

EXPERIMENT 4

AIM: To study and implement Platform as a Service using AWS Elastic Beanstalk / Microsoft Azure App Service

CO2: Analyze various cloud computing service models and implement them to solve the given problems

THEORY :

Q]What is AWS?

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon. It includes a mixture of infrastructure-as-a-service ([IaaS](#)), platform-as-a-service ([PaaS](#)) and packaged software-as-a-service ([SaaS](#)) offerings. AWS offers tools such as compute power, database storage and content delivery services.

Q]What is Platform as a Service?

Platform as a Service, also known as PaaS, is a [type of cloud computing service model](#) that offers a flexible, scalable cloud platform to develop, deploy, run, and manage apps. PaaS provides everything developers need for application development without the headaches of updating the operating system and development tools or maintaining hardware. Instead, the entire PaaS environment—or platform—is delivered by a third-party service provider via the cloud.

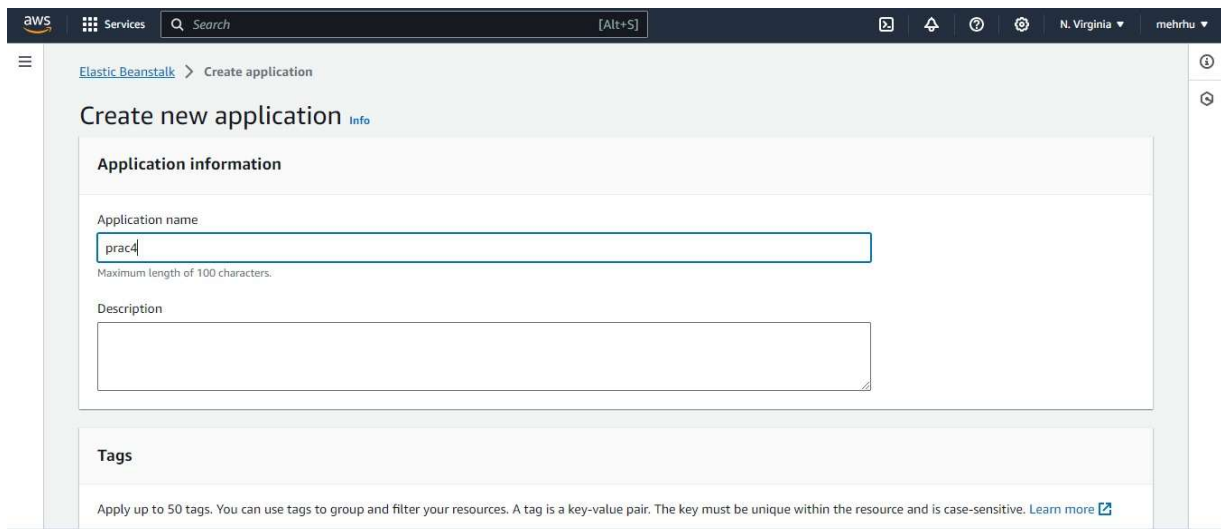
PaaS helps businesses avoid the hassle and cost of installing hardware or software to develop or host new custom applications. Development teams simply purchase pay-as-you-go access to everything they need to build custom apps, including infrastructure, development tools, operating systems, and more.

Q]What is AWS Elastic Beanstalk?

AWS Elastic Beanstalk is an AWS-managed service for web applications. Elastic Beanstalk is a pre-configured [EC2](#) server that can directly take up your application code and environment configurations and use it to automatically provision and deploy the required resources within AWS to run the web application. Unlike EC2 which is Infrastructure as a service, Elastic Beanstalk is a Platform As A Service (PAAS) as it allows users to directly use a pre-configured server for their application.

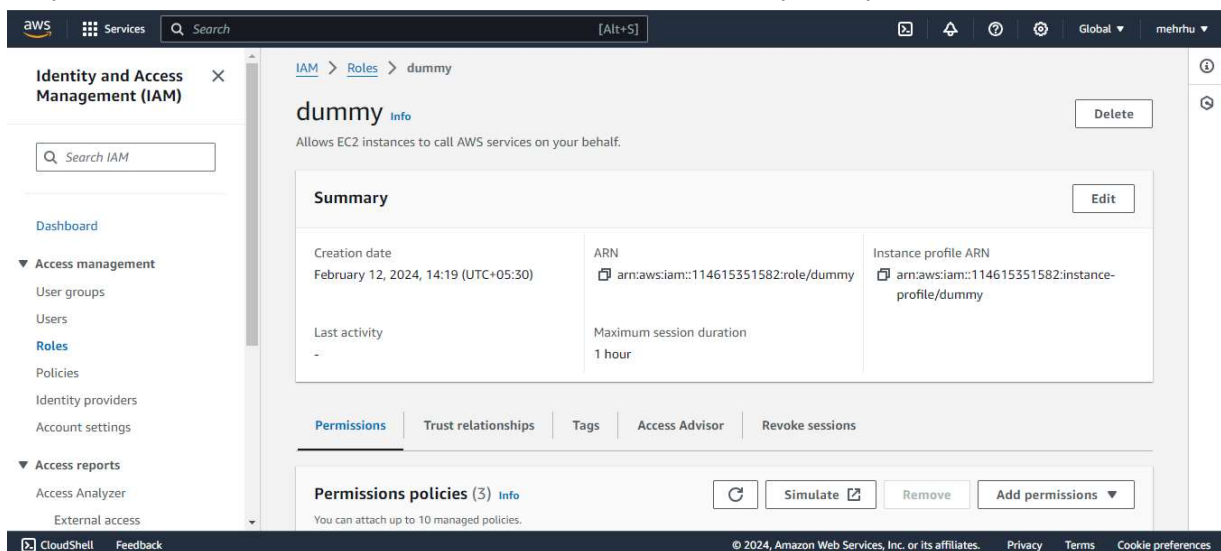
Output:

Step1: Open ElasticBeanstalk Dashboard and click on create new application.



The screenshot shows the 'Create new application' page in the AWS Elastic Beanstalk console. The page has a dark header with the AWS logo, 'Services' menu, a search bar, and navigation icons. The main content area is titled 'Create new application' with an 'Info' link. Below this is the 'Application information' section, which contains a text input for 'Application name' (containing 'prac4') and a larger text area for 'Description'. A note below the name field states 'Maximum length of 100 characters.' Below the description field is a 'Tags' section with a note: 'Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)'.

Step 2: Create a service role in IAM console and add required permissions.



The screenshot shows the AWS IAM console 'dummy' role page. The left sidebar shows the 'Identity and Access Management (IAM)' menu with options like 'Dashboard', 'Access management', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Access reports', 'Access Analyzer', and 'External access'. The main content area is titled 'dummy' with an 'Info' link and a 'Delete' button. Below the title is a description: 'Allows EC2 instances to call AWS services on your behalf.' The 'Summary' section contains a table with the following data:

Summary		
Creation date	ARN	Instance profile ARN
February 12, 2024, 14:19 (UTC+05:30)	arn:aws:iam::114615351582:role/dummy	arn:aws:iam::114615351582:instance-profile/dummy
Last activity	Maximum session duration	
-	1 hour	

Below the summary is a 'Permissions' tab with buttons for 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'. At the bottom, there is a 'Permissions policies (3)' section with a note 'You can attach up to 10 managed policies.' and buttons for 'Simulate', 'Remove', and 'Add permissions'.

Step 3: Use this role to add to the already existing role in COnfigure service access.

aws

Services

Search

[Alt+S]

N. Virginia

mehru

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 service access

Info

Service access

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

Service role

☐ Create and use new service role

☒ Use an existing service role

Existing service roles

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

dummy

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.

Step 4: Click on skip to review and then on next button.

aws

Services

Search

[Alt+S]

N. Virginia

mehru

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

Review

Info

Step 1: Configure environment

Edit

Environment information

Environment tier

Web server environment

Application name

prac4

Environment name

Prac4-env

Application code

Sample application

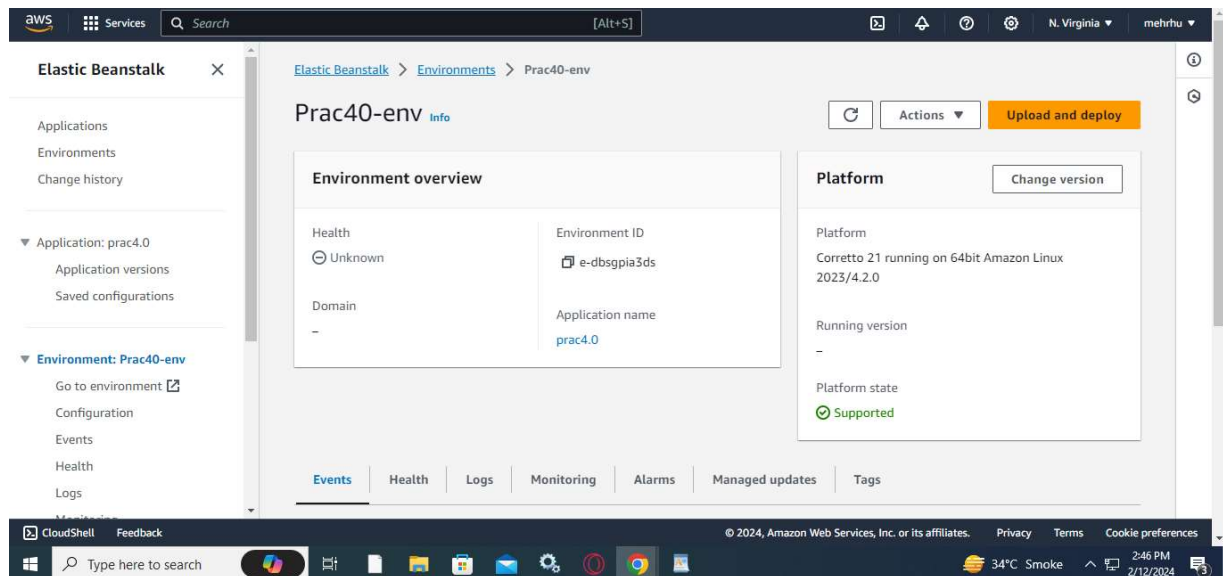
Platform

arn:aws:elasticbeanstalk:us-east-1:platform/Corretto 21 running on 64bit Amazon Linux 2023/4.2.0

Step 2: Configure service access

Edit

Step 5: Click on submit and wait for your application environment to step and then upload java zip file of you application and deploy your application.



Conclusion: We have successfully implemented Platform as a Service using AWS ElasticBeanstalk for java application.