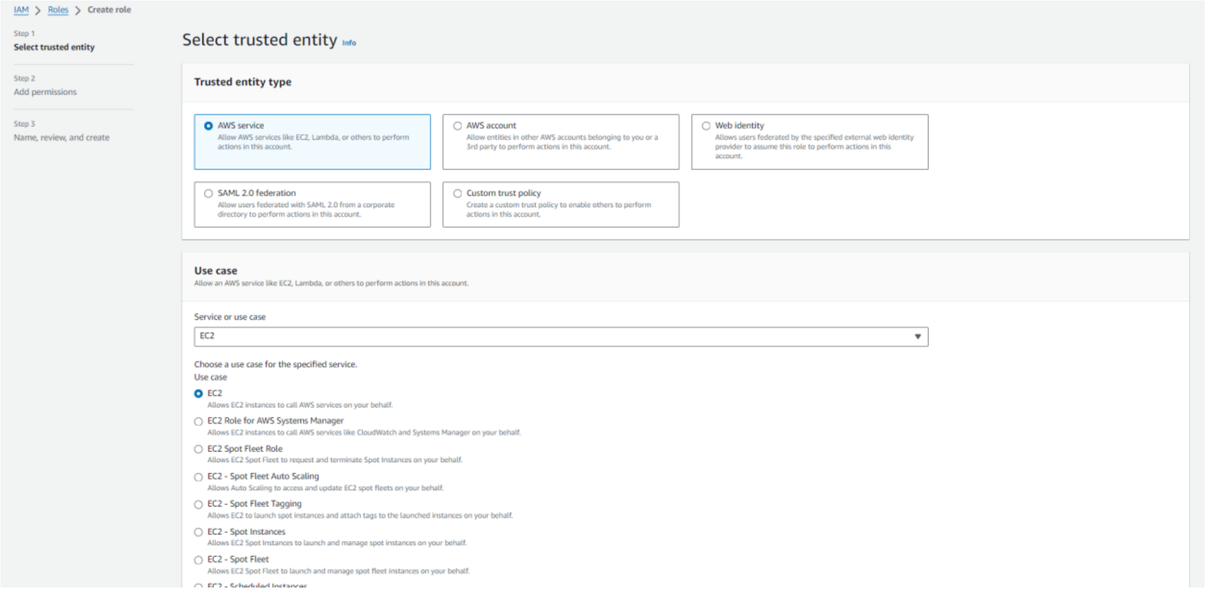
Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads
Operate your non AWS workloads using the same authentication and

X.509 Standard
Use your own existing PKI infrastructure or use AWS Certificate Manager

Temporary credentials
Use temporary credentials with ease and benefit from the enhanced

Step 7- Create a new role and select the Usecase as EC2



Select trusted entity Info

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.

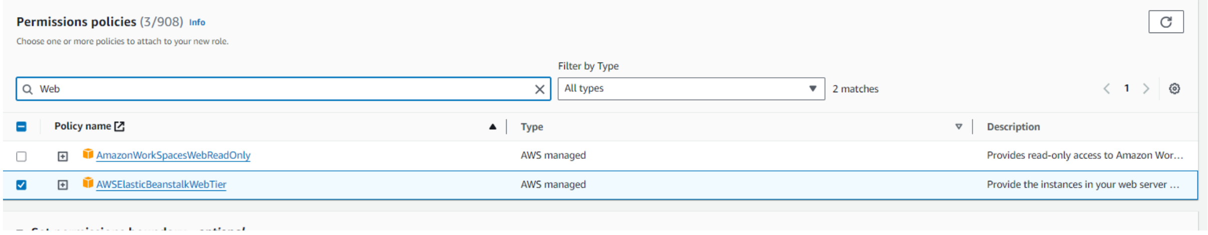
○ EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

○ 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

Step 8- Select Web Tier, Worker Tier and MultiContainer docker tier

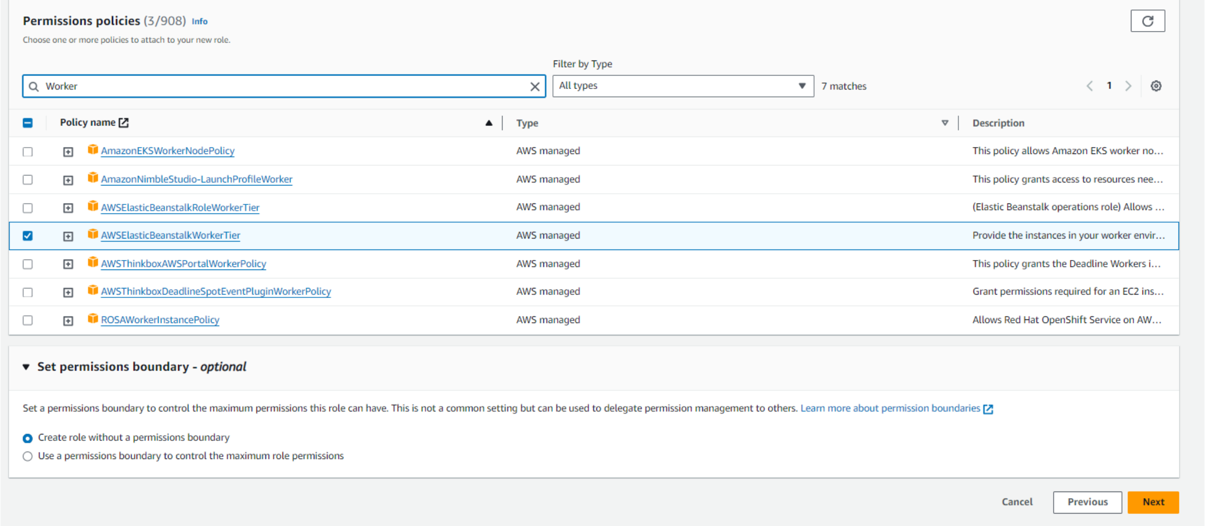


Permissions policies (3/908) Info

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

Search: Web
Filter by Type: All types
2 matches

	Policy name	Type	Description
□	AmazonWorkSpacesWebReadOnly	AWS managed	Provides read-only access to Amazon Wor…
☑	AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server …



Permissions policies (3/908) Info

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

Filter by Type

Worker | All types | 7 matches

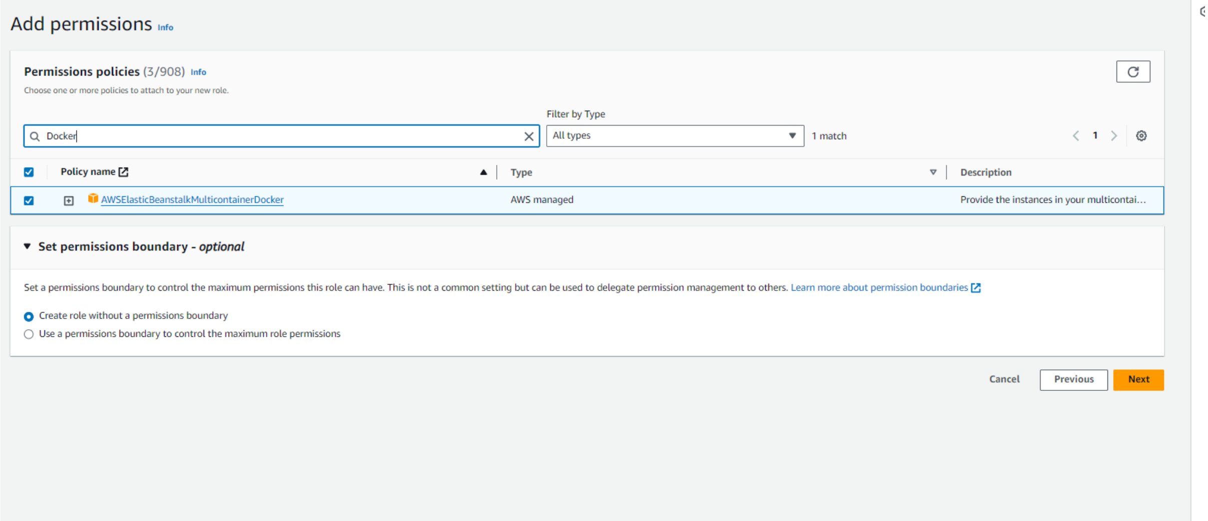
	Policy name	Type	Description
□	AmazonEKSWorkerNodePolicy	AWS managed	This policy allows Amazon EKS worker no...
□	AmazonNimbleStudio-LaunchProfileWorker	AWS managed	This policy grants access to resources nee...
□	AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allows ...
☑	AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker envir...
□	AWSThinkboxAWSPortalWorkerPolicy	AWS managed	This policy grants the Deadline Workers i...
□	AWSThinkboxDeadlineSpotEventPluginWorkerPolicy	AWS managed	Grant permissions required for an EC2 ins...
□	ROSAWorkerInstancePolicy	AWS managed	Allows Red Hat OpenShift Service on AW...

Set permissions boundary - *optional*

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others. Learn more about permission boundaries

● Create role without a permissions boundary

○ Use a permissions boundary to control the maximum role permissions



Add permissions Info

Permissions policies (3/908) Info

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

Filter by Type

Docker | All types | 1 match

	Policy name	Type	Description
☑	AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicontai...

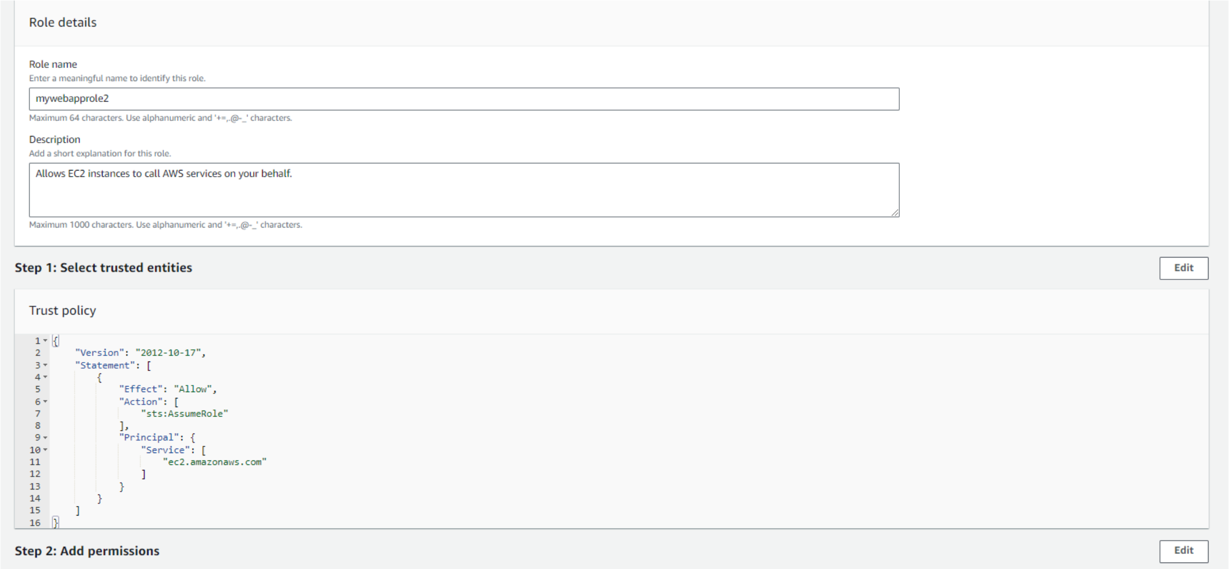
Set permissions boundary - *optional*

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others. Learn more about permission boundaries

● Create role without a permissions boundary

○ Use a permissions boundary to control the maximum role permissions

Step 9- Provide a Name for Webapp Role



Role details

Role name

Enter a meaningful name to identify this role.

mywebapprole2

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.

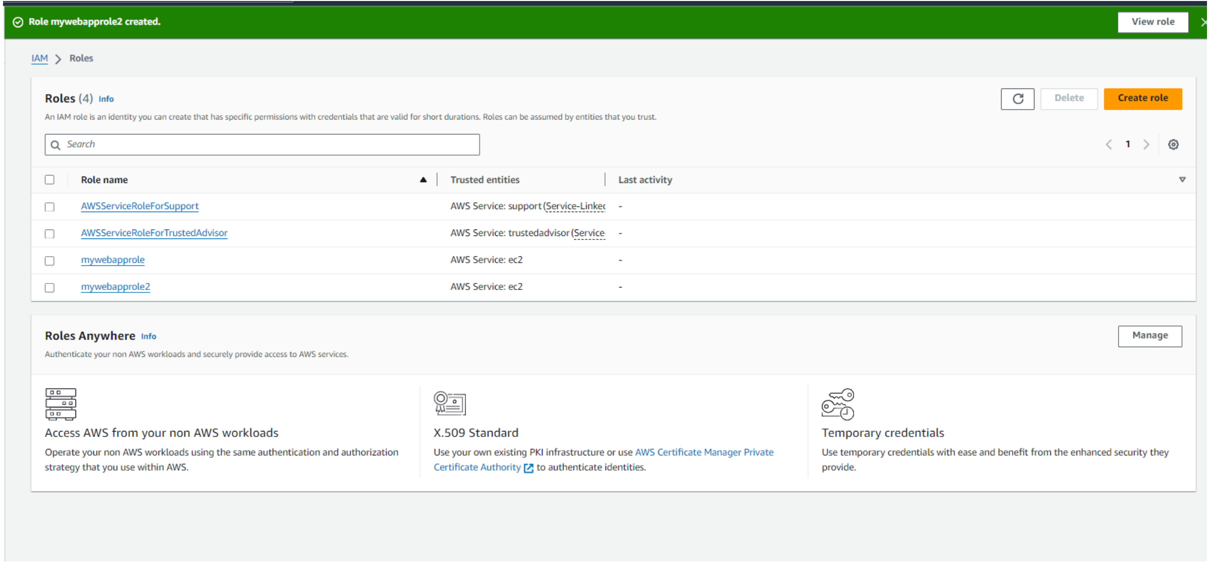
Step 1: Select trusted entities

Trust policy

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

Step 2: Add permissions

Step 10- role was Created Successfully



Role mywebapprole2 created.

Roles (4) 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.

□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linkec	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	mywebapprole	AWS Service: ec2	-
□	mywebapprole2	AWS Service: ec2	-

Roles Anywhere Info

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.

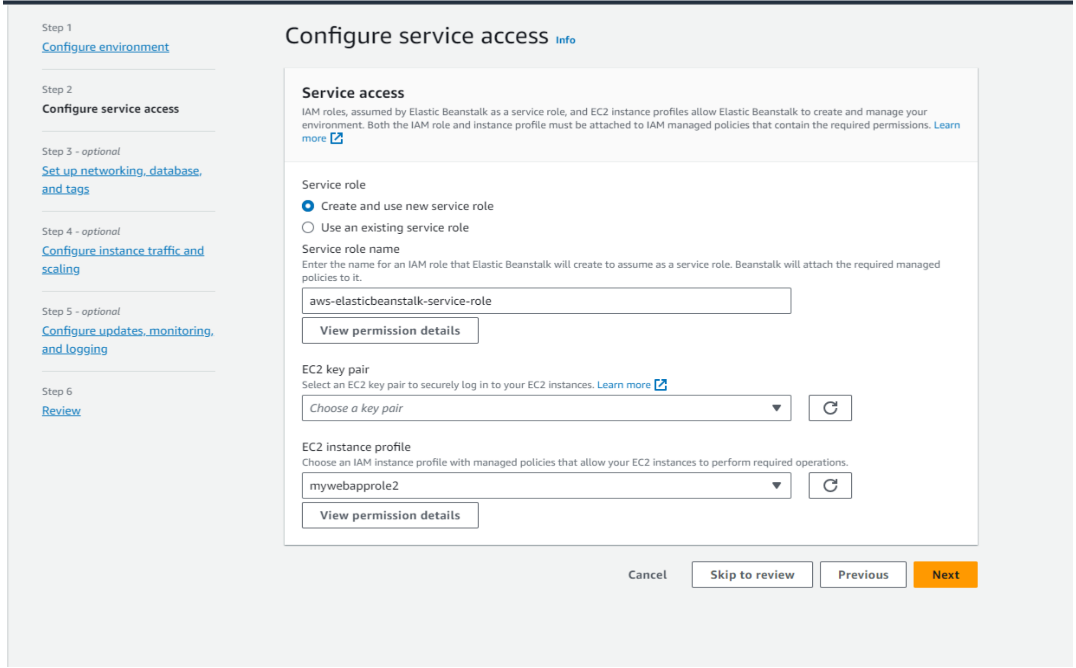
X.509 Standard

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.

Step 11- For this Access Create a new use case and service role and for Instance profile select as mywebapprole2



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

EC2 key pair

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

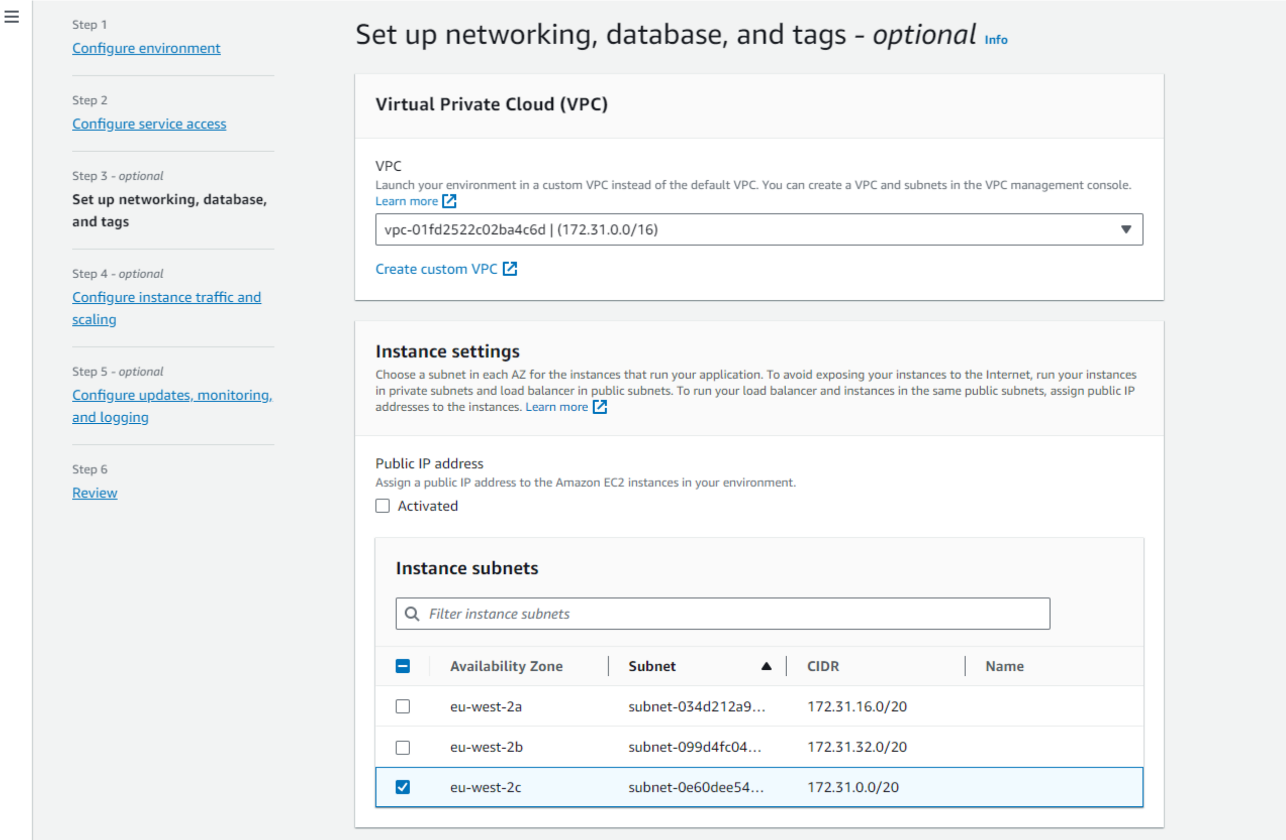
Choose a key pair

EC2 instance profile

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

mywebapprole2

Step 12- Select the VP name as given and from the IP address provided select the same down below for Instance Settings and Database



Set up networking, database, and tags - *optional* Info

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-01fd2522c02ba4c6d | (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

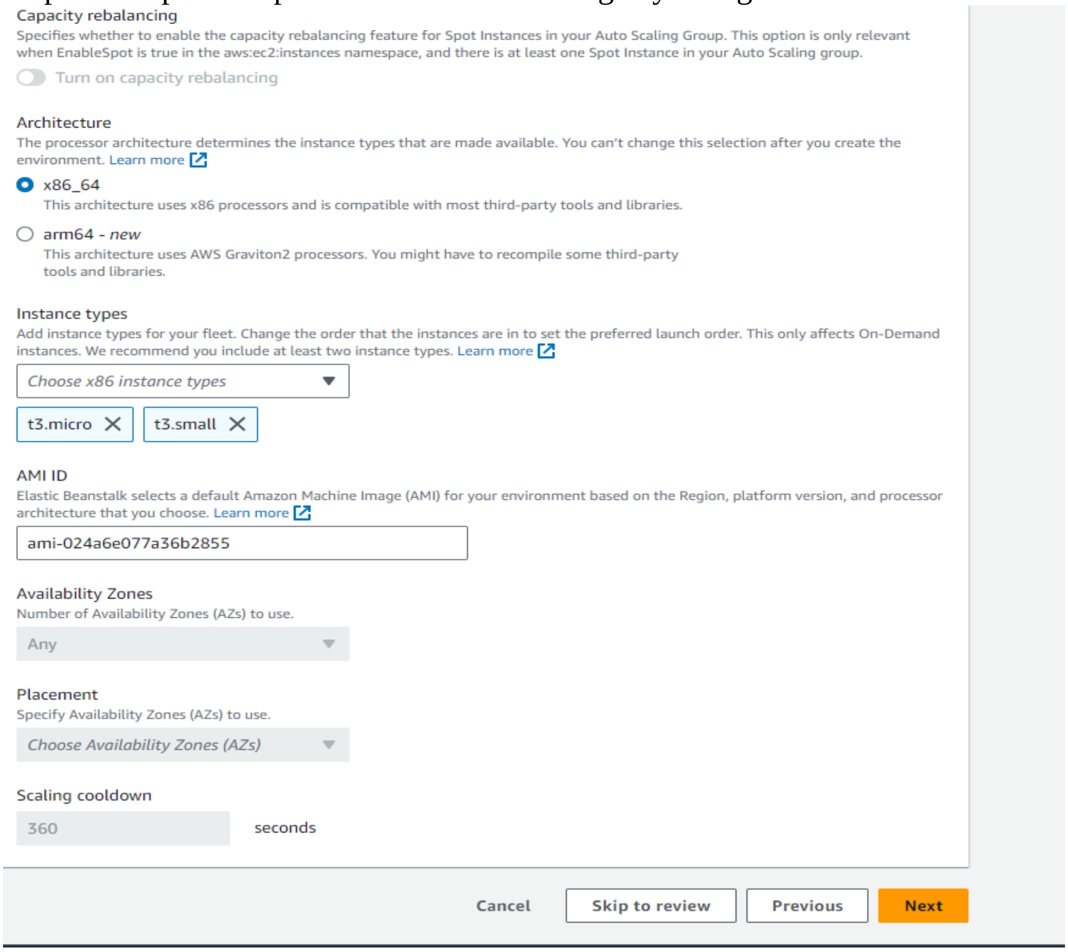
Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
☐ Activated

Instance subnets

Filter instance subnets

☐	Availability Zone	Subnet	CIDR	Name
☐	eu-west-2a	subnet-034d212a9...	172.31.16.0/20	
☐	eu-west-2b	subnet-099d4fc04...	172.31.32.0/20	
☑	eu-west-2c	subnet-0e60dee54...	172.31.0.0/20	

Step 13- Skip the Step 4 as it is without making any changes



Capacity rebalancing
Specifies whether to enable the capacity rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when EnableSpot is true in the aws:ec2:instances namespace, and there is at least one Spot Instance in your Auto Scaling group.
Turn on capacity rebalancing

Architecture
The processor architecture determines the instance types that are made available. You can't change this selection after you create the environment. Learn more

● 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

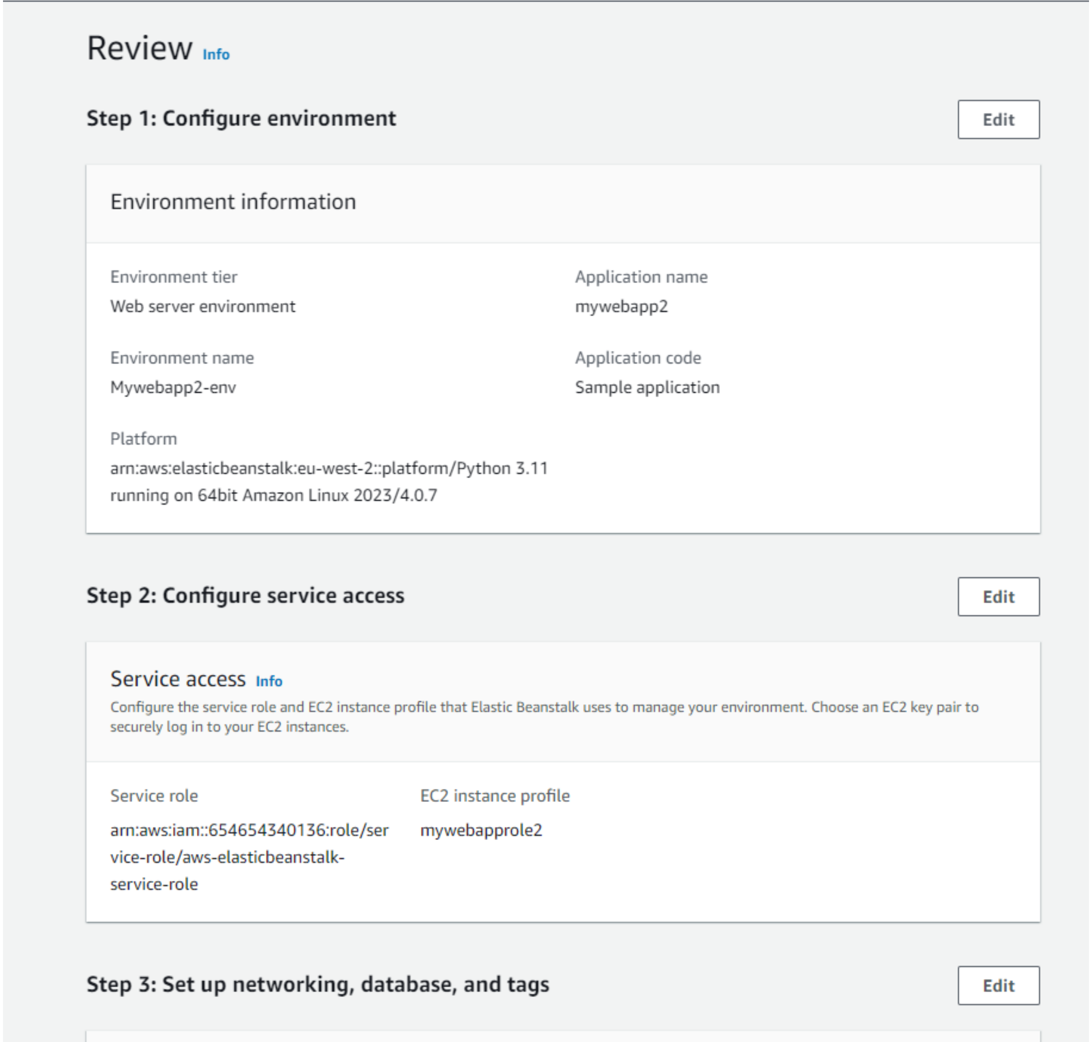
ami-024a6e077a36b2855

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

Step 14- Skip the Step 5 Section and directly go to Review Page where you can view the complete Changes made.and Submit it



Review

Step 1: Configure environment

Environment information

Environment tier
Web server environment

Application name
mywebapp2

Environment name
Mywebapp2-env

Application code
Sample application

Platform
arn:aws:elasticbeanstalk:eu-west-2::platform/Python 3.11 running on 64bit Amazon Linux 2023/4.0.7

Step 2: Configure service access

Service access

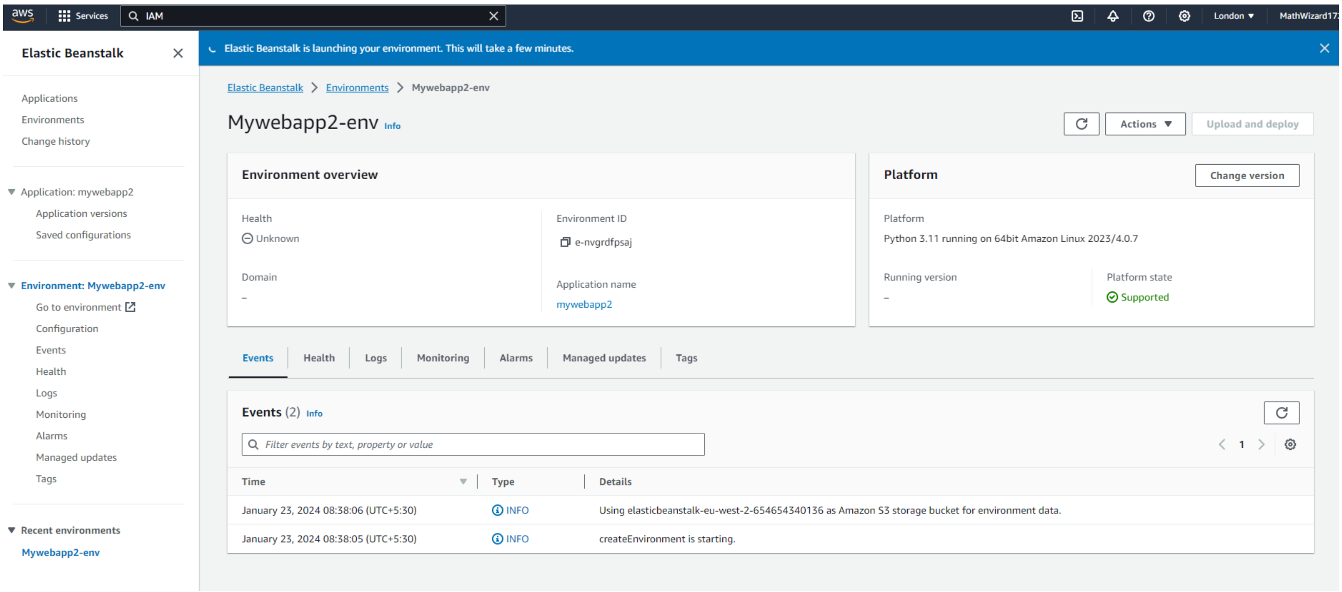
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role
arn:aws:iam::654654340136:role/service-role/aws-elasticbeanstalk-service-role

EC2 instance profile
mywebapprole2

Step 3: Set up networking, database, and tags

Step 15- After the review the Elastic BeanStalk will run environment and it will take some few minutes



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

Mywebapp2-env

Environment overview

Health
Unknown

Environment ID
e-nvgrdfpsaj

Domain
–

Application name
mywebapp2

Platform

Platform
Python 3.11 running on 64bit Amazon Linux 2023/4.0.7

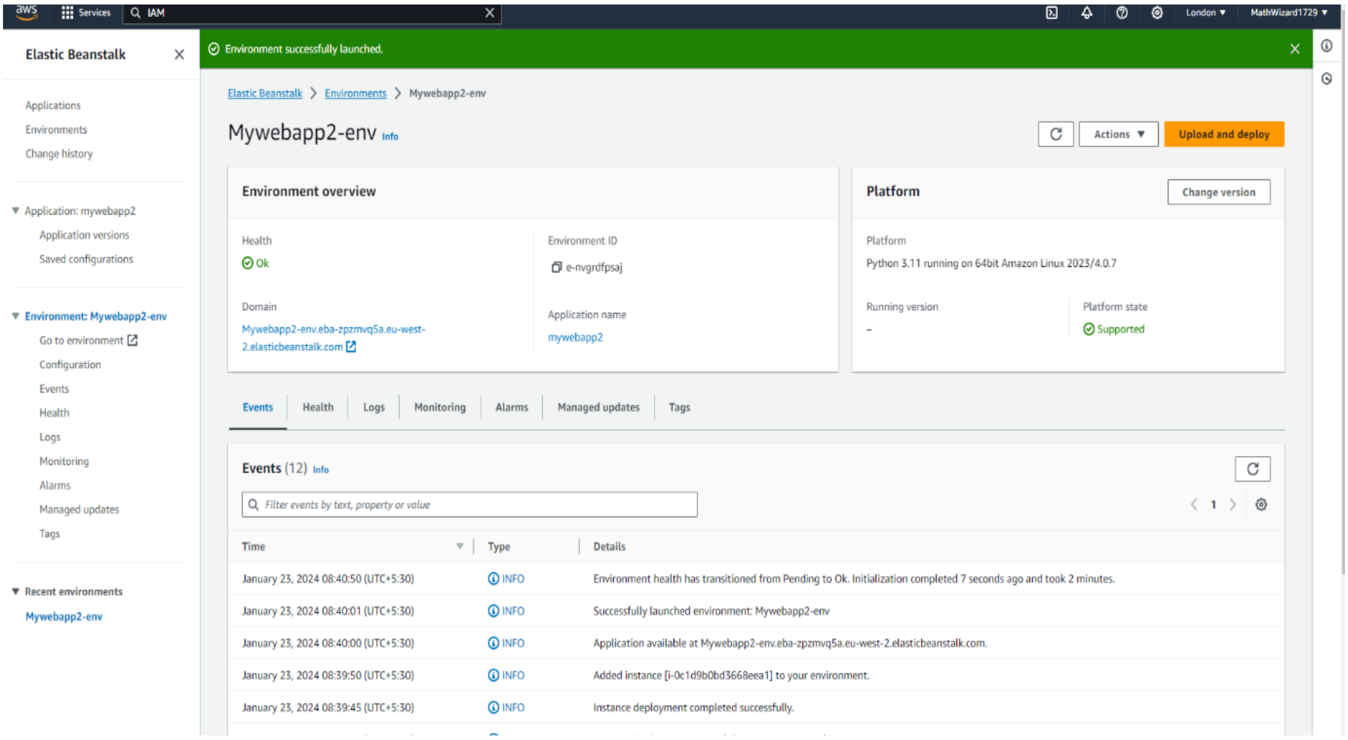
Running version
–

Platform state
Supported

Events (2)

Time	Type	Details
January 23, 2024 08:38:06 (UTC+5:30)	INFO	Using elasticbeanstalk-eu-west-2-654654340136 as Amazon S3 storage bucket for environment data.
January 23, 2024 08:38:05 (UTC+5:30)	INFO	createEnvironment is starting.

Step 16- Environment is successfully launched



Environment successfully launched.

Mywebapp2-env

Environment overview

Health: Ok

Environment ID: e-nvgrdfpsaj

Domain: Mywebapp2-env.eba-zpzmvq5a.eu-west-2.elasticbeanstalk.com

Application name: mywebapp2

Platform

Platform: Python 3.11 running on 64bit Amazon Linux 2023/4.0.7

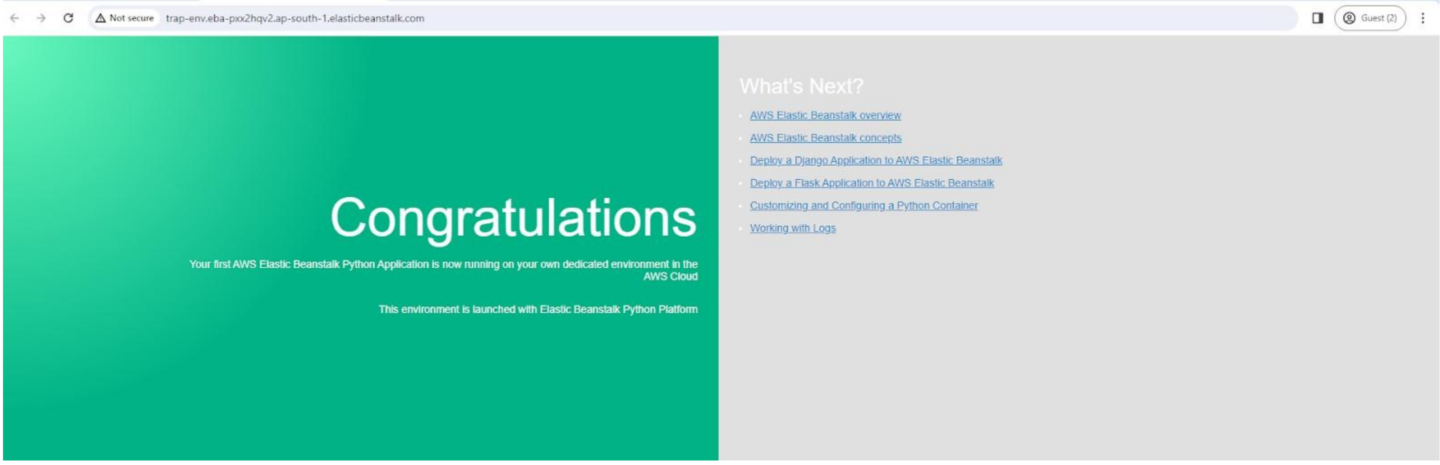
Running version: -

Platform state: Supported

Events (12)

Time	Type	Details
January 23, 2024 08:40:50 (UTC+5:30)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 7 seconds ago and took 2 minutes.
January 23, 2024 08:40:01 (UTC+5:30)	INFO	Successfully launched environment: Mywebapp2-env
January 23, 2024 08:40:00 (UTC+5:30)	INFO	Application available at Mywebapp2-env.eba-zpzmvq5a.eu-west-2.elasticbeanstalk.com.
January 23, 2024 08:39:50 (UTC+5:30)	INFO	Added instance [i-0c1d9b0bd3668eea1] to your environment.
January 23, 2024 08:39:45 (UTC+5:30)	INFO	Instance deployment completed successfully.

Step 17- The Output is shown as like this



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

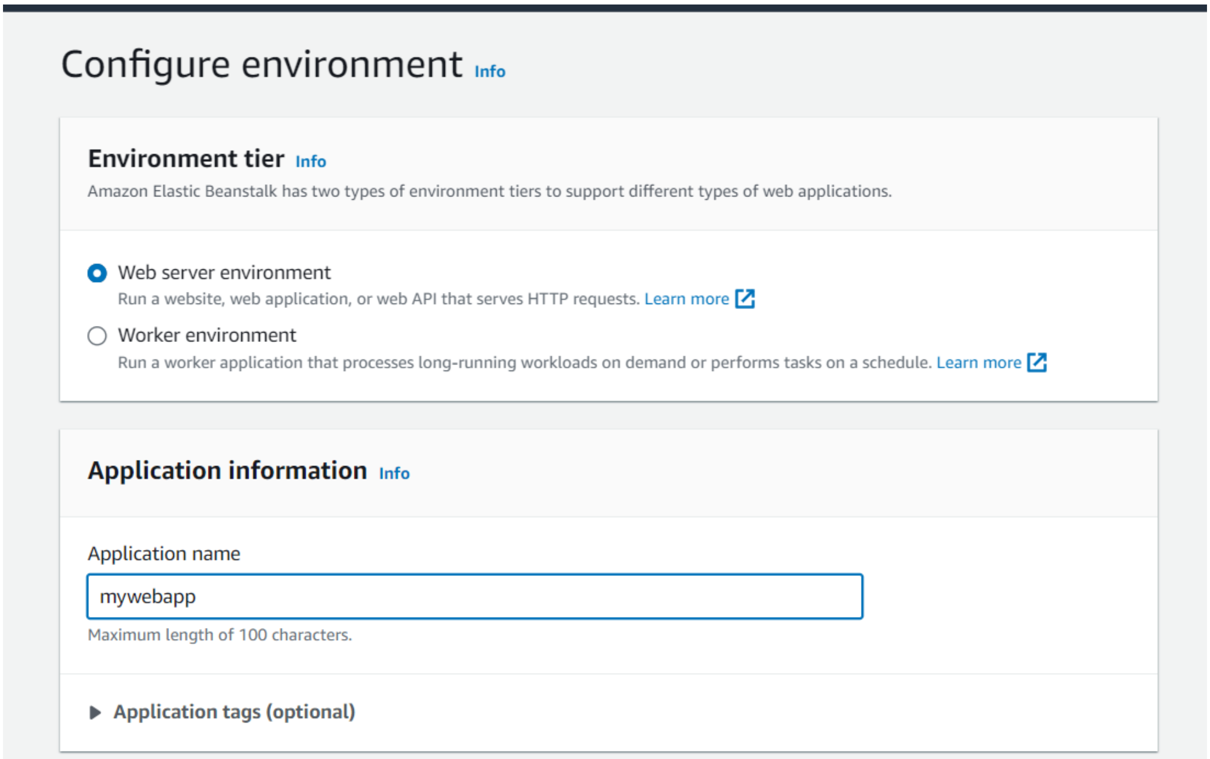
What's Next?

- 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

For Java

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

Step 3:- Provide the name for the Elastic Bean Stalk Application



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

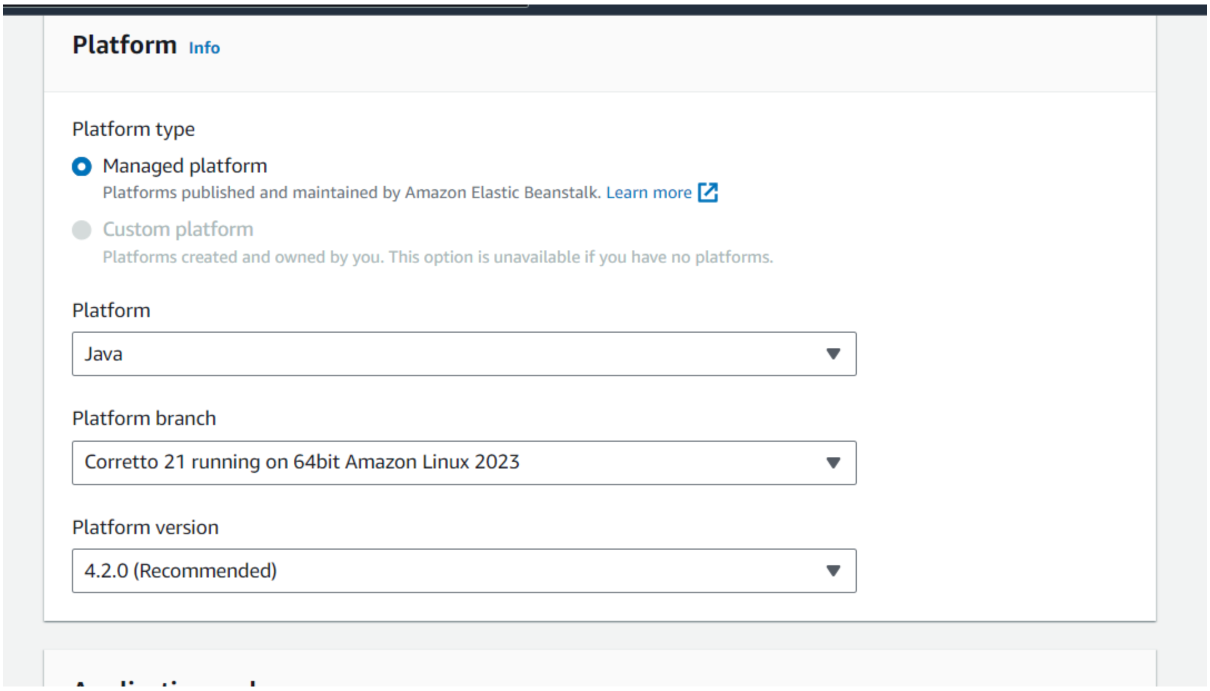
Application information Info

Application name: mywebapp

Maximum length of 100 characters.

▶ Application tags (optional)

Step 4- Select Platform as Java and go on further



Platform Info

Platform type

● Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

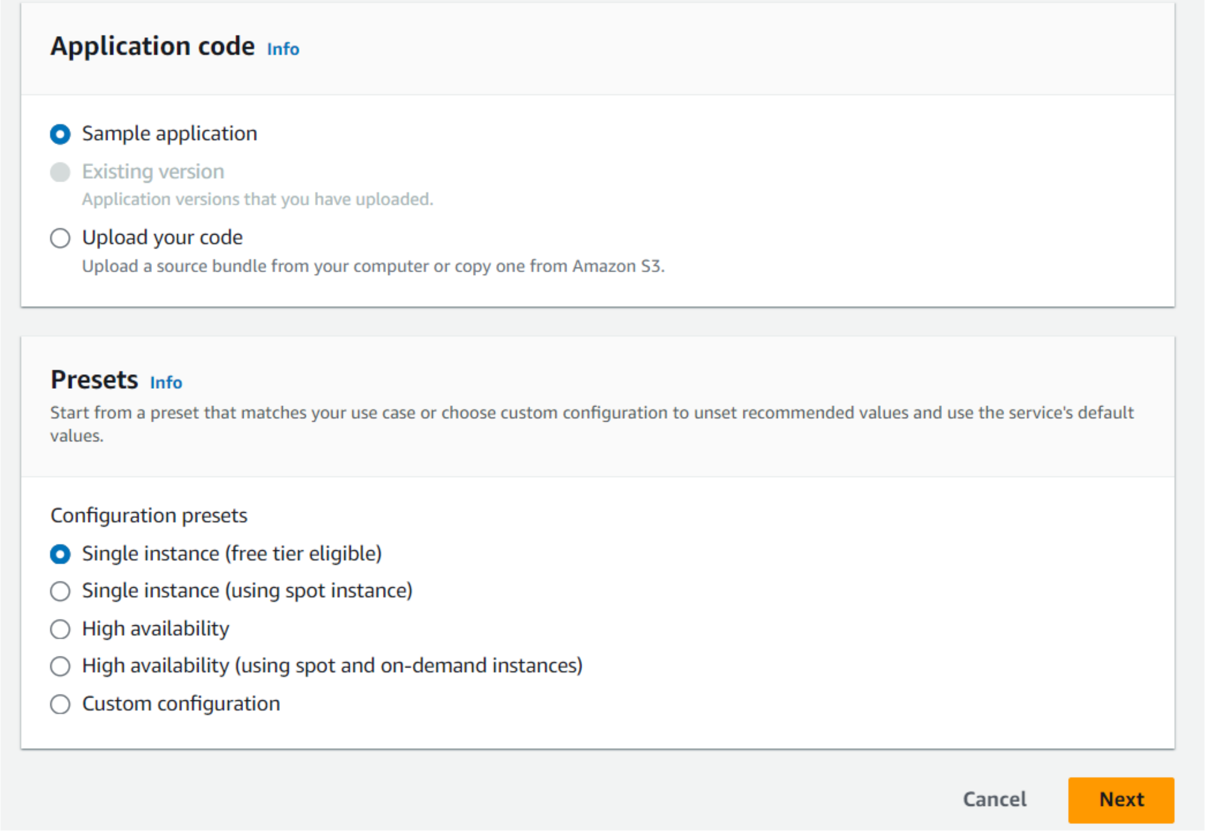
○ Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform: Java

Platform branch: Corretto 21 running on 64bit Amazon Linux 2023

Platform version: 4.2.0 (Recommended)

Step 5-Keep it as Single Instance.



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)

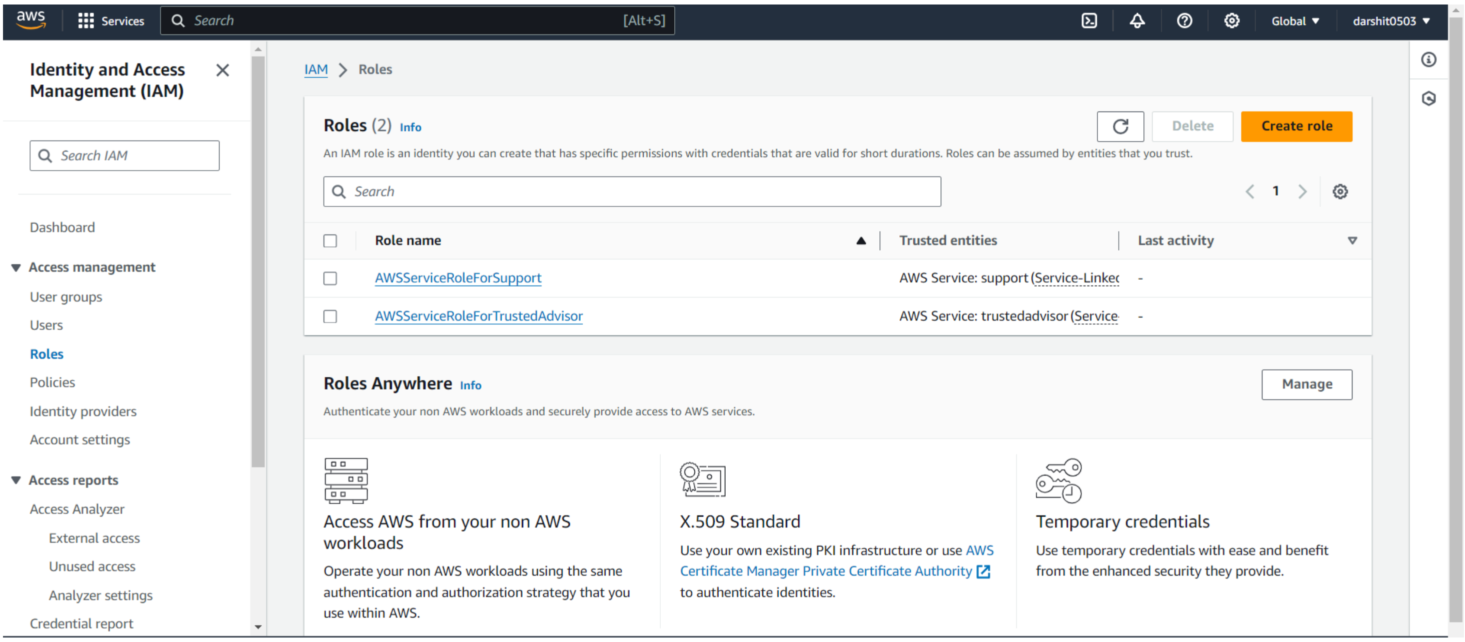
○ Single instance (using spot instance)

○ High availability

○ High availability (using spot and on-demand instances)

○ Custom configuration

Step 6- Create Role under IAM User



Roles (2) 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.

□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linkec	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-	-

Roles Anywhere Info

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

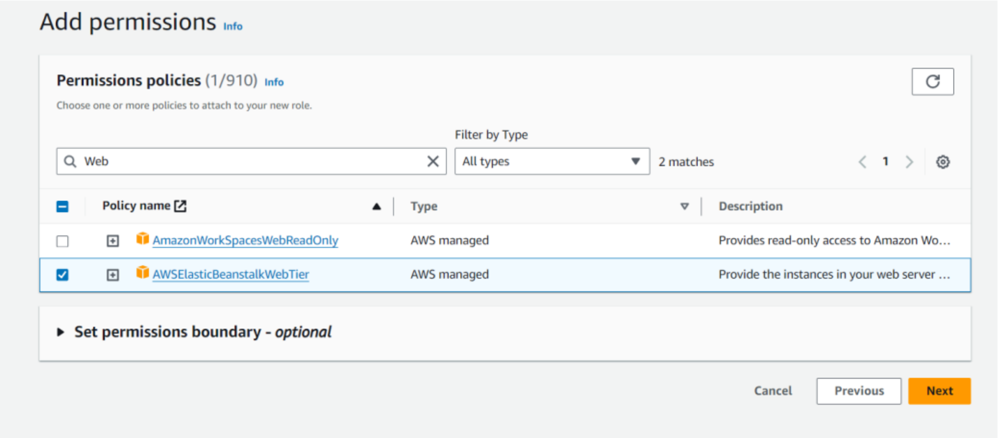
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

Step 7- Select Usecase as EC2 and select Next and from the following list Select3 options

a) WebTier



Add permissions Info

Permissions policies (1/910) Info

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

Filter by Type

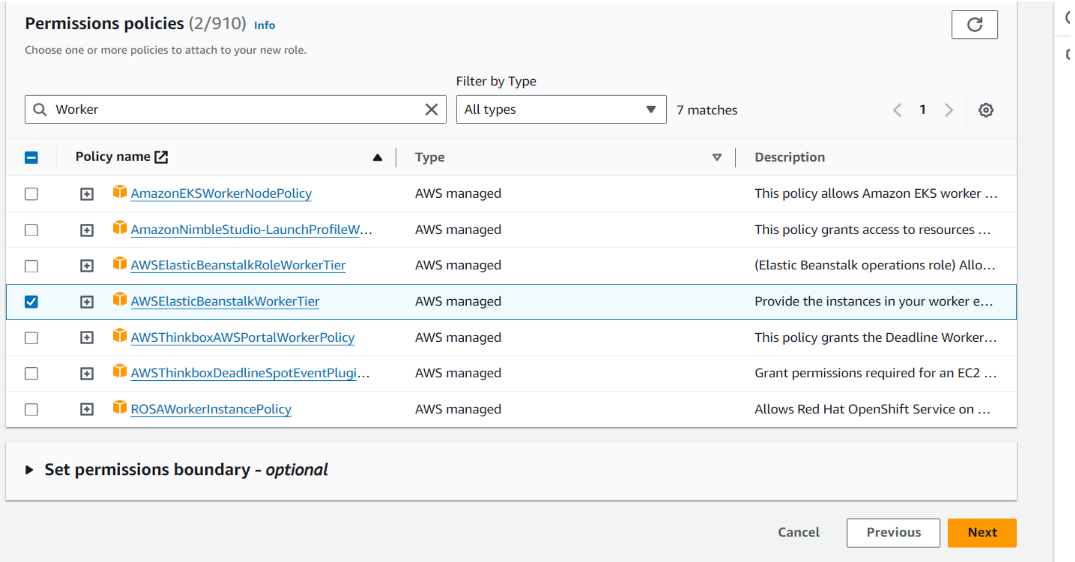
Web | All types | 2 matches

	Policy name	Type	Description
□	AmazonWorkSpacesWebReadOnly	AWS managed	Provides read-only access to Amazon Wo…
☑	AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server …

▸ Set permissions boundary - *optional*

Cancel | Previous | Next

b) WorkerTier



Permissions policies (2/910) Info

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

Filter by Type

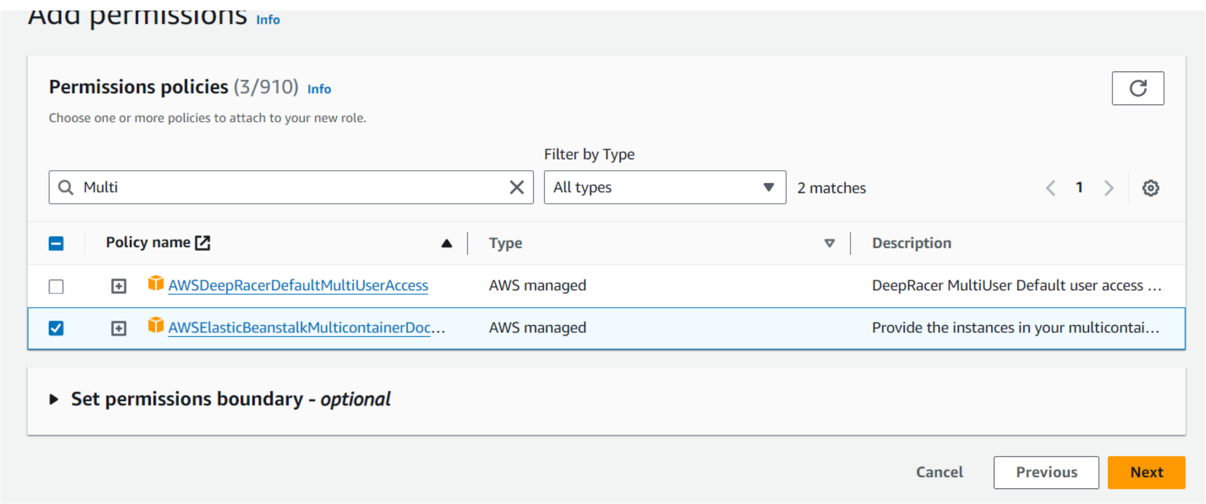
Worker | All types | 7 matches

	Policy name	Type	Description
□	AmazonEKSWorkerNodePolicy	AWS managed	This policy allows Amazon EKS worker …
□	AmazonNimbleStudio-LaunchProfileW…	AWS managed	This policy grants access to resources …
□	AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allo…
☑	AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker e…
□	AWSThinkboxAWSPortalWorkerPolicy	AWS managed	This policy grants the Deadline Worker…
□	AWSThinkboxDeadlineSpotEventPlugi…	AWS managed	Grant permissions required for an EC2 …
□	ROSAWorkerInstancePolicy	AWS managed	Allows Red Hat OpenShift Service on …

▸ Set permissions boundary - *optional*

Cancel | Previous | Next

c) MultiContainerDocker list



Add permissions Info

Permissions policies (3/910) Info

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

Filter by Type

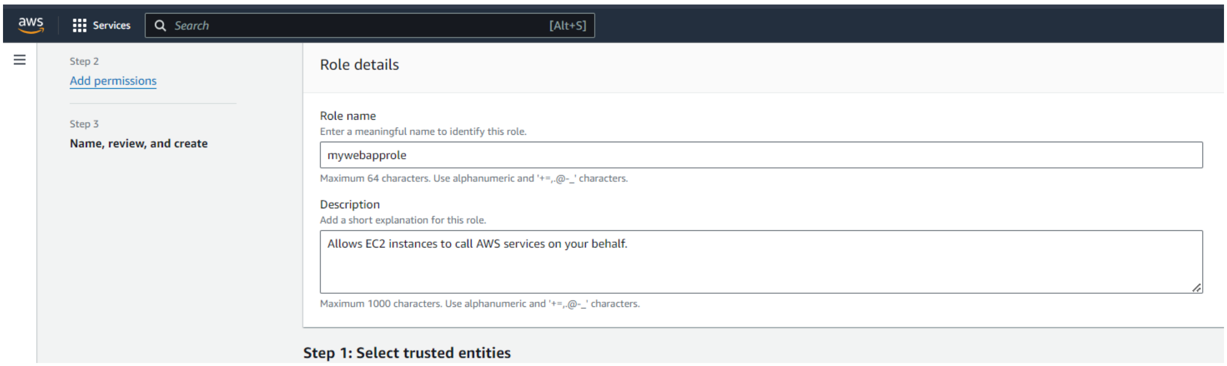
Multi | All types | 2 matches

	Policy name	Type	Description
□	AWSDeepRacerDefaultMultiUserAccess	AWS managed	DeepRacer MultiUser Default user access …
☑	AWSElasticBeanstalkMulticontainerDoc…	AWS managed	Provide the instances in your multicontai…

▸ Set permissions boundary - *optional*

Cancel | Previous | Next

Step 8 - Provide the name for the Role



Role details

Role name

Enter a meaningful name to identify this role.

mywebapprole

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

Description

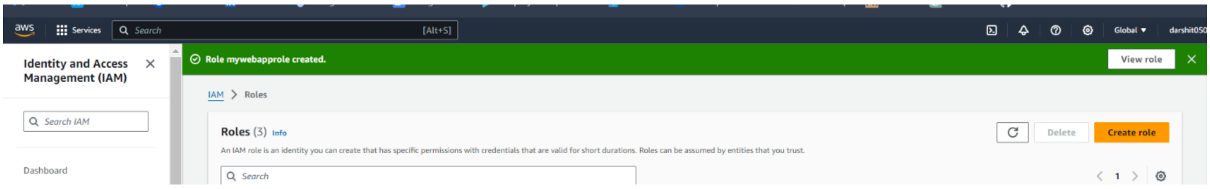
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

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

Step 1: Select trusted entities

Step 9- The Role was created Successfully

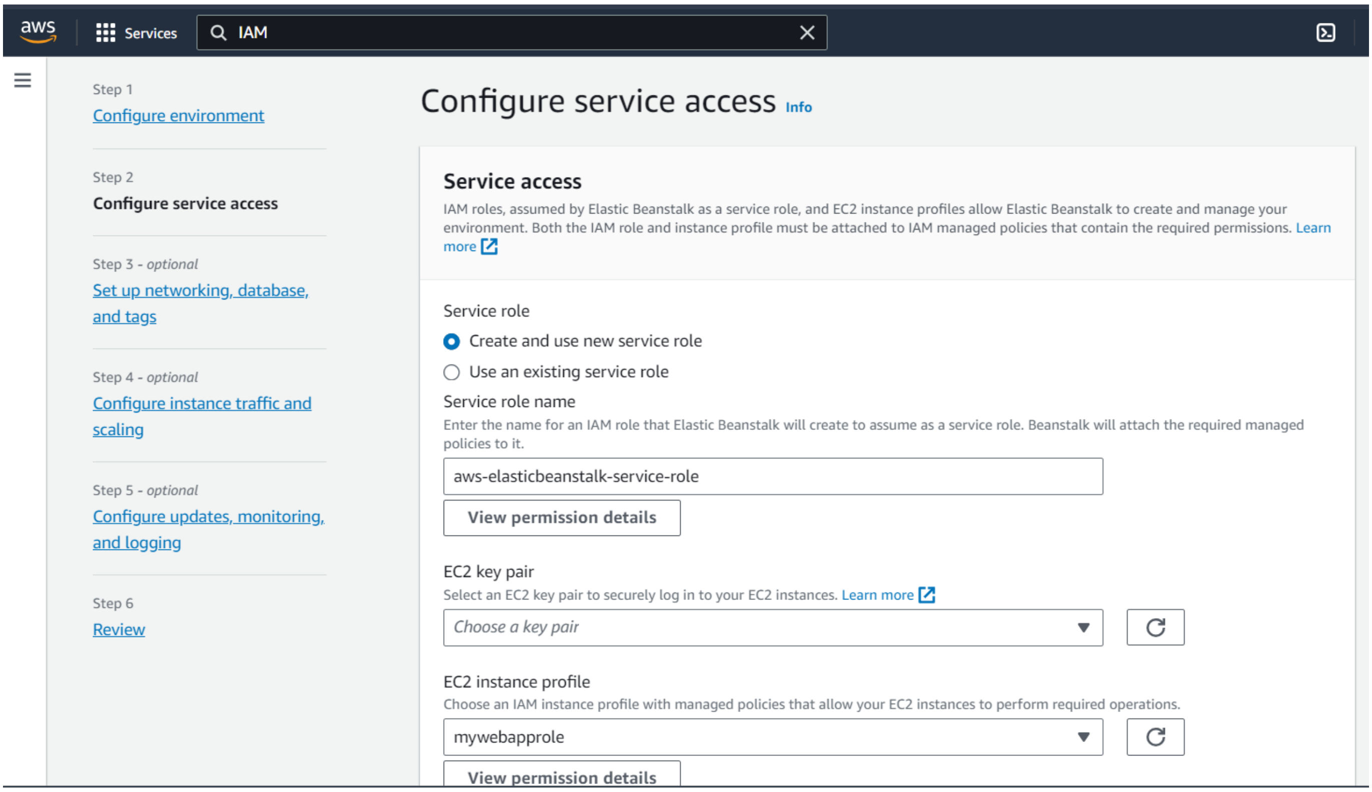


Role mywebapprole created.

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.

Step 10- For the Configure Service Access select new service Role and under domain EC2 Instance Role select mywebapprole which was created above



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

EC2 key pair

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

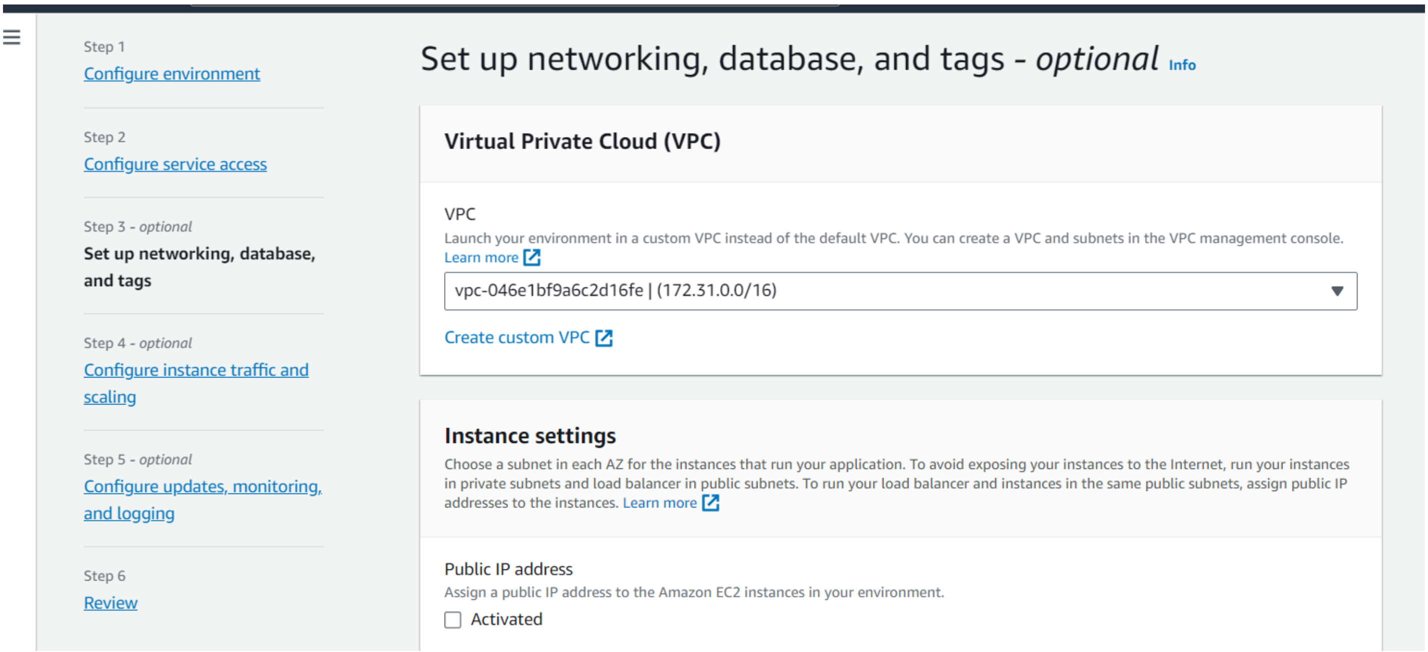
Choose a key pair

EC2 instance profile

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

mywebapprole

Step 11- Select VPC which was been provided there



Step 1
Configure environment

Step 2
Configure service access

Step 3 - *optional*
Set up networking, database, and tags

Step 4 - *optional*
Configure instance traffic and scaling

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Set up networking, database, and tags - *optional* Info

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-046e1bf9a6c2d16fe | (172.31.0.0/16) ▼

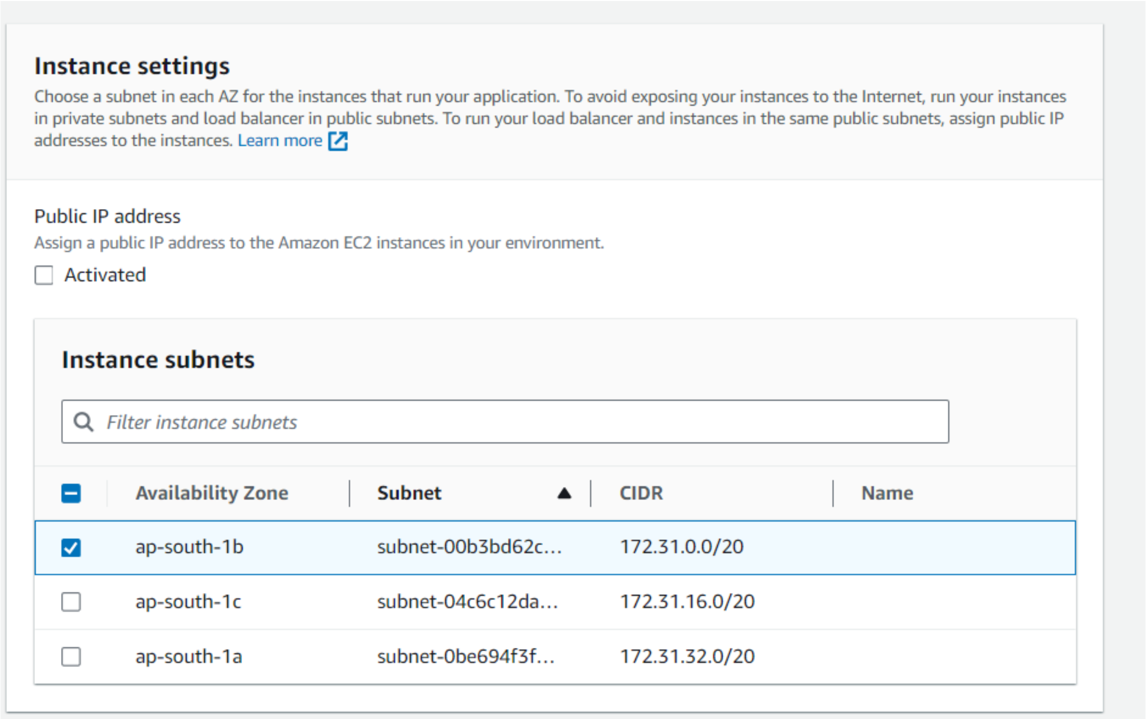
Create custom VPC

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

Step 12- Select the IP Address Same as above which was been provided in VPC in Instance and Database



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

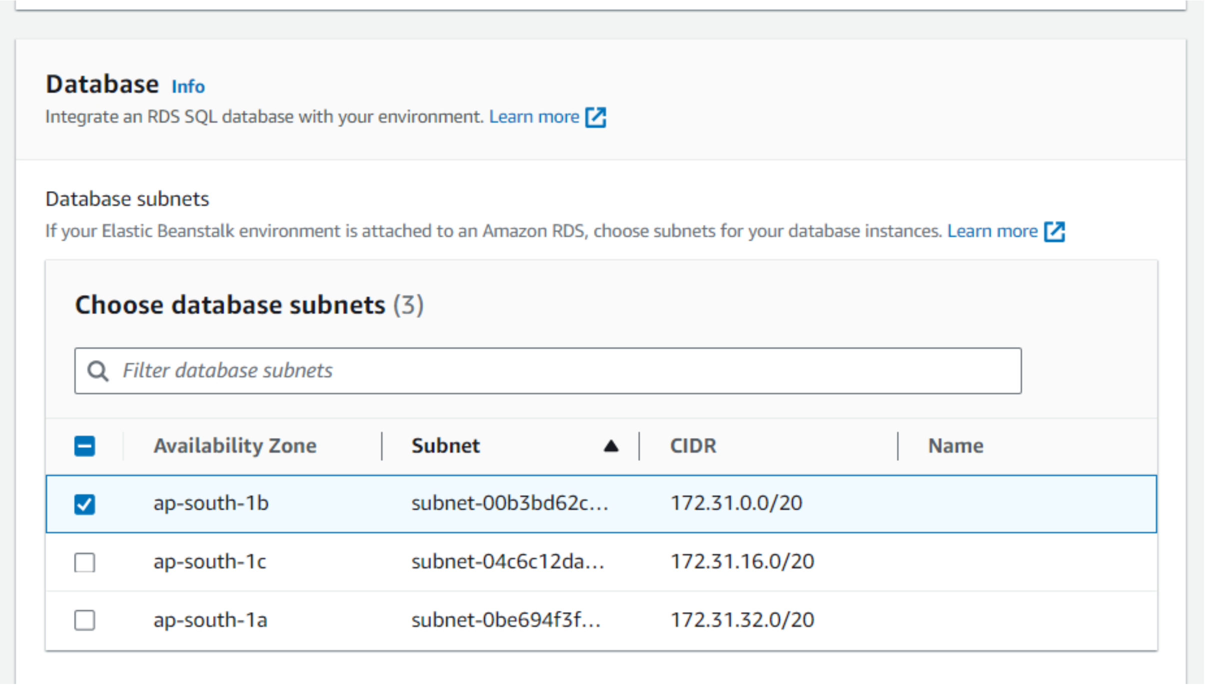
Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
☐ Activated

Instance subnets

Filter instance subnets

▬	Availability Zone	Subnet ▲	CIDR	Name
☑	ap-south-1b	subnet-00b3bd62c...	172.31.0.0/20	
☐	ap-south-1c	subnet-04c6c12da...	172.31.16.0/20	
☐	ap-south-1a	subnet-0be694f3f...	172.31.32.0/20	

For Database



Database Info

Integrate an RDS SQL database with your environment. Learn more

Database subnets

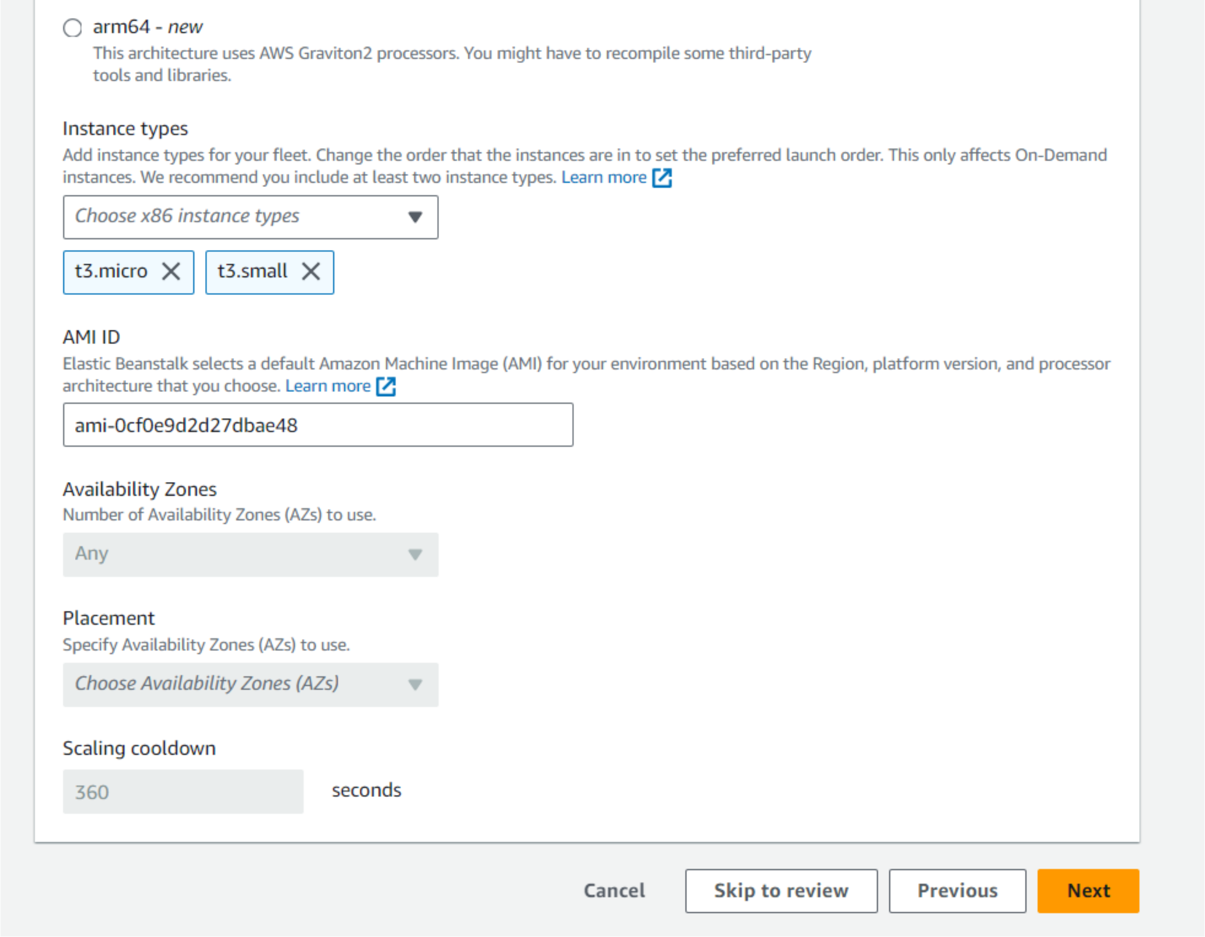
If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. Learn more

Choose database subnets (3)

Filter database subnets

☐	Availability Zone	Subnet ▲	CIDR	Name
☑	ap-south-1b	subnet-00b3bd62c…	172.31.0.0/20	
☐	ap-south-1c	subnet-04c6c12da…	172.31.16.0/20	
☐	ap-south-1a	subnet-0be694f3f…	172.31.32.0/20	

Step 13- Skip the next steps and go to the review Section



○ 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-0cf0e9d2d27dbae48

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

Any

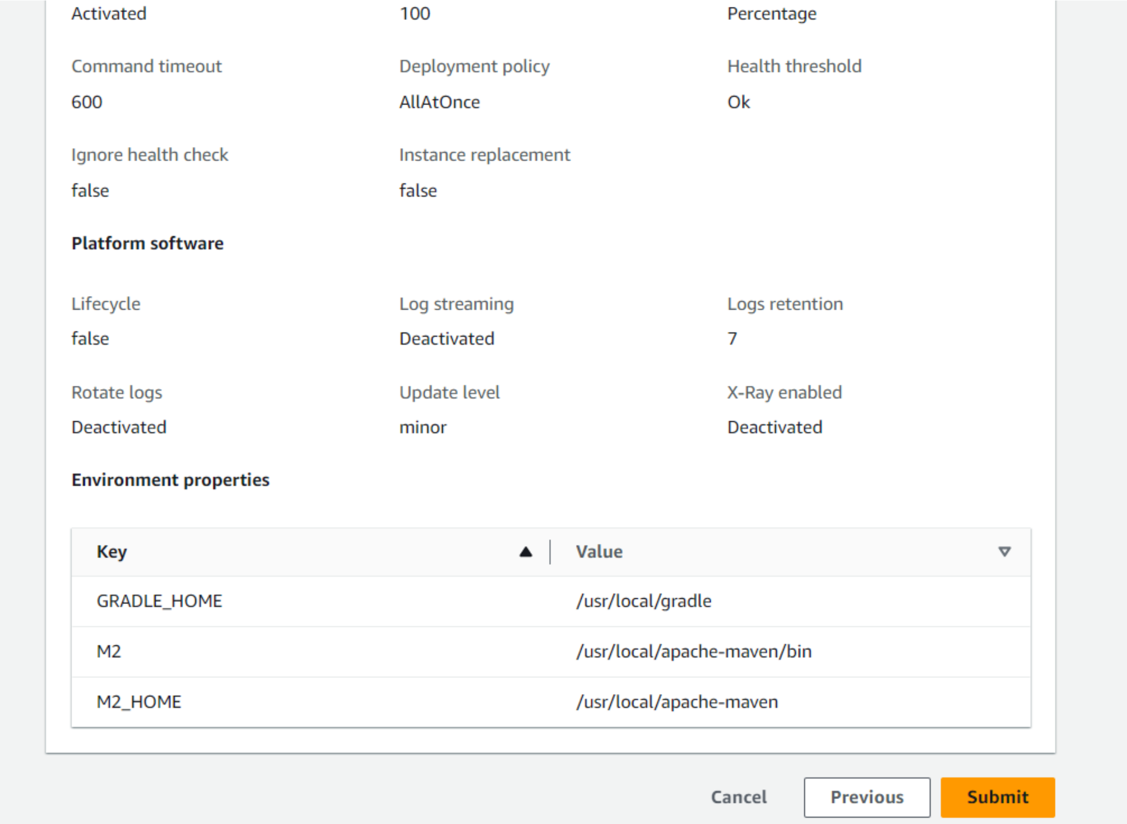
Placement
Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs)

Scaling cooldown

360 seconds

Step 14- Submit the Review



Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	
false	false	

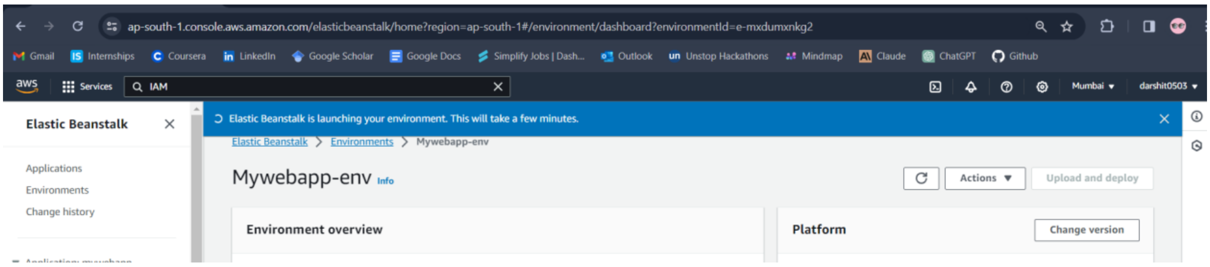
Platform software

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

Step 15- After the review the Elastic BeanStalk will run environment and it will take some few minutes



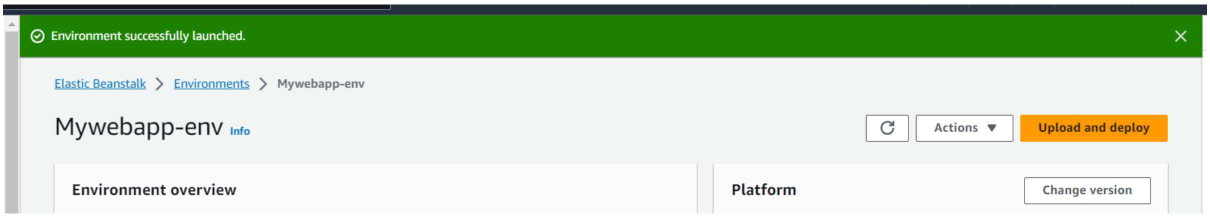
ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/environment/dashboard?environmentId=e-mxdumxnkg2

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

Elastic Beanstalk > Environments > Mywebapp-env

Mywebapp-env Info

Step 16- Environment runs successfully

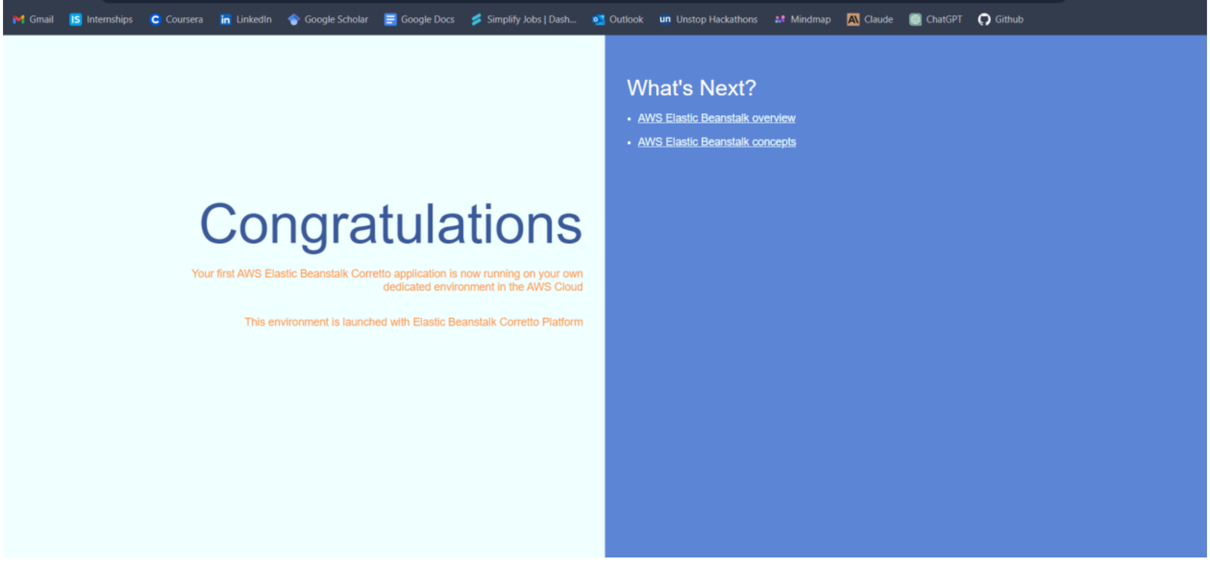


Environment successfully launched.

Elastic Beanstalk > Environments > Mywebapp-env

Mywebapp-env Info

Step 17- Output of the following



Congratulations

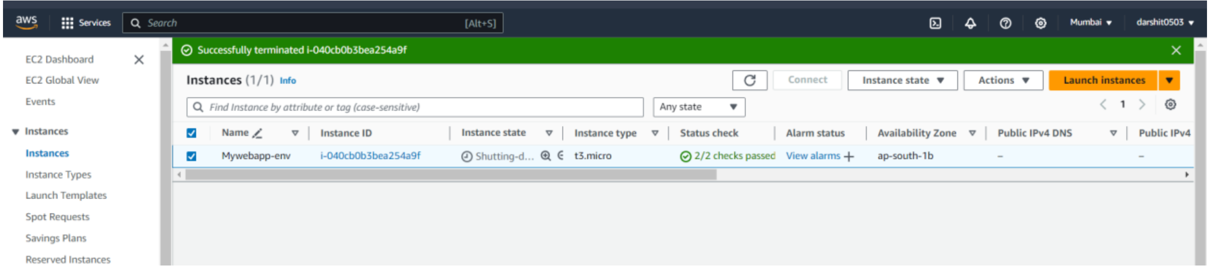
Your first AWS Elastic Beanstalk Corretto application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Corretto Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts

Step 18- Terminate the following Instance



Successfully terminated i-040cb0b3bea254a9f

Instances (1/1) Info

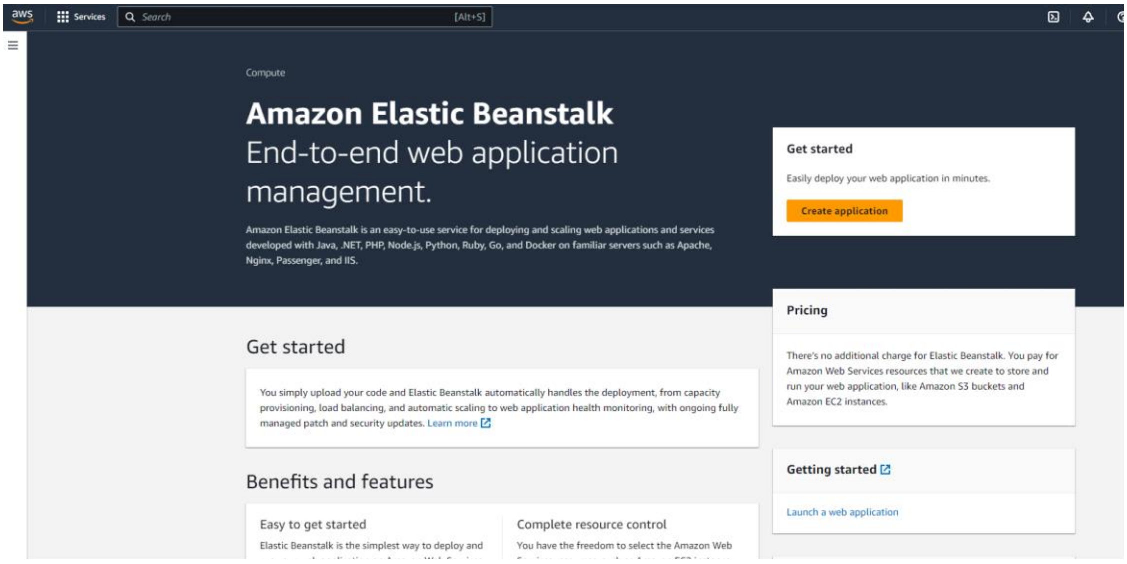
Find Instance by attribute or tag (case-sensitive)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
☑	Mywebapp-env	i-040cb0b3bea254a9f	Shutting-d...	t3.micro	2/2 checks passed	View alarms +	ap-south-1b	–	–

For Tomcat

Elastic Beanstalk by using Tomcat Application

1. Go to your Elastic BeanStalk



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.

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

Getting started

Launch a web application

Benefits and features

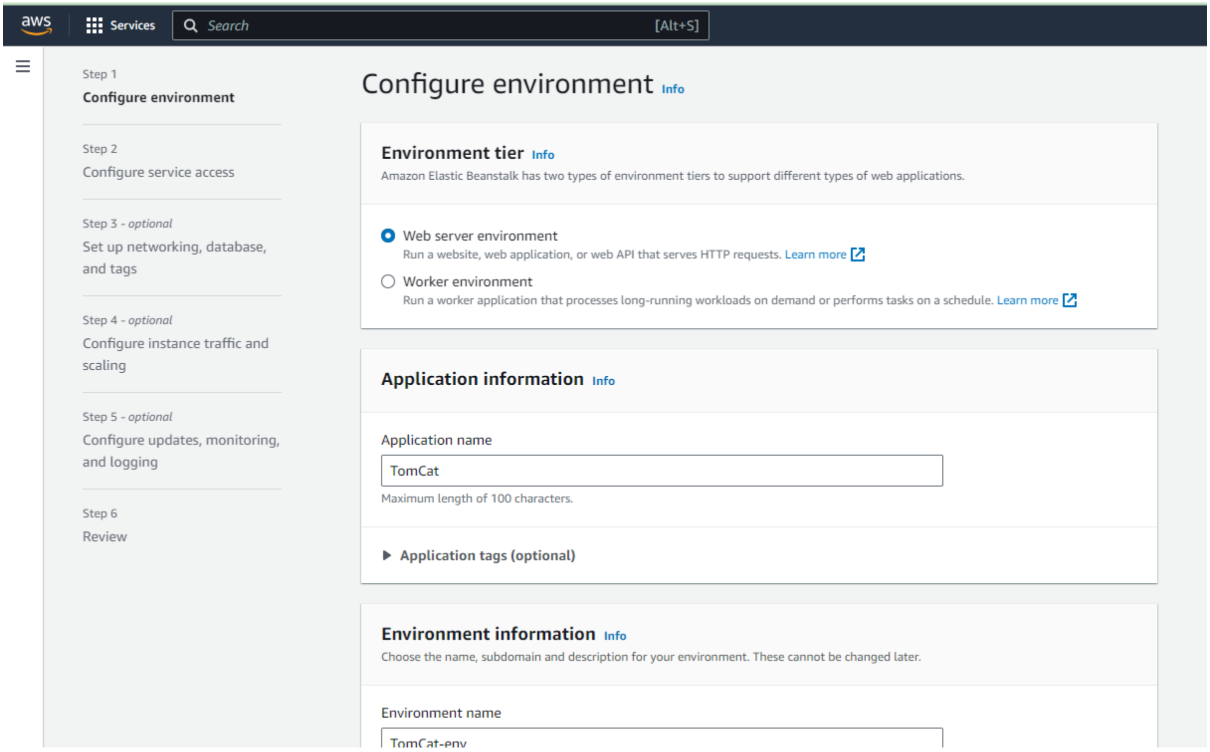
Easy to get started

Elastic Beanstalk is the simplest way to deploy and

Complete resource control

You have the freedom to select the Amazon Web

2. Provide the name of the Application



Step 1
Configure environment

Step 2
Configure service access

Step 3 - *optional*
Set up networking, database, and tags

Step 4 - *optional*
Configure instance traffic and scaling

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

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: TomCat

Maximum length of 100 characters.

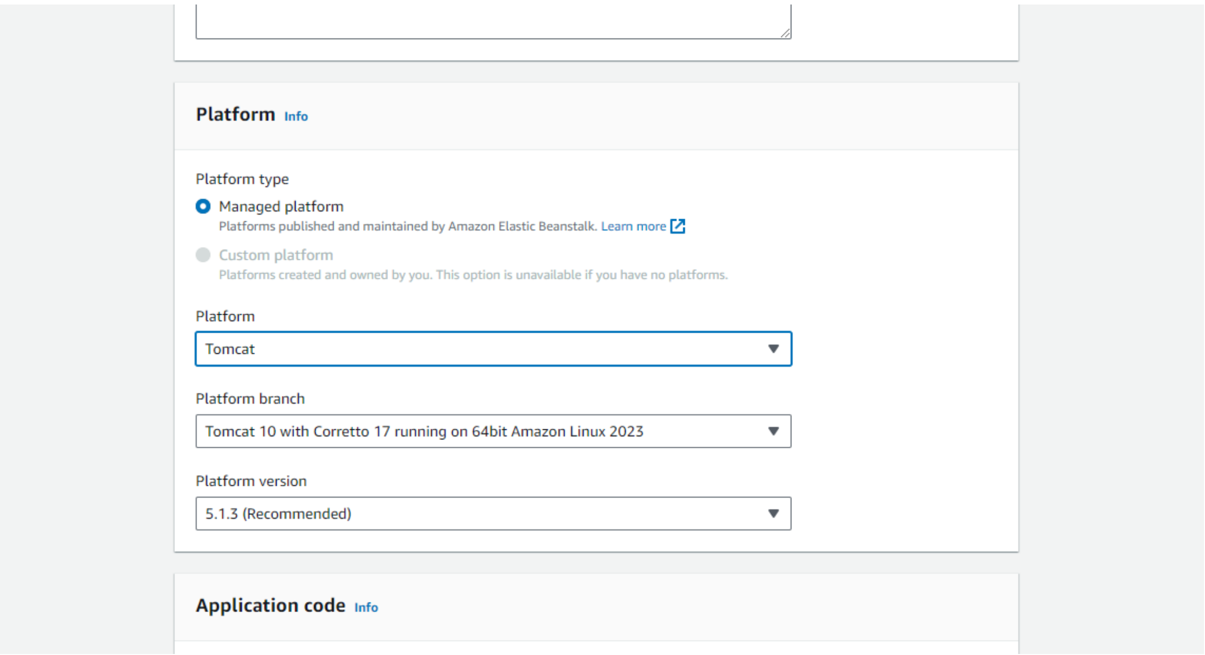
▶ Application tags (optional)

Environment information Info

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

Environment name: TomCat-env

3. Choose the Platform as TomCat



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.

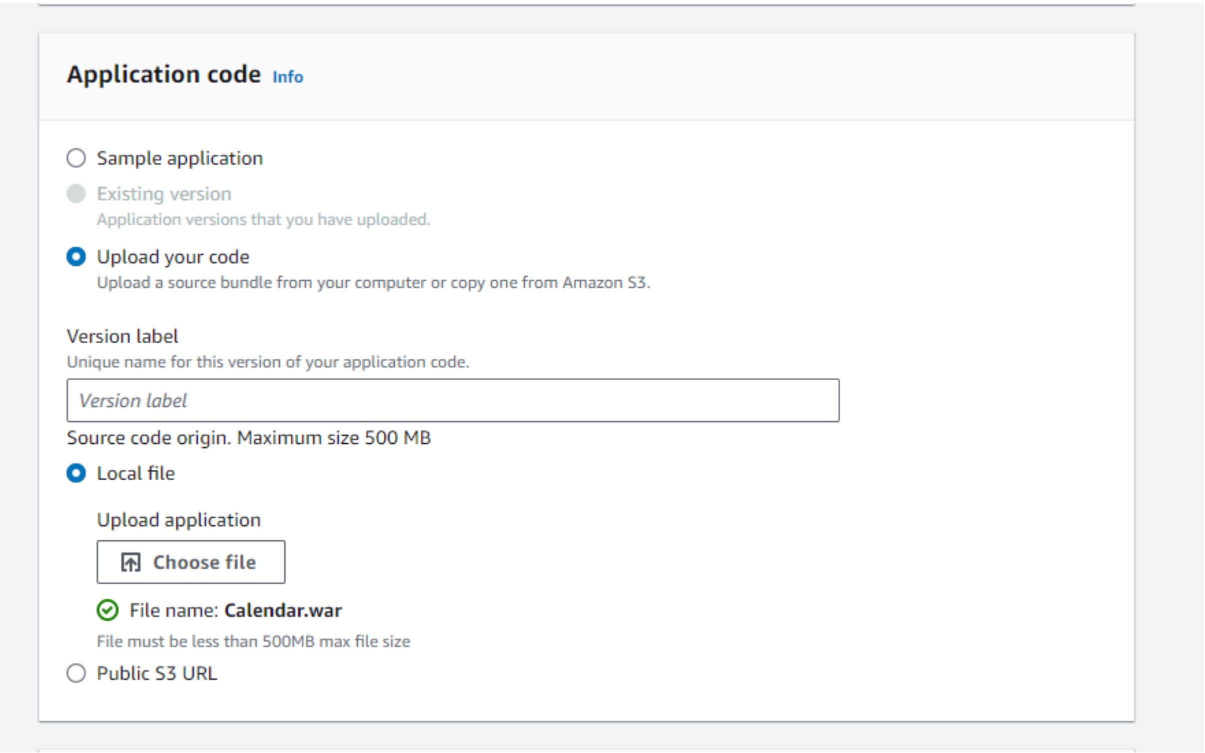
Platform: Tomcat

Platform branch: Tomcat 10 with Corretto 17 running on 64bit Amazon Linux 2023

Platform version: 5.1.3 (Recommended)

Application code Info

4. Download “calendar.war” File and Upload the 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.

Version label

Source code origin. Maximum size 500 MB

● Local file

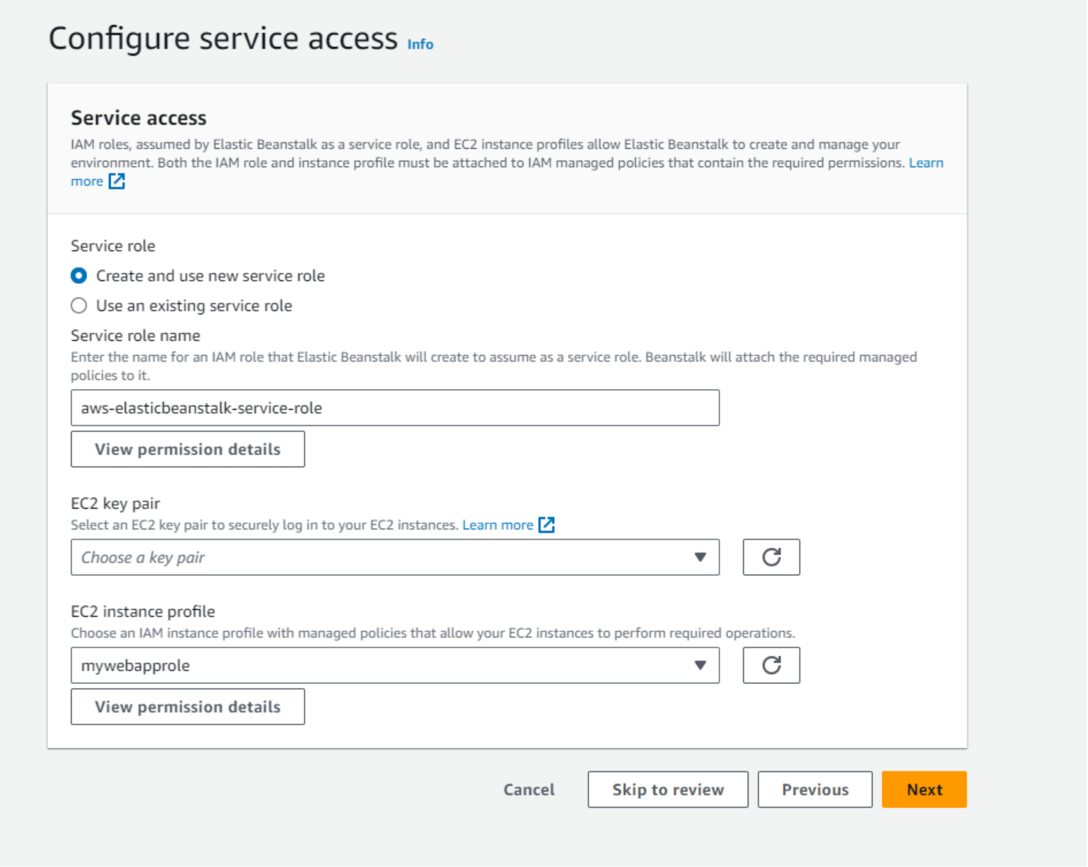
Upload application

✓ File name: **Calendar.war**

File must be less than 500MB max file size

○ Public S3 URL

5. Create a New service Role and use the existing EC2 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.

aws-elasticbeanstalk-service-role

EC2 key pair

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

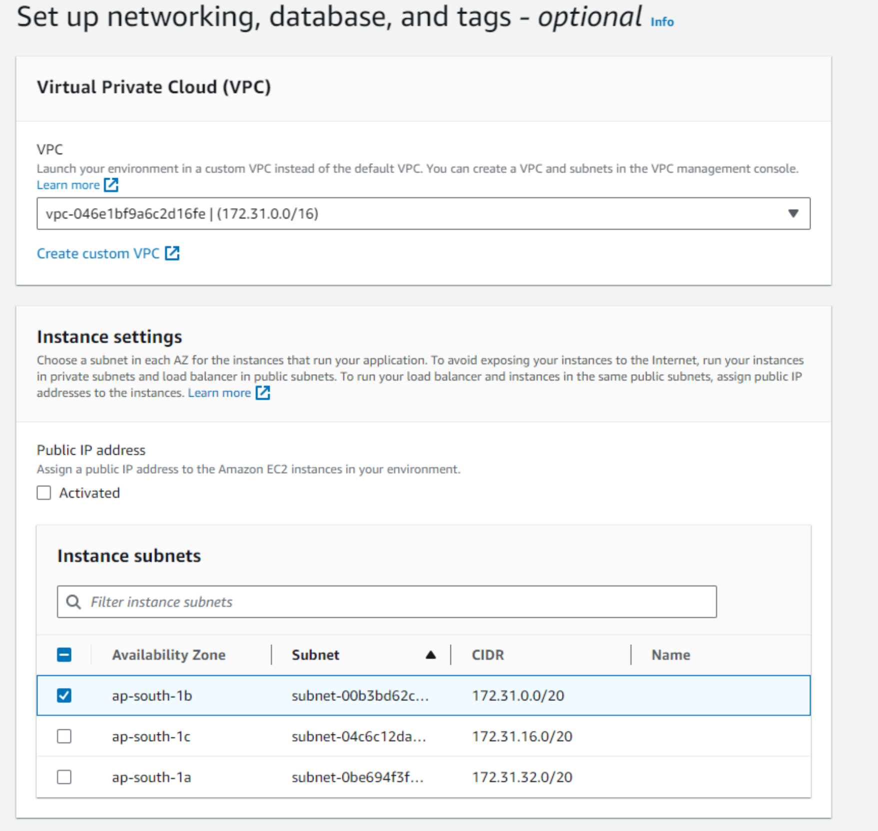
Choose a key pair

EC2 instance profile

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

mywebapprole

6. Use the VPC and select the Same IP Address in Instance and Database



Set up networking, database, and tags - *optional* Info

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-046e1bf9a6c2d16fe | (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.

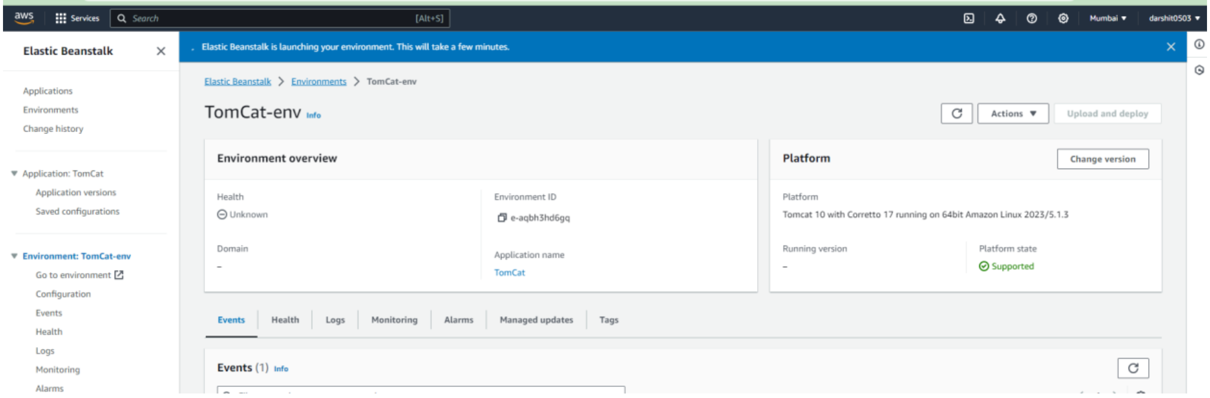
□ Activated

Instance subnets

Filter instance subnets

☐	Availability Zone	Subnet ▲	CIDR	Name
☑	ap-south-1b	subnet-00b3bd62c...	172.31.0.0/20	
□	ap-south-1c	subnet-04c6c12da...	172.31.16.0/20	
□	ap-south-1a	subnet-0be694f3f...	172.31.32.0/20	

7. Skip to the Review Section and Submit it and you will see the Environment will start running



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

Elastic Beanstalk > Environments > TomCat-env

TomCat-env Info

Environment overview

Health: Unknown

Environment ID: e-aqbh3hd6gq

Domain: -

Application name: TomCat

Platform

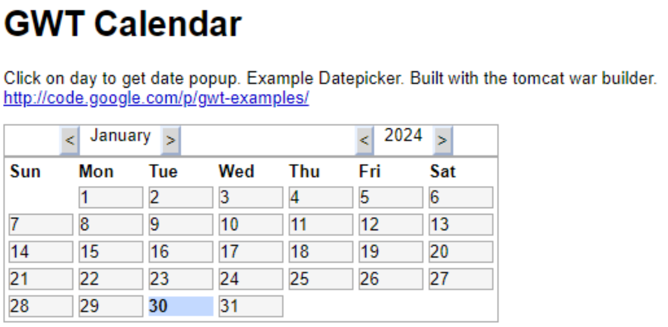
Platform: Tomcat 10 with Corretto 17 running on 64bit Amazon Linux 2023/5.1.3

Running version: -

Platform state: Supported

Events (1) Info

8. The Output is seen as follows



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			