

Name: Anthea Gamjya

Roll no: A022

Sap Id: 86092300023

Submission Date: 30/1/24

Practical 2: Platform as a service using AWS

Implementation of PAAS using Elastic Beanstalk

Writeup:

- **Platform as a service:** Platform as a Service (PaaS) is a cloud computing service model that provides a platform allowing customers to develop, run, and manage applications without dealing with the complexities of infrastructure management. PaaS sits in the middle of the cloud computing service models, with Infrastructure as a Service (IaaS) providing the basic computing infrastructure and Software as a Service (SaaS) delivering complete applications.

Key characteristics and features of Platform as a Service:

1. Development Tools: Provides tools for application development.
2. Automated Deployment: Offers automated provisioning and scaling.
3. Multi-Tenancy: Supports multiple users on a shared platform.
4. Scalability: Allows applications to easily scale based on demand.
5. Database Integration: Includes built-in database services.
6. Middleware Services: Facilitates communication and integration.
7. Security and Compliance: Manages infrastructure security; users handle application security.
8. Cost Efficiency: Cost-effective with pay-as-you-go pricing.

- **Elastic Beanstalk:** Amazon Elastic Beanstalk is a fully managed Platform as a Service (PaaS) offering from Amazon Web Services (AWS). It simplifies the deployment and management of applications by abstracting the underlying infrastructure, allowing developers to focus on writing code without dealing with the complexities of infrastructure management.

Key features of Elastic Beanstalk include:

1. Platform Support: Elastic Beanstalk supports various programming languages and frameworks, such as Java, .NET, Node.js, Python, Ruby, Go, and Docker.
2. Easy Deployment: Developers can deploy their applications using a simple web-based interface, command-line tools, or directly from their integrated development environment (IDE).
3. Auto-Scaling: Elastic Beanstalk can automatically scale the number of instances running an application based on demand, ensuring optimal performance and resource utilization.

4. Integration with Other AWS Services: Developers can leverage other AWS services like Amazon RDS for databases, Amazon S3 for storage, and more, seamlessly integrated with Elastic Beanstalk.

- **Components of Beanstalk:**

The key components of Elastic Beanstalk include:

1. Application: This represents the entire application that you want to deploy on Elastic Beanstalk. It includes the source code, dependencies, and configurations required for your application to run.
 2. Environment: An environment in Elastic Beanstalk is a collection of AWS resources (such as Amazon EC2 instances, load balancers, and Auto Scaling groups) that are created to run your application. Environments can be created for different stages of your application (e.g., development, testing, production).
 3. Version: A version in Elastic Beanstalk is a specific iteration of your application code and configuration. When you deploy your application, you create a new version, and Elastic Beanstalk manages the deployment of that version to the specified environment.
 4. Application Version Repository: Elastic Beanstalk allows you to store and manage different versions of your application code and configurations. You can use either the Elastic Beanstalk console or the AWS Management Console to manage these versions.
 5. Application Server: The application server is the software that runs your application. Elastic Beanstalk supports various programming languages and frameworks, and it automatically configures the appropriate application server based on your application's requirements.
 6. Web Server: Elastic Beanstalk uses web servers (such as Apache or Nginx) to handle incoming requests and distribute them to the application server instances.
 7. Auto Scaling Groups: Elastic Beanstalk leverages Auto Scaling groups to automatically adjust the number of instances running your application based on demand. This ensures that your application can handle varying levels of traffic and load.
 8. Load Balancer: Elastic Beanstalk includes a load balancer to distribute incoming traffic across multiple instances of your application, improving fault tolerance and ensuring that each instance receives a balanced share of requests.
 9. Database: While Elastic Beanstalk primarily focuses on the application layer, it can integrate with other AWS services, including Amazon RDS for managing databases.
 10. Logging and Monitoring: Elastic Beanstalk integrates with Amazon CloudWatch for monitoring the performance and health of your application. You can access logs and metrics to troubleshoot issues and optimize your application.
- IAM: IAM stands for Identity and Access Management. It is a service provided by Amazon Web Services (AWS) that enables the management of access to AWS resources securely. IAM allows you to control who can access your AWS resources

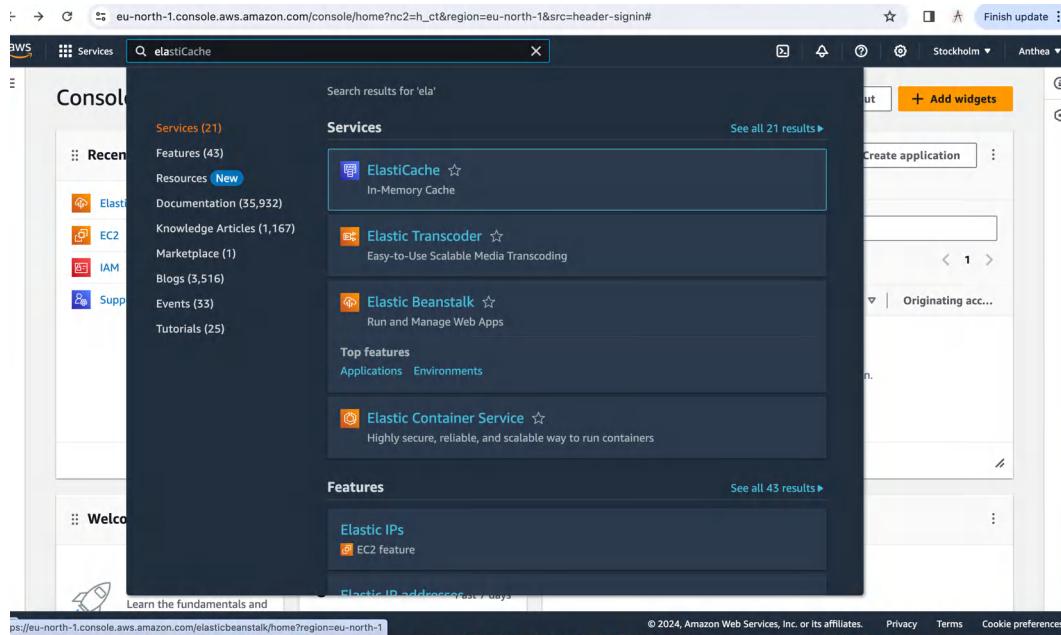
and what actions they can perform. Here are key components and concepts associated with AWS Identity and Access Management (IAM):

1. **Groups:** Groups are collections of IAM users. By associating users with groups, you can manage permissions for multiple users collectively. This simplifies the process of granting the same set of permissions to multiple users.
2. **Roles:** IAM roles define a set of permissions for making AWS service requests. Roles are not associated with a specific user or group; instead, they are assumed by AWS services, EC2 instances, or applications. Roles are often used to grant permissions to AWS resources.
3. **Permissions:** Permissions in IAM are associated with actions and resources. They determine what actions (e.g., read, write, delete) a user, group, or role can perform on specific AWS resources.
4. **IAM Console:** The IAM Console is a web-based interface provided by AWS for managing IAM users, groups, roles, and policies. Administrators can use the console to configure and monitor IAM resources.

For server

Steps:

- 1) Sign into your aws account
- 2) In services search elastic beanstalk



- 3) The following page will be loaded
- 4) Now create a new application

The screenshot shows the AWS Elastic Beanstalk Applications page. The left sidebar has 'Elastic Beanstalk' selected under 'Applications'. Below it, 'Recent environments' list 'Riverdale-env' and 'Anthea-env'. The main content area shows a table titled 'Applications (2) Info' with columns: Application name, Environments, Date created, and Last modified. It lists 'anthea' (Environments: Anthea-env (terminated), Date created: January 29, 2024 20:48:33, Last modified: January 29, 2024 20:48:33) and 'riverdale' (Environments: Riverdale-env (terminated), Date created: January 29, 2024 21:35:46, Last modified: January 29, 2024 21:35:46). A 'Create application' button is at the top right.

5) Write a application name and write some description > create

The screenshot shows the 'Create new application' wizard. The first step, 'Application information', has 'Application name' set to 'mikrokosmos' and 'Description' set to 'anything'. The second step, 'Tags', shows a note about applying up to 50 tags and a 'Add new tag' button. At the bottom are 'Cancel' and 'Create' buttons.

6) Now create an environment

The screenshot shows the 'Application mikrokosmos environments (0) Info' page. The left sidebar shows 'Application: mikrokosmos' selected. The main content area shows a table with columns: Environment, Health, Date created, Domain, Running v..., Platform, and Platform. A note says 'No environments' and 'No environments currently exist for this application.' A 'Create environment' button is at the bottom.

7) In configure environment the following is seen

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
mikrokosmos
Maximum length of 100 characters.

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

Environment name

8) Select a platform for server

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

.NET on Windows Server

Platform branch

IIS 10.0 running on 64bit Windows Server 2019

Platform version

2.13.2 (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

9) Now duplicate the we page and now search for IAM in services

Search results for 'iam'

Services (66)

- Features (110)
- Resources [New](#)
- Documentation (146,927)
- Knowledge Articles (357)
- Marketplace (60)
- Blogs (10,179)
- Events (263)
- Tutorials (87)

Features

- See all 110 results▶
- IAM** ☆
Manage access to AWS resources
- IAM Identity Center** ☆
Manage workforce user access to multiple AWS accounts and cloud applications
- MediaStore** ☆
Store and deliver video assets for live or on-demand media workflows
- MediaLive** ☆
Convert video inputs into live outputs for broadcast and streaming delivery
- Compliance**
Systems Manager feature

10) The IAM dashboard will be loaded

The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with navigation links like 'Dashboard', 'Access management', 'Access reports', and 'Identity providers'. The main area has sections for 'Security recommendations' (with a warning about root user MFA), 'IAM resources' (listing 4 Roles), and 'AWS Account' (Account ID: 339713188308). A 'Quick Links' section includes a link to 'My security credentials'.

11) Click on Roles

12) Now create a new role

The screenshot shows the 'Roles' page in the IAM service. It lists four existing roles: 'AWSServiceRoleForAutoScaling', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', and 'river'. There are buttons for 'Create role' and 'Manage'.

13) Select EC2 in use case for trusted entity

The screenshot shows the 'Create role' wizard at Step 1: 'Select trusted entity'. It shows options for 'Trusted entity type' (AWS service, AWS account, Web identity, SAML 2.0 federation, Custom trust policy) and 'Use case' (EC2, EC2 Role for AWS Systems Manager, EC2 Spot Fleet Role, EC2 - Spot Fleet Auto Scaling, EC2 - Spot Fleet Tagging, EC2 - Spot Instances, EC2 - Spot Fleet, EC2 - Scheduled Instances). The 'EC2' option is selected.

14) Next

15) In add permissions search for beanstalk

Add permissions [Info](#)

Permissions policies (910) [Info](#)

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

Filter by Type All types 14 matches

| Policy name | Type | Description |
|---|-------------|---|
| AdministratorAccess-AWSEI... | AWS managed | Grants account administrative permis... |
| AWSElasticBeanstalkCustom... | AWS managed | Provide the instance in your custom pl... |
| AWSElasticBeanstalkEnhanc... | AWS managed | AWS Elastic Beanstalk Service policy f... |
| AWSElasticBeanstalkManag... | AWS managed | This policy is for the AWS Elastic Bean... |
| AWSElasticBeanstalkMultico... | AWS managed | Provide the instances in your multicon... |
| AWSElasticBeanstalkReadOnly | AWS managed | Grants read-only permissions. Explicitl... |
| AWSElasticBeanstalkRoleCore | AWS managed | AWSElasticBeanstalkRoleCore (Elastic ... |
| AWSElasticBeanstalkRoleCWL | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleECS | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleRDS | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleSNS | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleWeb | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkWebTier | AWS managed | Provide the instances in your web serv... |
| AWSElasticBeanstalkWorker | AWS managed | Provide the instances in your worker e... |

16) Select the following 3 -> docker,webTier,WorkerTier

Set permissions boundary - optional

Cancel Previous Next

17) Next enter the Role details

Role details

Role name: euphoria

Description: Allows EC2 instances to call AWS services on your behalf.

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-         ...
11-       ]
12-     }
13-   ]
14- }

```

18) create role

Step 2: Add permissions

| 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.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create role

19) Role is created

Identity and Access Management (IAM)

Role euphoria created.

Roles (5) 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 |
|---------------------------------|--|----------------|
| AWSServiceRoleForAutoScaling | AWS Service: autoscaling (Service-Linker) | 34 minutes ago |
| AWSServiceRoleForSupport | AWS Service: support (Service-Linker) | - |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service-Linker) | - |
| euphoria | AWS Service: ec2 | - |
| river | AWS Service: ec2 | 38 minutes ago |

Roles Anywhere Info

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

Manage

20) Go back to original web page of aws server(not the duplicate one)

21) Refresh the service roles and select the new created role

The screenshot shows the 'Configure service access' step of the AWS Elastic Beanstalk setup wizard. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Configure environment'. Step 2 is 'Configure service access', which is currently selected. Step 3 is 'Set up networking, database, and tags'. Step 4 is 'Configure instance traffic and scaling'. Step 5 is 'Configure updates, monitoring, and logging'. Step 6 is 'Review'. The main content area is titled 'Service access'. It explains that 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. There are two options: 'Create and use new service role' (radio button not selected) and 'Use an existing service role' (radio button selected). Below this, there is a section for 'Existing service roles' with a dropdown menu containing 'euphoria'. Other sections include 'EC2 key pair' (dropdown menu containing 'Choose a key pair') and 'EC2 instance profile' (dropdown menu containing 'euphoria'). At the bottom, there is a 'View permission details' button. Navigation buttons at the bottom right include 'Cancel', 'Skip to review', 'Previous', and a large orange 'Next' button.

22) For VPC there will be a default address which we select from the dropdown

The screenshot shows the 'Set up networking, database, and tags - optional' step of the AWS Elastic Beanstalk setup wizard. The sidebar on the left shows steps 1 through 6. Step 1 is 'Configure environment'. Step 2 is 'Configure service access', which is currently selected. Step 3 is 'Set up networking, database, and tags'. Step 4 is 'Configure instance traffic and scaling'. Step 5 is 'Configure updates, monitoring, and logging'. Step 6 is 'Review'. The main content area is titled 'Virtual Private Cloud (VPC)'. It explains that you can launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. A dropdown menu shows 'vpc-097e5a4365f5c8ba5 | (172.31.0.0/16)'. Below this is a 'Create custom VPC' button. The next section is 'Instance settings', which discusses choosing a subnet in each AZ for instances. It includes a note about avoiding exposing instances to the Internet by running them in private subnets and load balancers in public subnets. A checkbox for 'Activated' is checked. The final section is 'Instance subnets', which contains a table header with columns: Availability Zone, Subnet, CIDR, and Name. The table body is currently empty. Navigation buttons at the bottom right include 'Skip to review', 'Previous', and a large orange 'Next' button.

23) For subnet and database select the same cidr as the vpc address

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

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

Choose database subnets (3)

| Availability Zone | Subnet | CIDR | Name |
|---|---------------------|----------------|------|
| <input type="checkbox"/> eu-north-1b | subnet-02c5b89cd... | 172.31.32.0/20 | |
| <input checked="" type="checkbox"/> eu-north-1c | subnet-0980fd2da... | 172.31.0.0/20 | |
| <input type="checkbox"/> eu-north-1a | subnet-0c6aa5382... | 172.31.16.0/20 | |

24) Keep the default options for configure instance

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 instance traffic and scaling - optional

Instances Info
Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type: Container default

Size: 8 GB

IOPS: 100 IOPS

Throughput: 125 MiB/s

Amazon CloudWatch monitoring
The time interval between when metrics are reported from the EC2 instances

Monitoring interval

25) Similarly for configure updates

Configure updates, monitoring, and logging - optional

Monitoring Info

Health reporting
Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The EnvironmentHealth custom metric is provided free with enhanced health 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 ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming
 Activated (standard CloudWatch charges apply.)

Retention

26) Now review the dets

The screenshot shows the 'Review' step of the Elastic Beanstalk environment configuration wizard. It is divided into two main sections: 'Step 1: Configure environment' and 'Step 2: Configure service access'.
Step 1: Configure environment
This section contains 'Environment information' fields:

- Environment tier: Web server environment
- Application name: mikrokosmos
- Environment name: Mikrokosmos-env
- Application code: Sample application
- Platform: arn:aws:elasticbeanstalk:eu-north-1::platform/IIS 10.0 running on 64bit Windows Server 2019/2.13.2

Step 2: Configure service access
This section contains 'Service access' fields:

- Service role: EC2 instance profile
- EC2 instance profile: arn:aws:iam::339713188308:role/eup euphoria

A sidebar on the left lists configuration steps from Step 1 to Step 6, with 'Review' currently selected.

27) Environment is being launched

The screenshot shows the 'Mikrokosmos-env' environment page in the AWS Elastic Beanstalk console. A prominent message at the top says 'Elastic Beanstalk is launching your environment. This will take a few minutes.' The environment overview table shows the following details:

| Health | Environment ID |
|---------|----------------|
| Pending | e-sjrmn2qmf |

The platform section indicates:

- Platform: IIS 10.0 running on 64bit Windows Server 2019/2.13.2
- Running version: -
- Platform state: Supported

Events tab shows 3 events, and the logs tab is visible below.

28) Environment launched

The screenshot shows the 'Mikrokosmos-env' environment page after successful launch. A green banner at the top says 'Environment successfully launched.' The environment overview table shows the following details:

| Health | Environment ID |
|---------|----------------|
| Warning | e-sjrmn2qmf |

The platform section indicates:

- Platform: IIS 10.0 running on 64bit Windows Server 2019/2.13.2
- Running version: -
- Platform state: -

Logs tab is visible below.

29) Go back to instances if the instance is still running

The screenshot shows the AWS EC2 Instances page. The search bar at the top has 'Find Instance by attribute or tag (case-sensitive)' and the filter 'Instance state = running'. A table below lists one instance: 'Mikrokosmos...' (Instance ID: i-033d266e1e397b10f), which is 'Running' (t3.large), has 2/2 checks passed, and is in the 'eu-north-1c' availability zone.

30) Go back to environment and click on domain

The screenshot shows the AWS Elastic Beanstalk Environment Overview page for 'Mikrokosmos-env'. It displays the environment overview, platform details (IIS 10.0 on Windows Server 2019/2.13.2), and event logs. The domain listed is 'Mikrokosmos-env.eba-kmbv2bv5.eu-north-1.elasticbeanstalk.com'.

31) Elastic beanstalk for server is now running

The screenshot shows a web browser displaying the AWS Elastic Beanstalk success page. The main message is 'Congratulations! Your AWS Elastic Beanstalk ASP.NET application is now running on your own dedicated environment in the AWS Cloud'. To the right, there are sections for 'What's Next?' (links to AWS Elastic Beanstalk overview, concepts, deployment tools, and log management), 'AWS SDK for .NET' (links to the SDK home, toolkit, developer center, documentation, and GitHub repository), and 'AWS .NET Services' (link to generate test events for AWS X-Ray Service).

Later terminate the instances and the environment

For python

- 1) Login to aws account and go to services and select elastic beanstalk

The image consists of two screenshots of the AWS Management Console.

The top screenshot shows the search results for "beanstalk". The search bar at the top contains "beanstalk". Below it, the "Services" section is expanded, showing:

- Features (27)
- Resources (New)
- Documentation (60,521)
- Knowledge Articles (157)
- Marketplace (9)
- Blogs (4,241)
- Events (51)
- Tutorials (14)

The "Services" section lists:

- Elastic Beanstalk (selected): Run and Manage Web Apps
- Managed Apache Flink: Fully managed, highly available, and secure service for Apache Flink
- IoT Events: Detect and respond to events from IoT sensors and Industrial IoT equipment
- CloudWatch: Monitor Resources and Applications

The "Features" section lists:

- Applications (selected): Elastic Beanstalk feature
- Environments (selected): Elastic Beanstalk feature

The bottom screenshot shows the "Amazon Elastic Beanstalk" welcome page. The URL is eu-north-1.console.aws.amazon.com/elasticbeanstalk/home?region=eu-north-1#welcome. The page has a dark header with the title "Amazon Elastic Beanstalk" and the subtitle "End-to-end web application management." Below the header, there's a "Get started" section with a button labeled "Create application". To the right, there's a "Pricing" section stating "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." At the bottom, there are sections for "Benefits and features" (Easy to get started, Complete resource control) and "Getting started" (Launch a web application).

- 2) Create an application
- 3) Write the application name
- 4) The environment name will be automatically updated

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
 .eu-north-1.elasticbeanstalk.com [Check availability](#)

Environment description

5) Select platform as python

6) Click on next

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

Platform branch

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.

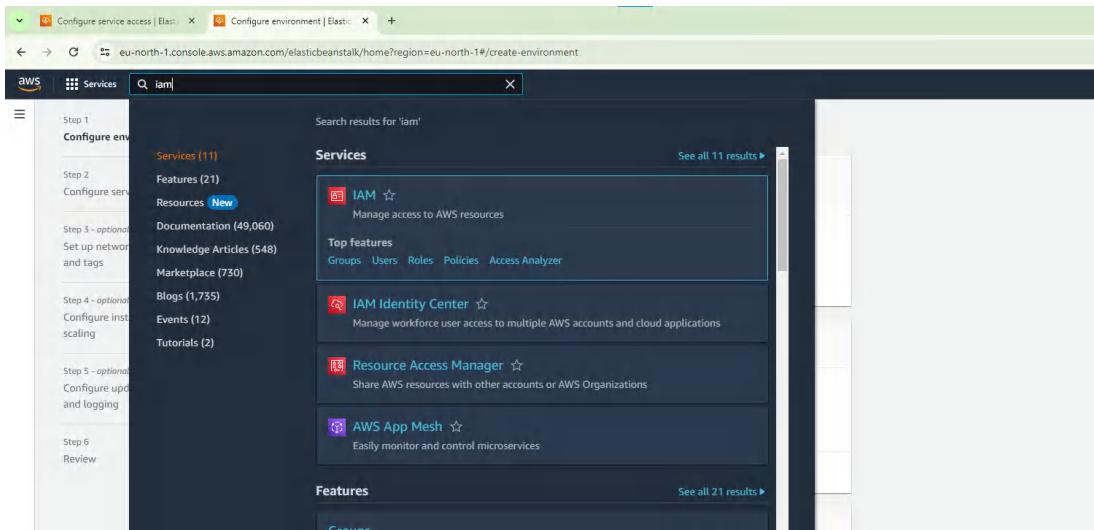
Presets [Info](#)

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

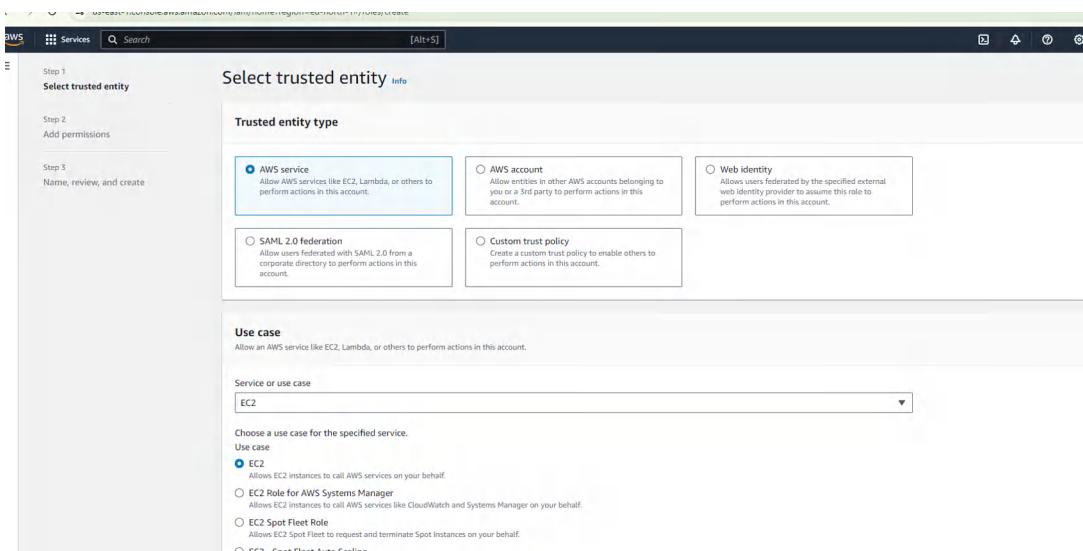
Configuration presets
 Single instance (free tier eligible)
 Single instance (using spot instance)
 High availability
 High availability (using spot and on-demand instances)
 Custom configuration

[Cancel](#) [Next](#)

7) Now duplicate the we page and in services search for IAM



- 8) Click on IAM and select Roles and click create a new Role
- 9) In select trusted entity ->For use case select EC2



- 10) Next
- 11) In permission policies search Beanstalk
- 12) Select the following 3 -> Docker,webTier,workerTier

Permissions policies (3/908) [Info](#)

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

Filter by Type All types 14 matches

| Policy name | Type | Description |
|--|-------------|-------------|
| <input type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk | AWS managed | Grants |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | Provide |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | AWS Ela |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | This pol |
| <input checked="" type="checkbox"/> AWSAdministratorAccessRole | AWS managed | Provide |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | Grants r |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | AWSEla |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | (Elasti |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | (Elasti |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | (Elasti |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | (Elasti |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS managed | Provide |
| <input checked="" type="checkbox"/> AWSAdministratorAccessRole | AWS managed | Provide |

13) For creating new role -> write a name

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

14) Role anthea will be created

Configure service access | Elastic | Roles | IAM | Global | us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/roles

aws Services Search [Alt+S] Global Arthea

Identity and Access Management (IAM)

Role anthea created.

IAM > Roles

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.

| Role name | Trusted entities | Last activity |
|---|---------------------------------------|---------------|
| <input type="checkbox"/> anthea | AWS Service: ec2 | - |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS Service: support (Service-Linker) | - |
| <input type="checkbox"/> AWSAdministratorAccessRole | AWS Service: trustedadvisor (Service) | - |

Roles Anywhere Info

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

15) Go back to the previous web page (not the duplicated one)

16) Refresh the service roles and add your created role

The screenshot shows the 'Configure service access' step of the Elastic Beanstalk environment creation wizard. The left sidebar lists optional steps: Step 3 (Set up networking, database, and tags), Step 4 (Configure instance traffic and scaling), and Step 5 (Configure updates, monitoring, and logging). The main panel is titled 'Service access' and contains the following configuration:

- Service role:** A radio button is selected for "Use an existing service role".
- Existing service roles:** A dropdown menu shows "anthea" selected.
- EC2 key pair:** A dropdown menu shows "Choose a key pair" selected.
- EC2 instance profile:** A dropdown menu shows "anthea" selected.

At the bottom are buttons for "Cancel", "Skip to review", "Previous", and a highlighted "Next" button.

17) For VPC there will be a default address select that

18) For subnets select the same number mentioned in the address

The screenshot shows the 'VPC' configuration step of the Elastic Beanstalk environment creation wizard. The left sidebar lists optional steps: Step 3 (Set up networking, database, and tags), Step 4 (Configure instance traffic and scaling), and Step 5 (Configure updates, monitoring, and logging). The main panel includes:

- VPC:** A note says "Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console." Below is a dropdown menu showing "vpc-097e5a4365f5c8ba5 | (172.31.0.0/16)" and a "Create custom VPC" link.
- Instance settings:** A note says "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." Below is a section for "Public IP address" with an "Activated" checkbox.
- Instance subnets:** A table lists subnets:

| Availability Zone | Subnet | CIDR | Name |
|-------------------|---------------------|----------------|------|
| eu-north-1b | subnet-02c5b89cd... | 172.31.32.0/20 | |
| eu-north-1c | subnet-0980fd2da... | 172.31.0.0/20 | |
| eu-north-1a | subnet-0c6aa5382... | 172.31.16.0/20 | |
- Database:** A note says "Integrate an RDS SQL database with your environment." Below is a section for "Database subnets" with a note about attaching to Amazon RDS.

19) In the following page don't make any changes just go on next

The screenshot shows the 'Configure updates, monitoring, and logging - optional' section of the AWS Elastic Beanstalk configuration interface. It includes settings for Health reporting (Enhanced), CloudWatch Custom Metrics (Instance and Environment), Log streaming (Activated), Retention (7 days), and Lifecycle (Keep logs after terminating environment).

20) Now review

The screenshot shows the 'Review' step of the configuration wizard, divided into three sections: Step 1: Configure environment, Step 2: Configure service access, and Step 3: Set up networking, database, and tags.

Step 1: Configure environment

| Environment information | |
|--|--------------------|
| Environment tier | Application name |
| Web server environment | webapp |
| Environment name | Application code |
| Webapp-env | Sample application |
| Platform | |
| arn:aws:elasticbeanstalk:eu-north-1::platform/Python | |
| 3.11 running on 64bit Amazon Linux 2023/4.0.7 | |

Step 2: Configure service access

| Service access <small>Info</small> | |
|--|----------------------|
| 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 instance profile |
| arn:aws:iam::339713188308:role/aws | anthea |
| -elasticbeanstalk-service-role | |

Step 3: Set up networking, database, and tags

21) next

22) Beanstalk will be launching your environment

The screenshot shows the AWS Elastic Beanstalk console with the environment 'Webapp-env' selected. A blue banner at the top states 'Elastic Beanstalk is launching your environment. This will take a few minutes.' Below this, the 'Environment overview' section displays basic details: Health (Unknown), Environment ID (e-njw7hmitwp), Domain (-), Application name (webapp), and Platform (Python 3.11 running on 64bit Amazon Linux 2023/4.0.7). The 'Events' tab is active, showing two recent log entries:

- January 23, 2024 09:00:21 (UTC+5:30) INFO Using elasticbeanstalk-eu-north-1-339713188308 as Amazon S3 storage bucket for environment data.
- January 23, 2024 09:00:20 (UTC+5:30) INFO createEnvironment is starting.

23) Go back to instances and check if the instance is still running

The screenshot shows the AWS EC2 Instances page. The left sidebar is expanded to show the 'Instances' section, which includes 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', and 'Capacity Reservations'. The main content area displays a table of instances:

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 IP | Elastic IP |
|------------|--------------------|----------------|---------------|--------------|--------------|-------------------|--------------------------|----------------|--------------|
| Webapp-env | i-08e94b7ad106746e | Running | t3.micro | Initializing | | eu-north-1c | ec2-51-21-124-92.eu-n... | 51.21.124.92 | 51.21.124.92 |

A message at the bottom says 'Select an instance'.

24) Now when you go back to beanstalk>environments you will see your environment being launched

25) Click on domain link

Environment successfully launched.

Webapp-env info

Environment overview

Health: Warning

Domain: Webapp-env.eba-ts2acxd.eu-north-1.elasticbeanstalk.com

Application name: webapp

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

Running version: -

Platform state: Supported

Events (11) info

| Time | Type | Details |
|--------------------------------------|------|---|
| January 23, 2024 09:03:21 (UTC+5:30) | WARN | Environment health has transitioned from Pending to Warning. Initialization completed 59 seconds ago and took 2 minutes. Unable to assume role "arn:aws:iam::339715188303:role/aws-elasticbeanstalk-service-role". Verify that the role exists and is configured correctly. |
| January 23, 2024 09:03:08 (UTC+5:30) | INFO | Successfully launched environment: Webapp-env |
| January 23, 2024 09:02:21 (UTC+5:30) | INFO | Added instance [i-0b8e94b7ad106746e] to your environment. |
| January 23, 2024 09:02:02 (UTC+5:30) | INFO | Instance deployment completed successfully. |

The aws service for python application is running

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

- 26) Go back to environment and terminate it
- 27) Also go back to instances and terminate it

Environment successfully launched.

Webapp-env

Events (12) info

| Time | Type | Details |
|--------------------------------------|------|--|
| January 23, 2024 09:03:26 (UTC+5:30) | WARN | Service role "arnaws:iam::339713188308:role/ews-elasticbeanstalk-service-role" is missing permissions required to check for managed updates. Verify the role's policies. |
| January 23, 2024 09:03:21 (UTC+5:30) | WARN | Environment health has transitioned from Pending to Warning. Initialization completed 59 seconds ago and took 2 minutes. Unable to assume role arn:aws:iam::339713188308:role/ews-elasticbeanstalk-service-role because it does not have the required permissions. |

Successfully terminated i-08e94b7ad106746e

Instances (1/1) info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 ... | Elas... |
|------------|--------------------|----------------|---------------|-------------------|---------------|-------------------|--------------------------|-----------------|---------|
| Webapp-env | i-08e94b7ad106746e | Shutting-d... | t3.micro | 2/2 checks passed | View alarms + | eu-north-1c | ec2-51-21-124-92.eu-n... | \$1.21.124.92 | \$1.4 |

For java:

Similary for java

Search results for 'elas'

Services (12)

- Features (27)
- Resources **New**
- Documentation (24,076)
- Knowledge Articles (888)
- Marketplace (1)
- Blogs (2,147)
- Events (23)
- Tutorials (18)

Services

- ElastiCache ☆ In-Memory Cache
- Elastic Transcoder ☆ Easy-to-Use Scalable Media Transcoding
- Elastic Beanstalk ☆ Run and Manage Web Apps
- Elastic Container Service ☆

Create application

Create environment

Enter app name ,and select **platform(java)** > next

Screenshot of the AWS Elastic Beanstalk 'Configure environment' step. The sidebar shows steps 1-6. 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

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.

Environment information [Info](#)
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Screenshot of the AWS Elastic Beanstalk 'Configure environment' step. The sidebar shows steps 1-6. 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'.

Platform

Platform branch

Platform version

Application code [Info](#)

Sample application
 Existing version
 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.

Duplicate and go to IAM

Select roles> create a new role

Screenshot of the AWS IAM search results for 'iam'. The search bar shows 'iam'. The results list includes:

- Services** (11)
 - IAM** Manage access to AWS resources
 - IAM Identity Center** Manage workforce user access to multiple AWS accounts and cloud applications
 - Resource Access Manager** Share AWS resources with other accounts or AWS Organizations
 - AWS App Mesh** Easily monitor and control microservices
- Features** (21)
 - Groups** IAM feature

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access Analyzer

In select. Entrusted entity for use case select EC2

| Policy name | Type | Description |
|-------------------------------|-------------|---|
| AdministratorAccess-AWSE... | AWS managed | Grants account administrative permis... |
| AWSElasticBeanstalkCusto... | AWS managed | Provide the instance in your custom pl... |
| AWSElasticBeanstalkEnhanc... | AWS managed | AWS Elastic Beanstalk Service policy f... |
| AWSElasticBeanstalkManag... | AWS managed | This policy is for the AWS Elastic Bean... |
| AWSElasticBeanstalkMultico... | AWS managed | Provide the instances in your multicon... |
| AWSElasticBeanstalkReadOnly | AWS managed | Grants read-only permissions. Explicitl... |
| AWSElasticBeanstalkRoleCore | AWS managed | AWSElasticBeanstalkRoleCore (Elastic ... |
| AWSElasticBeanstalkRoleCWL | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleECS | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleRDS | AWS managed | (Elastic Beanstalk operations role) Allo... |
| AWSElasticBeanstalkRoleSNS | AWS managed | (Elastic Beanstalk operations role) Allo... |

Select the following for permissions>beanstalk

The screenshot shows the 'Set permissions boundary - optional' step of the IAM role creation wizard. A list of AWS managed policies is displayed, with two specific ones checked: 'AWSElasticBeanstalkMultico...' and 'AWSElasticBeanstalkWebTier'. Both policies are described as providing instances for their respective services.

Enter role name

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. The 'Role name' field contains 'java'. The 'Description' field contains 'Allows EC2 instances to call AWS services on your behalf.' Below this, the 'Step 1: Select trusted entities' section shows a trust policy with the following JSON:

```
1- {  
2-   "Version": "2012-10-17",  
3-   "Statement": [  
4-     {  
5-       "Effect": "Allow",  
6-       "Action": [
```

The screenshot shows the IAM Roles page. A green banner at the top indicates 'Role anthea11 created.' The main table lists five roles, including 'anthea11', which was just created. The table columns include 'Role name', 'Trusted entities', and 'Last activity'. The 'anthea11' role is associated with the 'ec2' service and was created 6 days ago.

Go back to original page and refresh and the new created roles

The screenshot shows the 'Configure service access' step of the AWS Elastic Beanstalk environment creation wizard. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Configure environment', Step 2 is 'Configure service access' (which is currently selected), Step 3 is 'optional Set up networking, database, and tags', Step 4 is 'optional Configure instance traffic and scaling', Step 5 is 'optional Configure updates, monitoring, and logging', and Step 6 is 'Review'. The main content area is titled 'Service access' and contains the following fields:

- Service role:** A radio button group where 'Use an existing service role' is selected. Below it, a dropdown menu shows 'anthea11'.
- Existing service roles:** A dropdown menu showing 'anthea11'.
- EC2 key pair:** A dropdown menu showing 'Choose a key pair'.
- EC2 instance profile:** A dropdown menu showing 'anthea11'.

At the bottom right are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next' (which is highlighted).

The screenshot shows the 'Set up networking, database, and tags - optional' step of the AWS Elastic Beanstalk environment creation wizard. The sidebar shows steps 1 through 6. The main content area is titled 'Virtual Private Cloud (VPC)' and contains:

- VPC:** A dropdown menu showing 'vpc-097e5a4365f5c8ba5 | (172.31.0.0/16)'.
- Instance settings:** A section about creating a subnet in each AZ for instances running your application. It includes a note about avoiding exposure to the Internet and using private subnets and load balancer in public subnets. A checkbox for 'Activated' is present.
- Public IP address:** A section about assigning a public IP address to Amazon EC2 instances. A checkbox for 'Activated' is present.
- Instance subnets:** A table showing subnets across availability zones:

| Availability Zone | Subnet | CIDR | Name |
|---|---------------------|----------------|------|
| eu-north-1b | subnet-02c5b89cd... | 172.31.32.0/20 | |
| <input checked="" type="checkbox"/> eu-north-1c | subnet-0980fd2da... | 172.31.0.0/20 | |
| eu-north-1a | subnet-0c6aa5382... | 172.31.16.0/20 | |
- Database Info:** A section about integrating an RDS SQL database with the environment. It includes a note about choosing subnets for database instances. A checkbox for 'Enable database' is present.
- Database subnets:** A table showing subnets for database instances:

| Availability Zone | Subnet | CIDR | Name |
|---|---------------------|----------------|------|
| eu-north-1b | subnet-02c5b89cd... | 172.31.32.0/20 | |
| <input checked="" type="checkbox"/> eu-north-1c | subnet-0980fd2da... | 172.31.0.0/20 | |
| eu-north-1a | subnet-0c6aa5382... | 172.31.16.0/20 | |

For configure keep the default values

Configure instance traffic and scaling - optional

Instances Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type: (Container default)

Size: 8 GB

IOPS: 100

Throughput: 125 MiB/s

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

Configure updates, monitoring, and logging - optional

Monitoring Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The EnvironmentHealth custom metric is provided free with enhanced health 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 ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming: Activated (standard CloudWatch charges apply.)

Retention

Review Info

Step 1: Configure environment

Environment information

| | |
|------------------------|---|
| Environment tier | Application name |
| Web server environment | anthea |
| Environment name | Application code |
| Anthea-env | Sample application |
| Platform | arn:aws:elasticbeanstalk:eu-north-1:platform/Corretto 21 running on 64bit Amazon Linux 2023/4.2.0 |

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 instance profile |
| arn:aws:iam::339713188308:role/ant | anthea11 |

Your environment is being launched

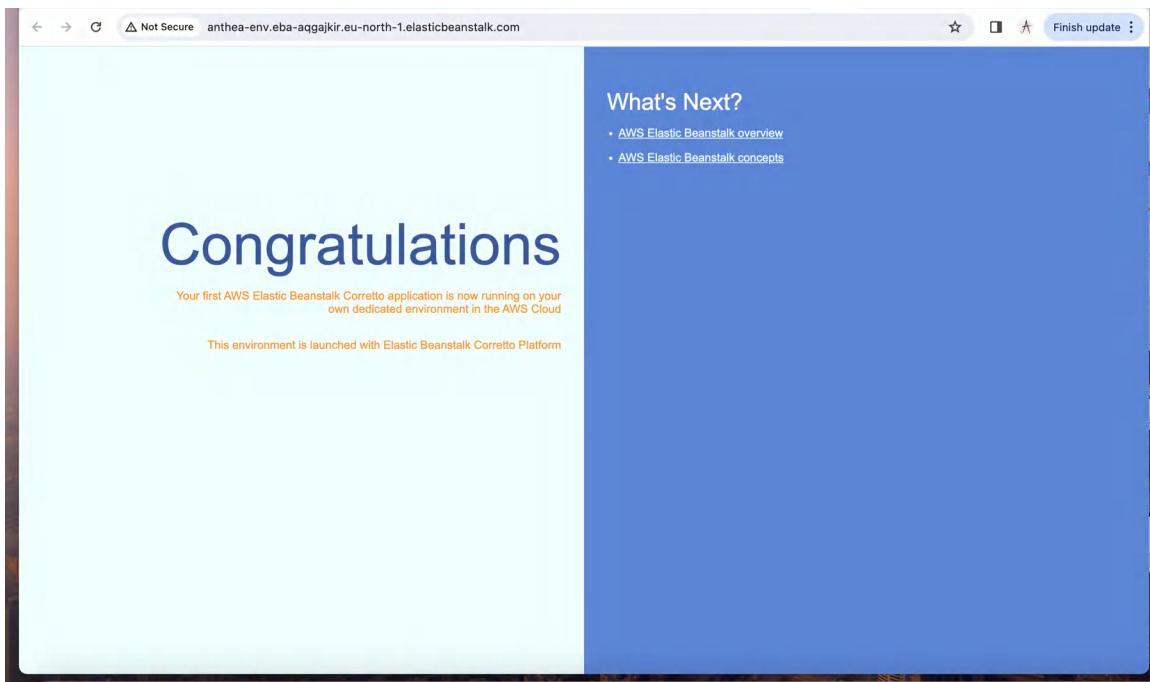
The screenshot shows the AWS Elastic Beanstalk console. At the top, a blue banner says "Elastic Beanstalk is launching your environment. This will take a few minutes." Below this, the "Anthea-env" environment overview is displayed. The "Environment overview" section shows "Health" as Pending and "Domain" as -. The "Platform" section shows "Corretto 21 running on 64bit Amazon Linux 2023/4.2.0" and "Platform state" as Supported. In the bottom right corner of the main window, there is a "Finish update" button. Below the main window, the browser address bar shows "eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances:instanceState=running". A second screenshot below shows the EC2 Instances page with one instance named "Anthea-env" in the "Running" state.

Go back to environment which will now be launched

Click on domain

The screenshot shows the AWS Elastic Beanstalk console. The "Environment overview" section now shows "Health" as Warning - View causes and "Domain" as "Anthea-env.eba-aggajkir.eu-north-1.elasticbeanstalk.com". The "Platform" section remains the same. At the bottom of the main window, there is a "Events" tab with 13 items. The browser address bar shows "eu-north-1.console.aws.amazon.com/elasticbeanstalk/environments/anthea-env/events?region=eu-north-1".

The elastic beanstalk aws is running for java

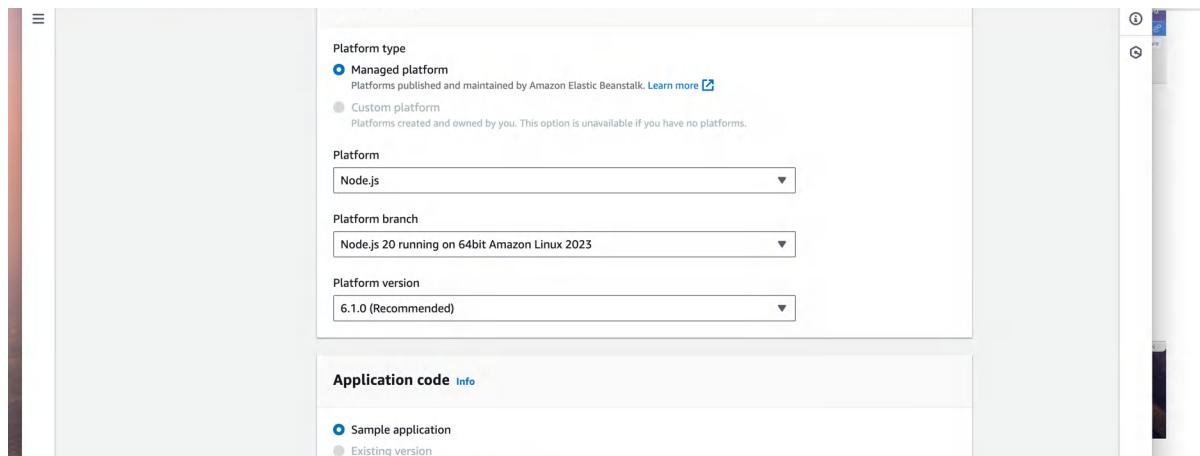


For node.js

Elastic beanstalk > create application > create environment

Two screenshots of the AWS Elastic Beanstalk interface. The top screenshot shows the "Environments" list in the "Elastic Beanstalk" section of the AWS Management Console. It displays one environment named "Anthea-env" (terminated), with columns for Environment name, Health, Application, Platform, Domain, and Status. The bottom screenshot shows the "Configure environment" step of the "Create environment" wizard. The left sidebar lists steps: 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), and Step 6 (Review). The main content area shows the "Environment tier" section (Web server environment selected), "Application information" section (Application name: "riverdale"), and "Environment information" section.

Select platform as node.js



Duplicate >IAM > roles > create a new role

The IAM Dashboard shows the following information:

- Security recommendations:**
 - Add MFA for root user (yellow warning icon)
 - Root user has no active access keys (green success icon)
- AWS Account:** Account ID: 339713188308, Account Alias: Create, Sign-in URL: https://339713188308.signin.aws.amazon.com/console
- IAM resources:** 0 User groups, 0 Users, 5 Roles, 0 Policies, 0 Identity providers
- Quick Links:** My security credentials

The Roles page displays the following details:

- 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.
- Search bar:** Q Search
- Table of roles:**

| Role name | Trusted entities | Last activity |
|---|--|----------------|
| AWSServiceRoleForAutoScaling | AWS Service: autoscaling (Service-Linker) | 26 minutes ago |
| AWSServiceRoleForSupport | AWS Service: support (Service-Linker) | - |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service-Linker) | - |

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.
- 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 Allows 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.

Services Search [Option+S] Global Anthea

| <input type="checkbox"/> | AdministratorAccess-AWSE... | AWS managed | Grants account administrative permis... |
|-------------------------------------|-------------------------------|-------------|--|
| <input type="checkbox"/> | AWSElasticBeanstalkCusto... | AWS managed | Provide the instance in your custom pl... |
| <input type="checkbox"/> | AWSElasticBeanstalkEnhanc... | AWS managed | AWS Elastic Beanstalk Service policy f... |
| <input type="checkbox"/> | AWSElasticBeanstalkManag... | AWS managed | This policy is for the AWS Elastic Bean... |
| <input checked="" type="checkbox"/> | AWSElasticBeanstalkMultico... | 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"/> | AWSElasticBeanstalkRoleCWL | AWS managed | (Elastic Beanstalk operations role) Alloc... |
| <input type="checkbox"/> | AWSElasticBeanstalkRoleECS | AWS managed | (Elastic Beanstalk operations role) Alloc... |
| <input type="checkbox"/> | AWSElasticBeanstalkRoleRDS | AWS managed | (Elastic Beanstalk operations role) Alloc... |
| <input type="checkbox"/> | AWSElasticBeanstalkRoleSNS | AWS managed | (Elastic Beanstalk operations role) Alloc... |
| <input type="checkbox"/> | AWSElasticBeanstalkRoleW... | AWS managed | (Elastic Beanstalk operations role) Alloc... |
| <input checked="" type="checkbox"/> | AWSElasticBeanstalkWebTier | AWS managed | Provide the instances in your web serv... |
| <input checked="" type="checkbox"/> | AWSElasticBeanstalkWorker... | AWS managed | Provide the instances in your worker e... |

▶ Set permissions boundary - optional

Enter the dets of role and create the role

Step 2
[Add permissions](#)

Step 3
Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
river

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.

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-       ]
}

```

Edit

The screenshot shows the AWS Identity and Access Management (IAM) service. In the left sidebar, under 'Access management', the 'Roles' section is selected. The main content area displays a table titled 'Roles (4) Info'. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'. The rows show four roles: 'AWSServiceRoleForAutoScaling' (AWS Service: autoscaling), 'AWSServiceRoleForSupport' (AWS Service: support), 'AWSServiceRoleForTrustedAdvisor' (AWS Service: trustedadvisor), and the newly created role 'river' (AWS Service: ec2). A green banner at the top of the page says 'Role river created.'

Go back to original and refresh and select the created role

This screenshot shows the 'Configure service access' step of the Elastic Beanstalk environment creation wizard. On the left, a sidebar lists steps: 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), and Step 6 (Review). The main content area is titled 'Configure service access' and contains sections for 'Service access', 'Service role', 'Existing service roles', 'EC2 key pair', and 'EC2 instance profile'. Under 'Service role', the 'Use an existing service role' radio button is selected, and 'river' is chosen from the dropdown. Under 'EC2 instance profile', 'river' is also chosen from the dropdown. At the bottom, there are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

This screenshot shows the 'Set up networking, database, and tags - optional' step of the Elastic Beanstalk environment creation wizard. The sidebar shows steps 1 through 5. The main content area is titled 'Set up networking, database, and tags - optional' and includes sections for 'Virtual Private Cloud (VPC)', 'Instance settings', and 'Public IP address'. Under 'Virtual Private Cloud (VPC)', it shows a VPC selection dropdown with 'vpc-097e5a4365f5c8ba5 | (172.31.0.0/16)'. Under 'Instance settings', it says 'Choose a subnet in each AZ for the instances that run your application.' Under 'Public IP address', there is a checkbox labeled 'Activated'. At the bottom, there is a 'Filter instance subnets' search bar.

Configure instance traffic and scaling - optional

Instances Info
Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type (Container default)

Size The number of gigabytes of the root volume attached to each instance.
8 GB

IOPS Input/output operations per second for a provisioned IOPS (SSD) volume.
100 IOPS

Throughput The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance
125 MiB/s

Amazon CloudWatch monitoring The time interval between when metrics are reported from the EC2 instances

Monitoring interval

Configure updates, monitoring, and logging - optional

Monitoring Info

Health reporting Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health 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 ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming
 Activated (standard CloudWatch charges apply)

Retention

Review Info

Step 1: Configure environment

Environment information

Environment tier Web server environment
Application name riverdale

Environment name Riverdale-env
Application code Sample application

Platform amazonelasticbeanstalk:eu-north-1:platform/Node.js/20 running on 64bit Amazon Linux 2023/6.1.0

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 instance profile**

Environment is being launched

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

Riverdale-env Info

| Environment overview | | Platform |
|----------------------|------------------|----------------|
| Health | Environment ID | Change version |
| Unknown | e-pnfierrpimu | |
| Domain | Application name | |
| - | riverdale | |

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

Running version
-

Platform state
Supported

The screenshot shows the AWS Elastic Beanstalk console. On the left, there's a sidebar with 'Elastic Beanstalk' at the top, followed by 'Applications', 'Environments', and 'Change history'. Below that, under 'Application: riverdale', are 'Application versions', 'Saved configurations', and 'Environment: Riverdale-env' which is expanded to show 'Go to environment', 'Configuration', 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The main area is titled 'Riverdale-env Info' and contains sections for 'Environment overview' and 'Platform'. In 'Environment overview', it shows 'Health' (with a warning icon), 'Domain' (Riverdale-env.eba-47dxxfup.eu-north-1.elasticbeanstalk.com), and 'Application name' (riverdale). In 'Platform', it shows 'Platform' (Node.js 20 running on 64bit Amazon Linux 2023/6.1.0), 'Running version' (empty), and 'Platform state' (Supported). At the bottom, there are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags', with 'Events (10) Info' currently selected. A large orange button at the top right says 'Upload and deploy'.

The elastic beanstalk node js is now running

The screenshot shows a web browser window with the URL 'Not Secure riverdale-env.eba-47dxxfup.eu-north-1.elasticbeanstalk.com'. The page has a green header with the text 'Congratulations' and a message: 'Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud'. Below this, it says 'This environment is launched with Elastic Beanstalk Node.js Platform'. To the right, there's a sidebar titled 'What's Next?' with a list of links: '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', and 'Working with Logs'. At the top right of the browser window, there are buttons for 'Finish update' and 'Share'.

