

Project 2: Website Orchestration

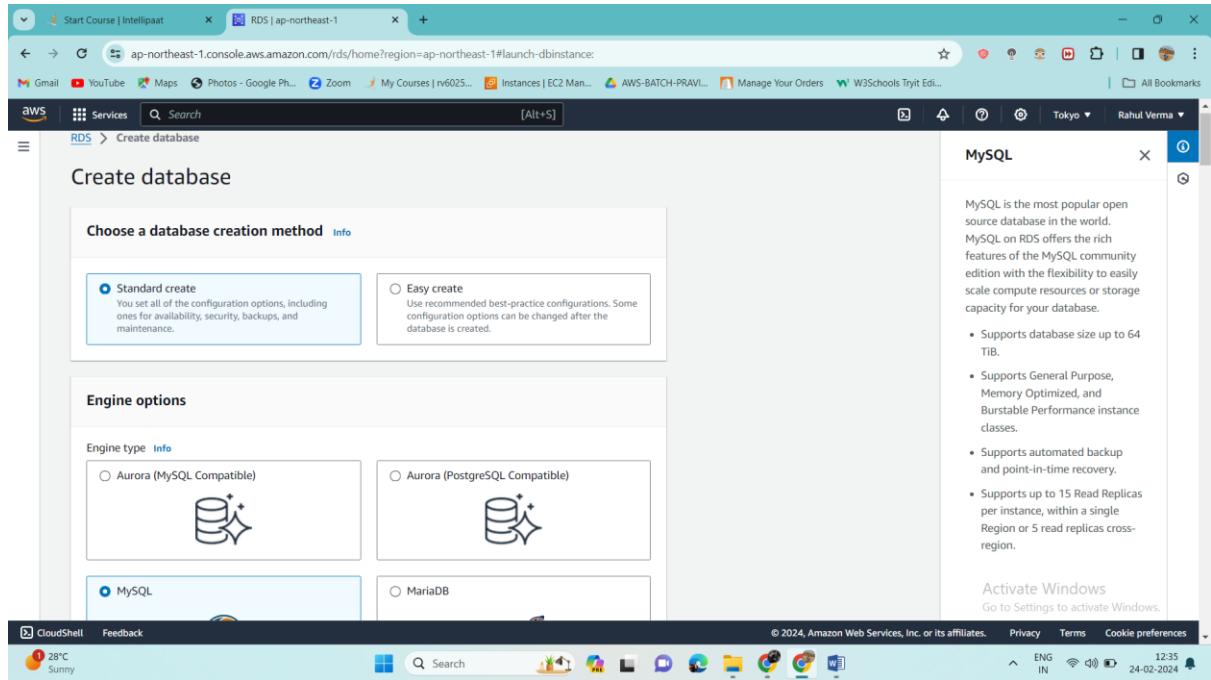
Industry: Internet related

Problem Statement: How to orchestrate a website with lesser time and higher availability along with Auto Scaling.

Topics: In this AWS project, you have to deploy a high-availability PHP application with an external Amazon RDS database to Elastic Beanstalk. Running a DB instance external to Elastic Beanstalk decouples the database from the lifecycle of your environment. This lets you connect to the same database from multiple environments, swap one database for another, or perform a blue/green deployment without affecting your database.

Highlights: Launch a DB instance in Amazon RDS Create an Elastic Beanstalk Environment Configure Security Groups and Scaling

First we will create one RDS-



Choose a database creation method [Info](#)

Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible) 

Aurora (PostgreSQL Compatible) 

MySQL 

MariaDB 

MySQL

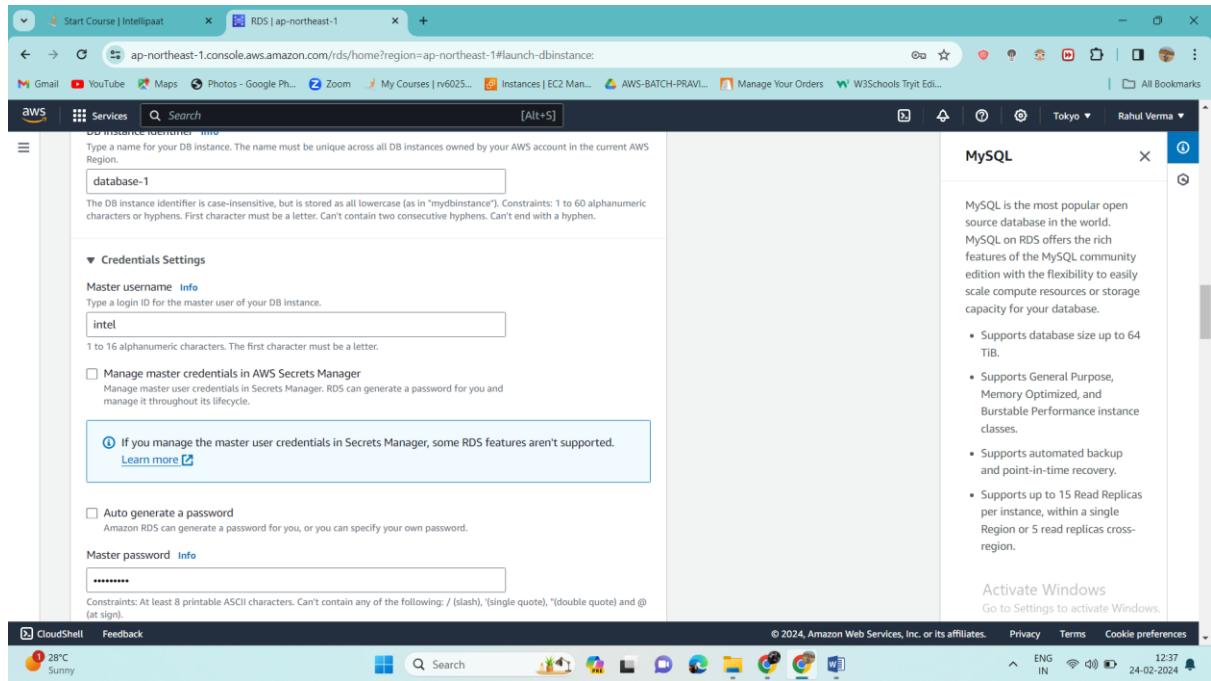
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

username intel and password- intel123



Database identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

intel

1 to 16 alphanumeric characters. The first character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. [Learn more](#)

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

MySQL

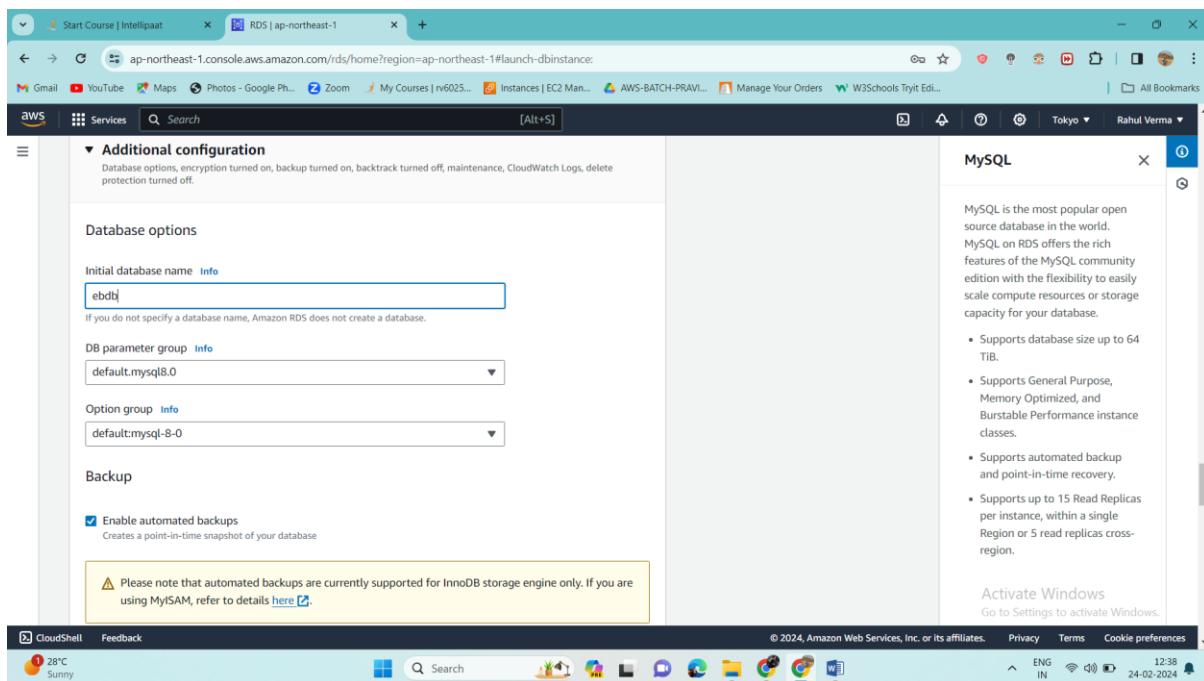
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

Database name ebdb



Initial database name [Info](#)
ebdb

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)
default.mysql8.0

Option group [Info](#)
default:mysql-8-0

Backup

Enable automated backups
Creates a point-in-time snapshot of your database

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

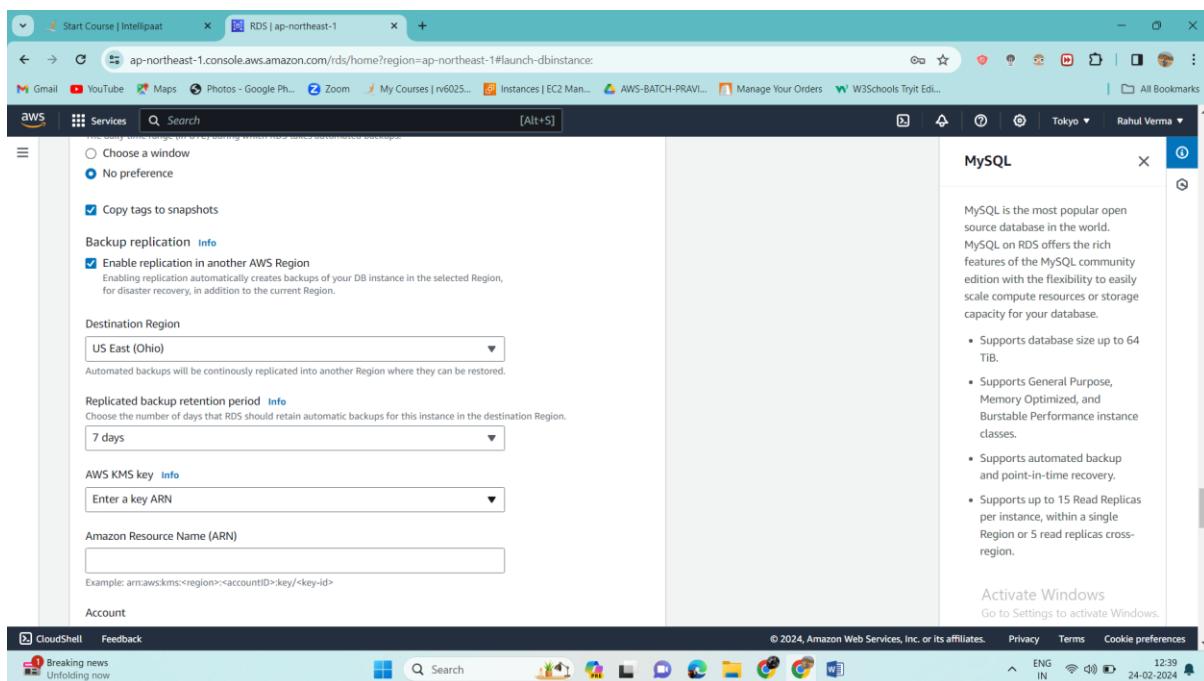
MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Activate Windows
Go to Settings to activate Windows.

Replication is enabled



Choose a window
 No preference

Copy tags to snapshots

Backup replication [Info](#)
 Enable replication in another AWS Region
Enabling replication automatically creates backups of your DB instance in the selected Region, for disaster recovery, in addition to the current Region.

Destination Region
US East (Ohio)

Automated backups will be continuously replicated into another Region where they can be restored.

Replicated backup retention period [Info](#)
Choose the number of days that RDS should retain automatic backups for this instance in the destination Region.
7 days

AWS KMS key [Info](#)
Enter a key ARN

Amazon Resource Name (ARN)
Example: arnaws:kms:<region>:<accountID>/key/<key-id>

Account

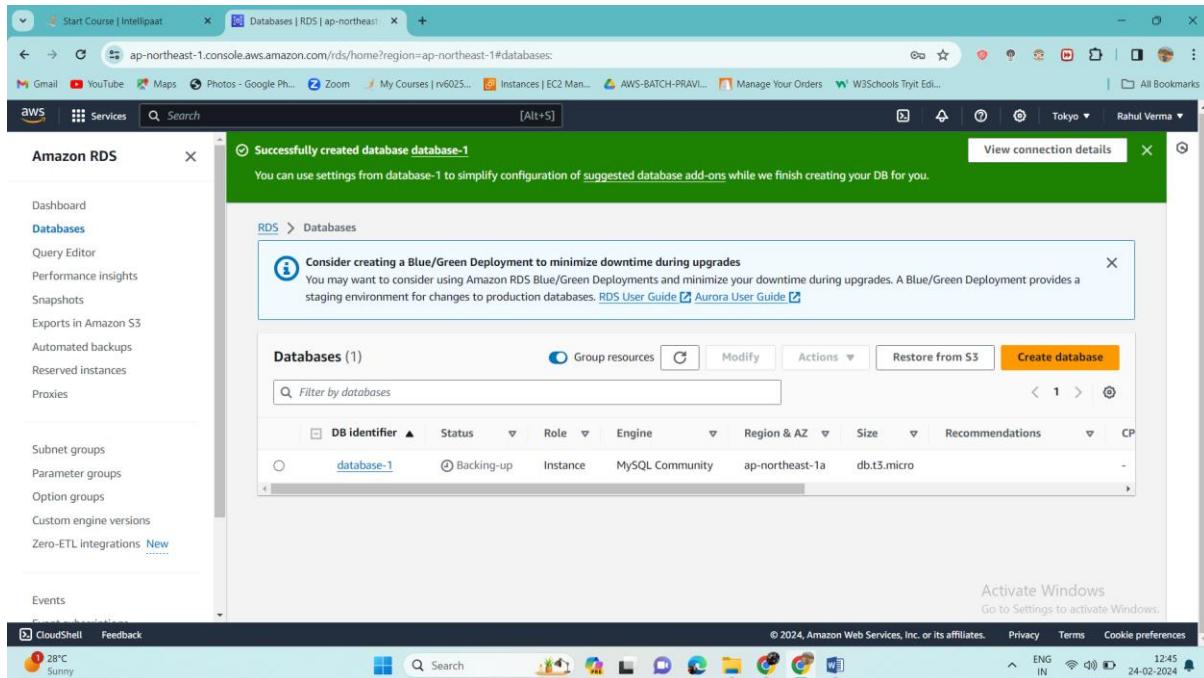
MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

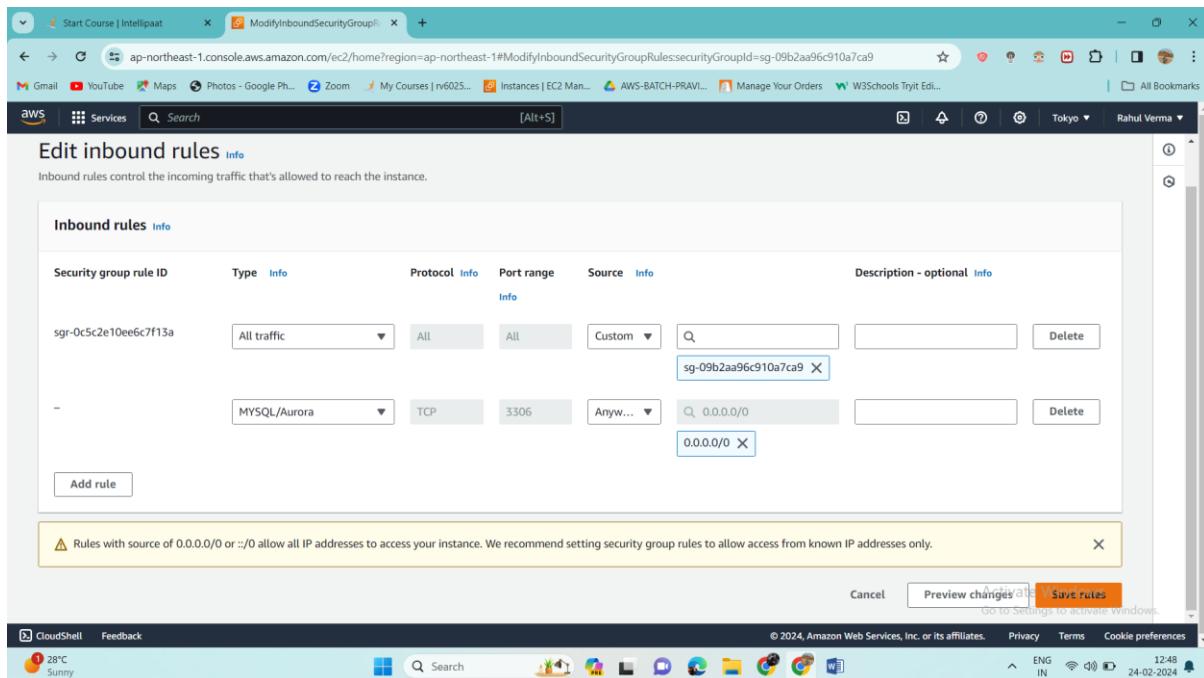
Activate Windows
Go to Settings to activate Windows.

Rest default settings and now create your RDS



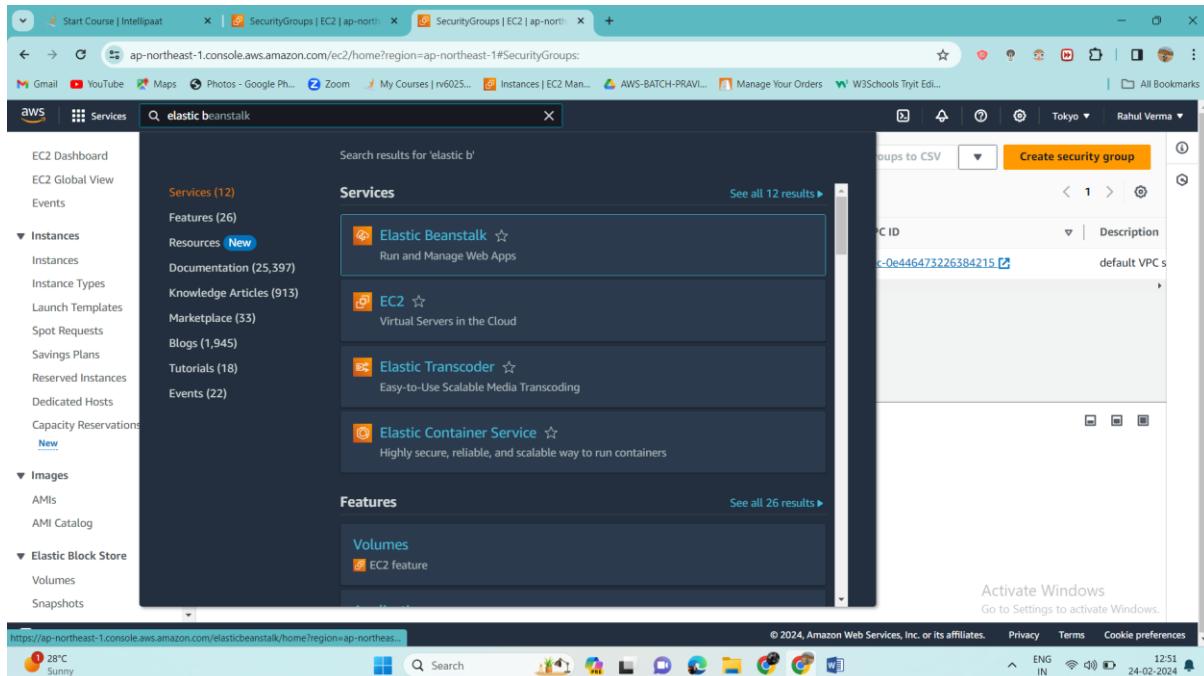
The screenshot shows the AWS RDS console with a success message: "Successfully created database database-1". It includes a note about using settings from the database to simplify configuration of suggested database add-ons. The main table lists one database entry: "database-1" (Backing-up, Instance, MySQL Community, ap-northeast-1a, db.t3.micro). The sidebar shows various RDS management options like Dashboard, Databases, and Query Editor.

Now we will configure RDS security group



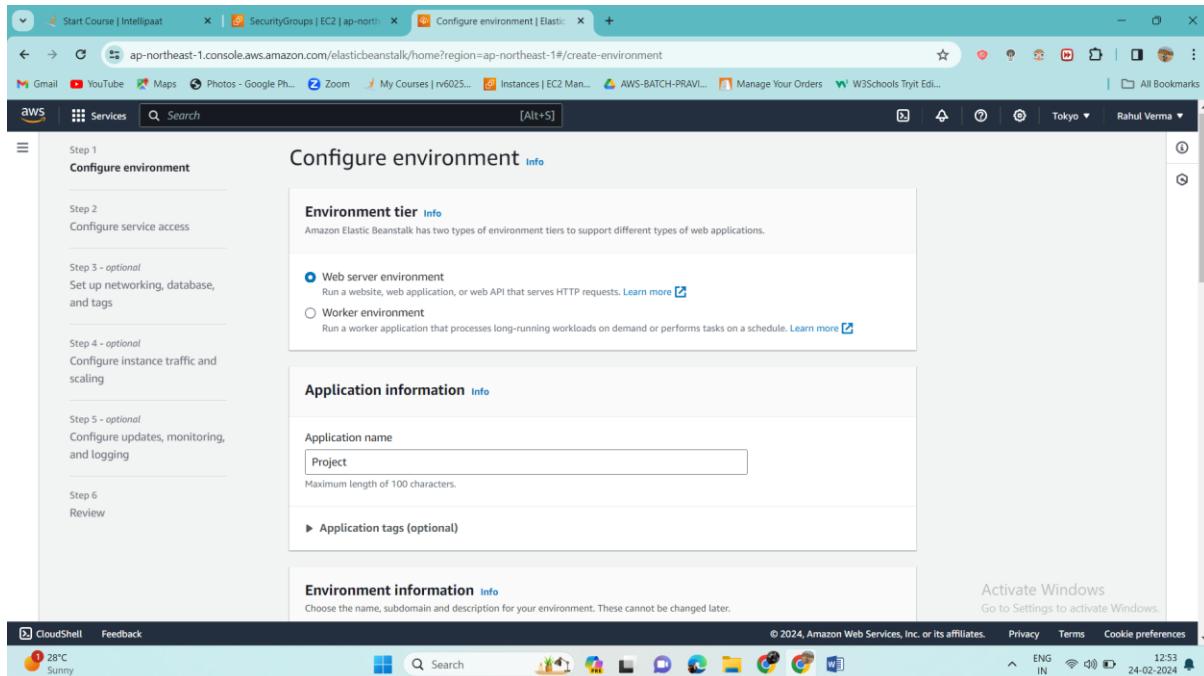
The screenshot shows the "Edit inbound rules" interface for an EC2 security group. It lists two rules: one for "All traffic" (Protocol: All, Port range: All, Source: Custom, Source IP: sg-09b2aa96c910a7ca9) and another for "MySQL/Aurora" (Protocol: TCP, Port range: 3306, Source: Anywhere, Source IP: 0.0.0.0/0). A warning message at the bottom states: "⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." The bottom right shows "Preview changes" and "Save rules" buttons.

Now we will create elastic beans stalk environment



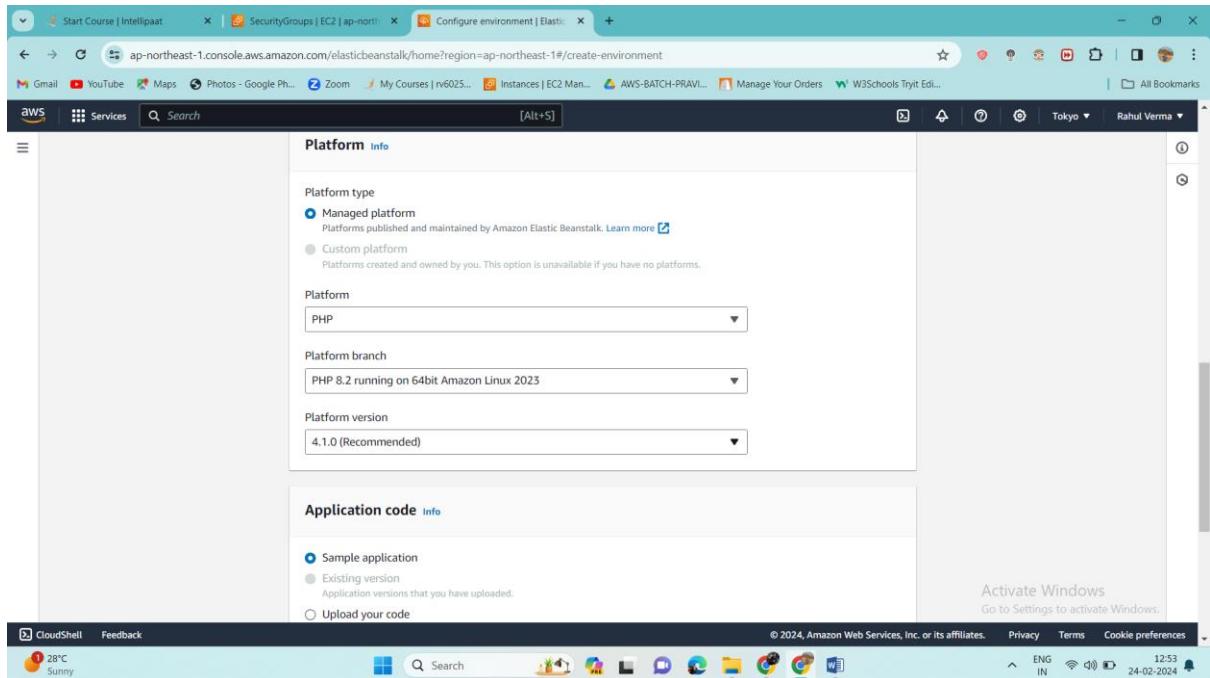
The screenshot shows the AWS Management Console search results for 'elastic beanstalk'. The search bar at the top contains the query 'elastic beanstalk'. The results are categorized into 'Services' and 'Features'. Under 'Services', 'Elastic Beanstalk' is listed as 'Run and Manage Web Apps'. Under 'Features', 'EC2' is listed as 'Virtual Servers in the Cloud', and 'Elastic Container Service' is listed as 'Highly secure, reliable, and scalable way to run containers'. The left sidebar shows navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store.

Will start configuring our environment

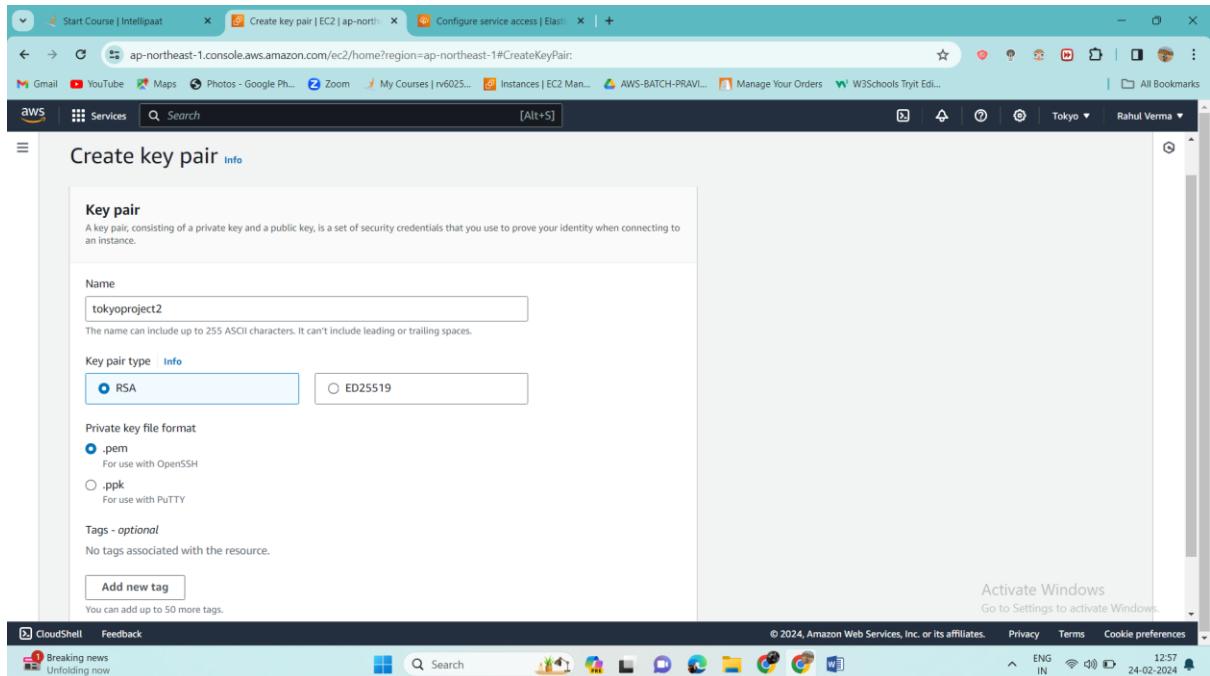


The screenshot shows the 'Configure environment' wizard for Elastic Beanstalk. The left sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled 'Configure environment' and contains three sections: 'Environment tier' (selected 'Web server environment'), 'Application information' (Application name: 'Project'), and 'Environment information' (Choose the name, subdomain and description for your environment. These cannot be changed later). The bottom right corner shows a message: 'Activate Windows' and 'Go to Settings to activate Windows.'

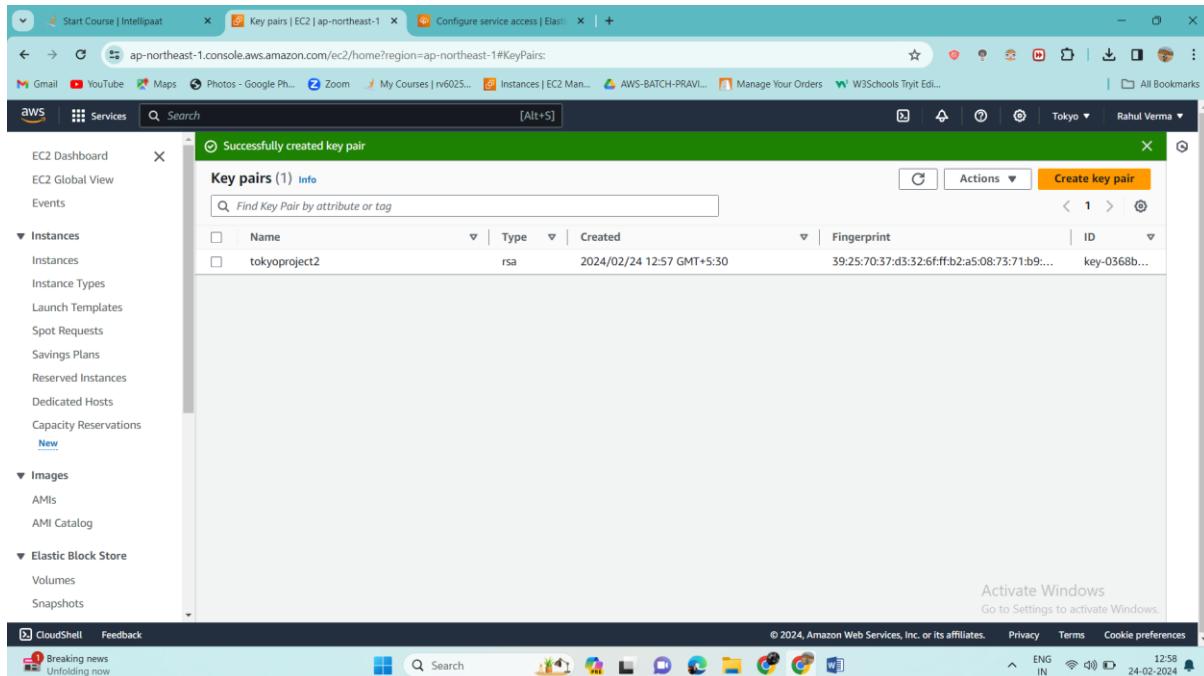
Will select our platform



We will create one key pair

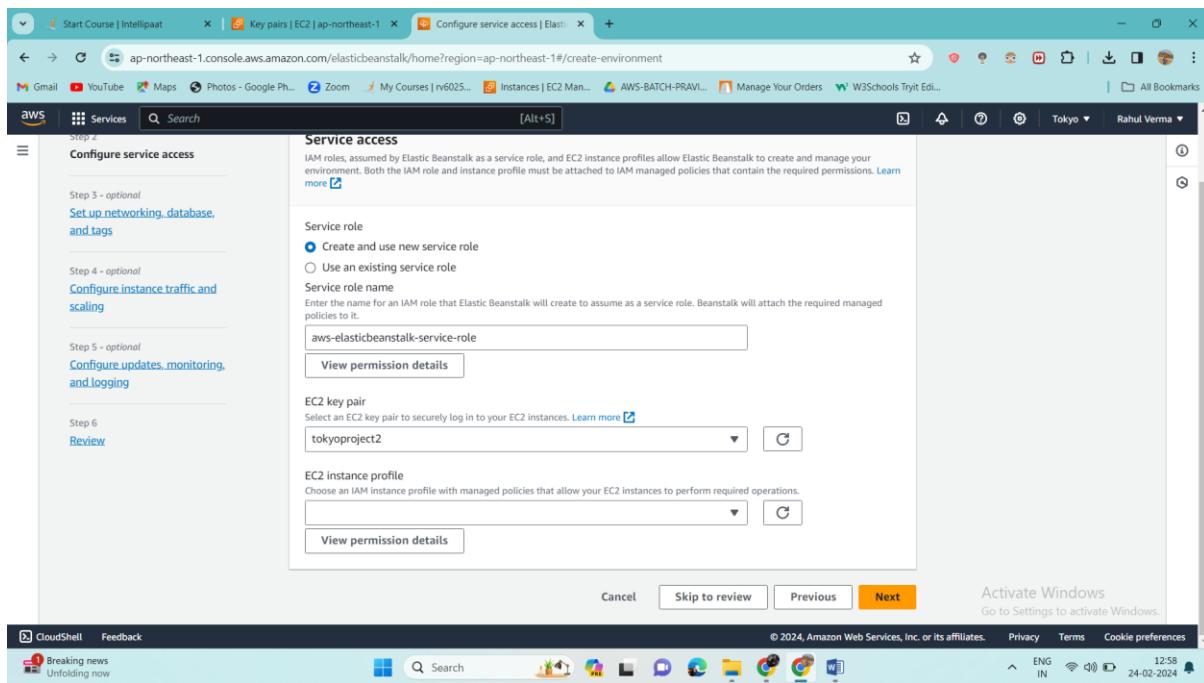


Key pair created successfully



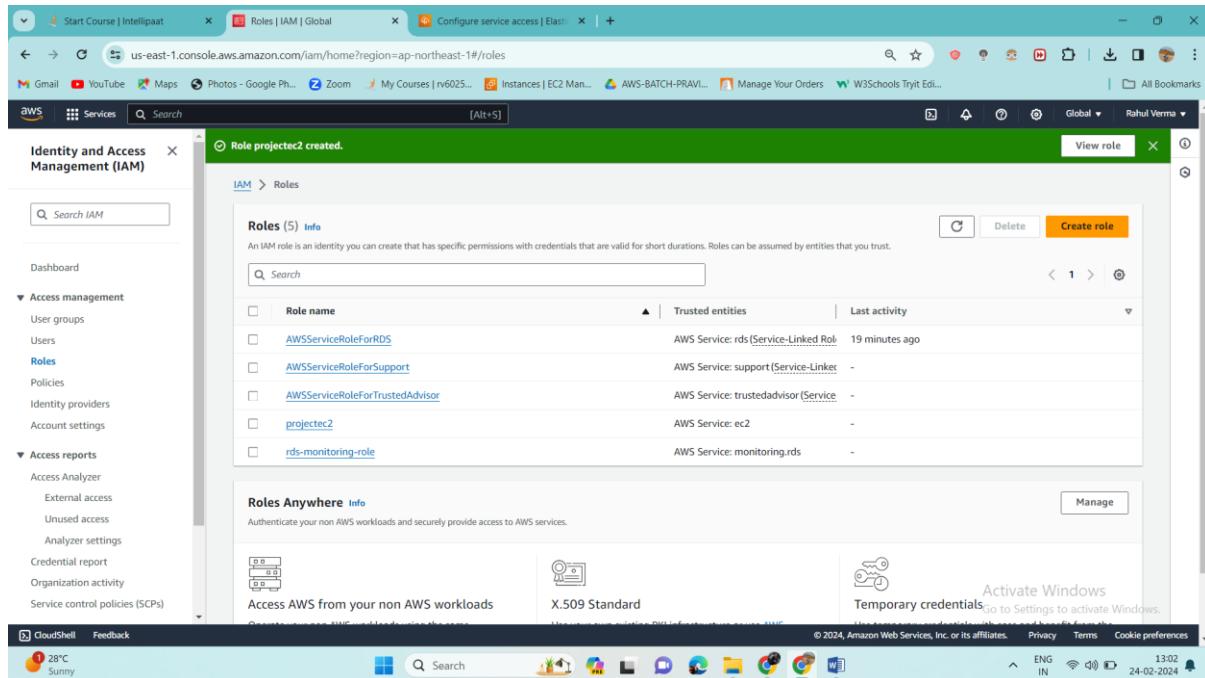
The screenshot shows the AWS EC2 Key Pairs page. A green banner at the top says "Successfully created key pair". Below it, a table lists the key pair "tokyoproject2" with details: Name (tokyoproject2), Type (rsa), Created (2024/02/24 12:57 GMT+5:30), Fingerprint (39:25:70:37:d5:32:6f:ff:b2:a5:08:73:71:b9:...), and ID (key-0368b...). The left sidebar shows navigation options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The bottom status bar shows the date as 24-02-2024.

Now we will click it on refresh button then our key pair will show in drop down menu then will select it



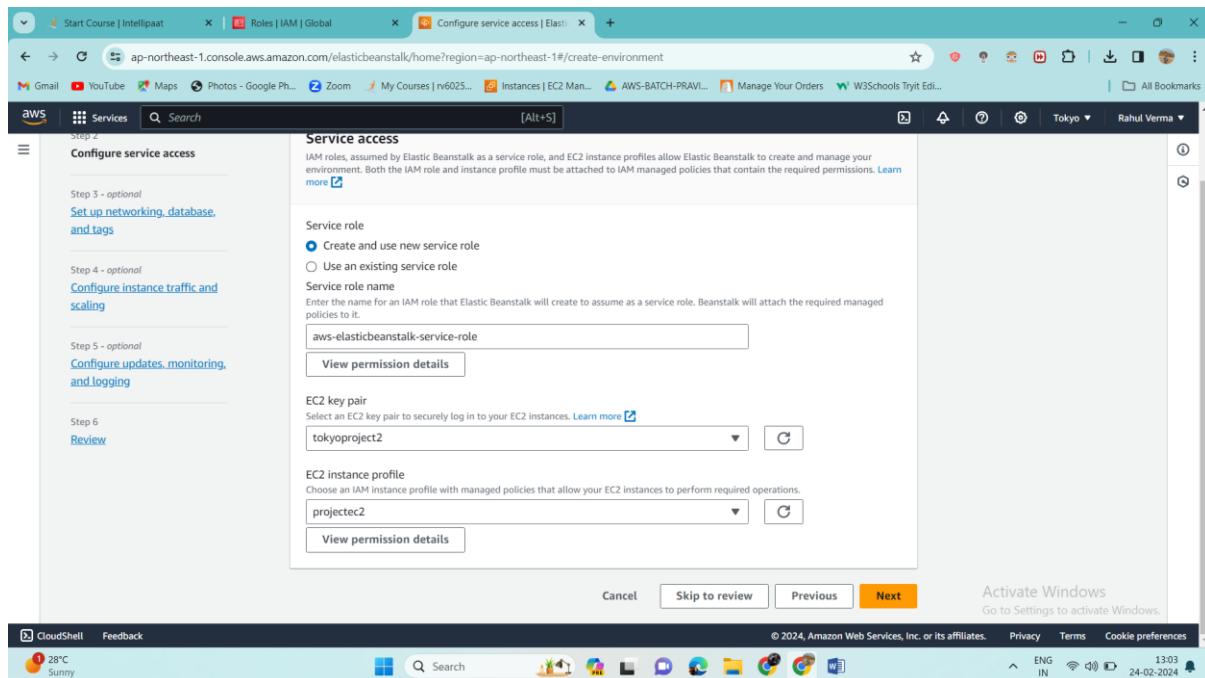
The screenshot shows the "Configure service access" step of the Elastic Beanstalk setup. On the left, a sidebar lists steps: Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main area shows "Service access" configuration. Under "EC2 key pair", a dropdown menu is open, showing "tokyoproject2" as the selected option. Other options in the dropdown include "Create new key pair" and "Select key pair". The bottom status bar shows the date as 24-02-2024.

We have created one IAM role- projectec2



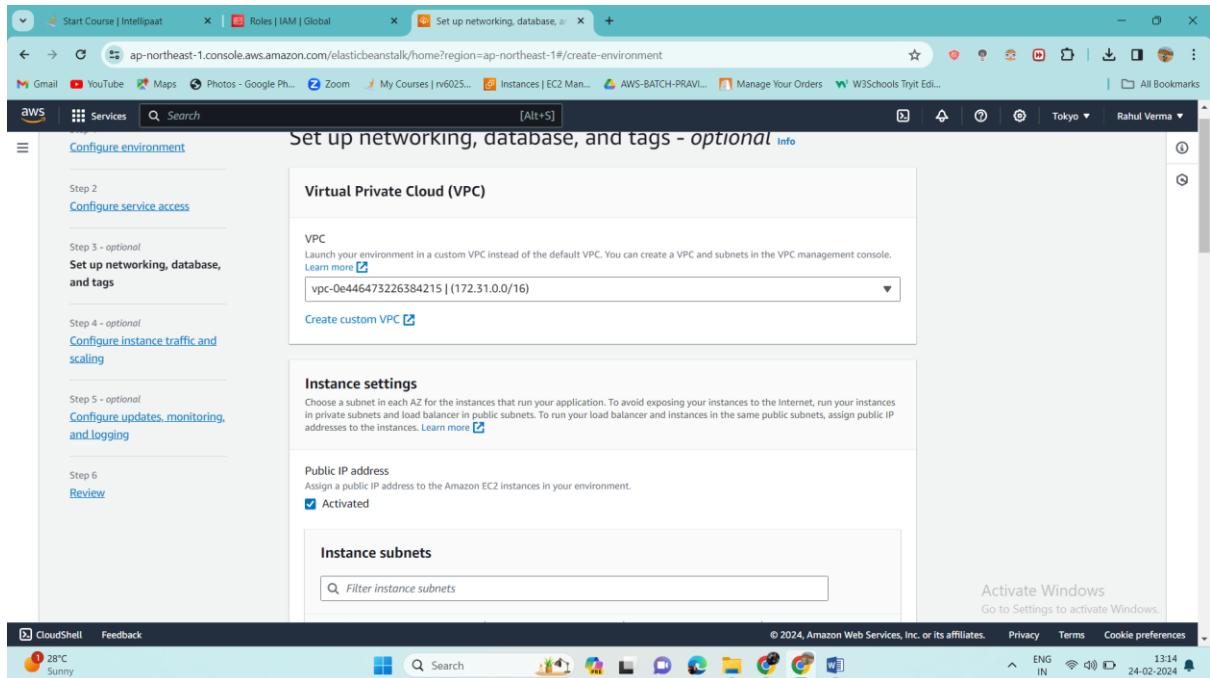
The screenshot shows the AWS IAM Roles page. The left sidebar is collapsed. The main content area shows a table of roles. The 'projectec2' role is listed in the second row. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'. The 'projectec2' role is associated with 'AWS Service: ec2' and was last active 19 minutes ago. There are buttons for 'View role' and 'Delete' in the top right of the table area. Below the table, there is a section titled 'Roles Anywhere' with a 'Manage' button.

Now we will use that in our environment



The screenshot shows the 'Configure service access' step of the Elastic Beanstalk environment creation wizard. The left sidebar shows steps 3 through 6. 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. It requires attaching IAM managed policies. There are sections for 'Service role' (radio buttons for 'Create and use new service role' and 'Use an existing service role', with the former selected), 'Service role name' (input field containing 'aws-elasticbeanstalk-service-role'), 'View permission details' (button), 'EC2 key pair' (input field containing 'tokyoproject2'), 'View permission details' (button), and 'EC2 instance profile' (input field containing 'projectec2'), with a 'View permission details' button below it. At the bottom are 'Cancel', 'Skip to review', 'Previous', 'Next' buttons, and an 'Activate Windows' link.

Default vpc with public access



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-0e446473226384215 | (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

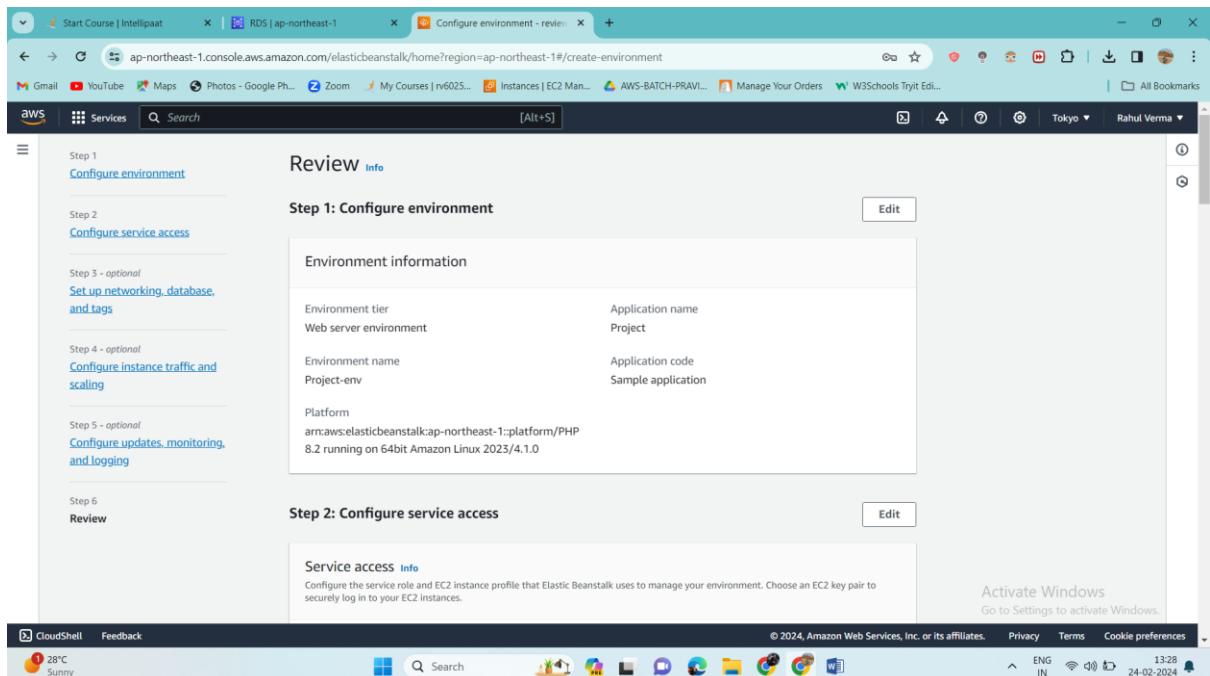
Instance subnets

Filter instance subnets

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback 28°C Sunny Search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 13:14 24-02-2024

Now we will review our environment



Configure environment - [Review](#) [Info](#)

Step 1: Configure environment

Environment information

Environment tier	Application name
Web server environment	Project
Environment name	Application code
Project-env	Sample application

Platform
arn:aws:elasticbeanstalk:ap-northeast-1:platform/PHP 8.2 running on 64bit Amazon Linux 2023/4.1.0

Step 2: Configure service access

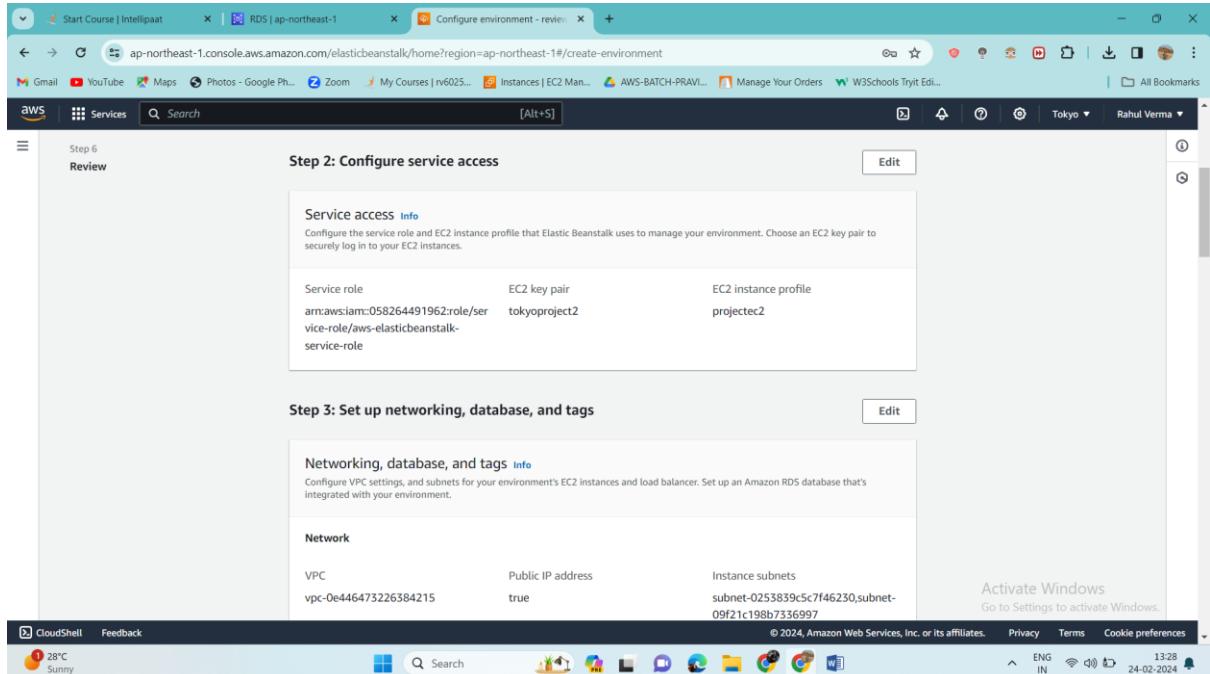
Service access [Info](#)

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.

Activate Windows
Go to Settings to activate Windows.

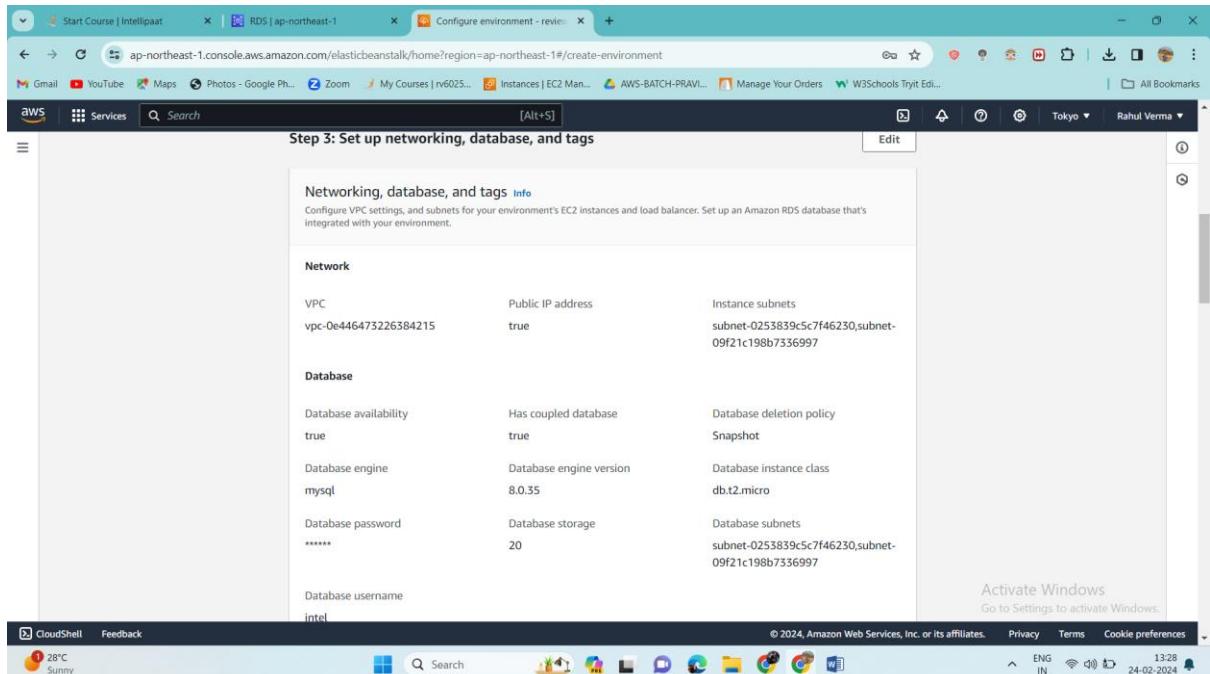
CloudShell Feedback 28°C Sunny Search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 13:28 24-02-2024

2



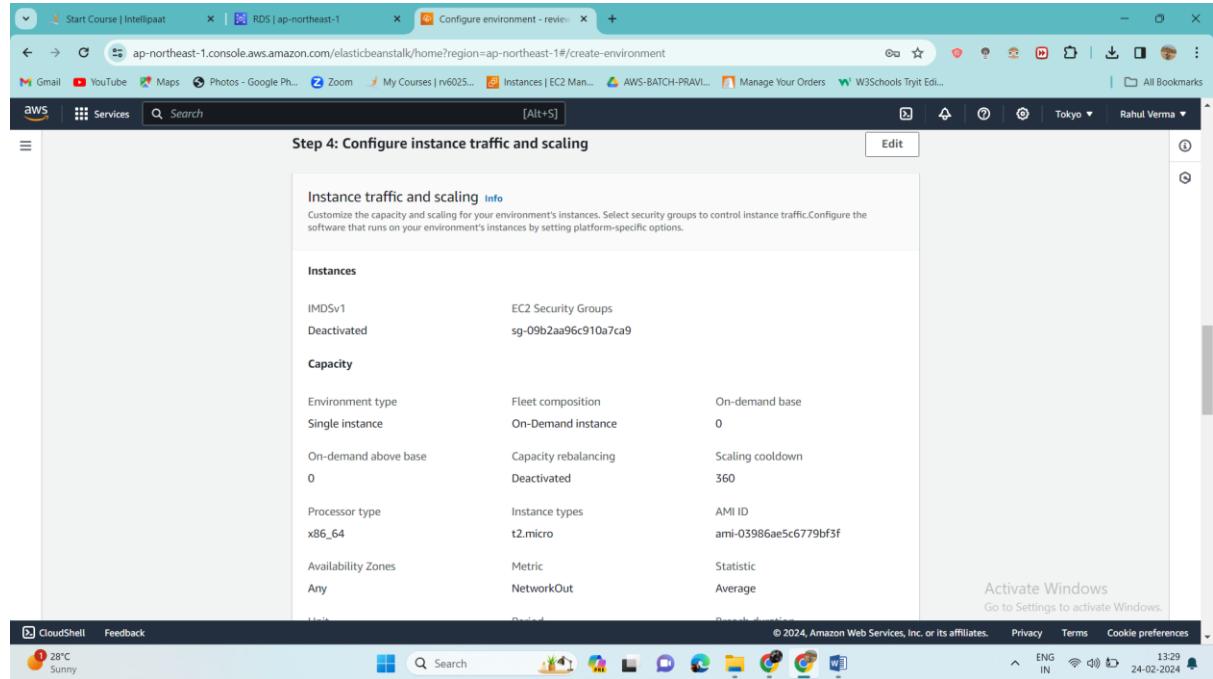
The screenshot shows the 'Configure environment - review' step of the AWS Elastic Beanstalk environment creation process. The 'Step 2: Configure service access' section is active, showing the configuration of service roles and EC2 instance profiles. The 'Service access' info panel indicates that the service role 'arn:aws:iam::058264491962:role/service-role/aws-elasticbeanstalk-service-role' is paired with the EC2 key pair 'tokyoproject2'. The 'EC2 instance profile' is set to 'projectec2'. The 'Step 3: Set up networking, database, and tags' section is also visible, showing the configuration of the VPC, Public IP address, and Instance subnets. The AWS navigation bar and browser interface are visible at the top and bottom of the screen.

3



The screenshot shows the 'Configure environment - review' step of the AWS Elastic Beanstalk environment creation process. The 'Step 3: Set up networking, database, and tags' section is active, showing the configuration of the VPC, Public IP address, and Instance subnets. The 'Networking, database, and tags' info panel indicates that the VPC 'vpc-0e446473226384215' has a Public IP address 'true' and Instance subnets 'subnet-0253839c5cf46230, subnet-09f21c198b7336997'. The 'Database' section shows the configuration of the database availability, engine, password, storage, and username. The AWS navigation bar and browser interface are visible at the top and bottom of the screen.

4



Step 4: Configure instance traffic and scaling

Instance traffic and scaling Info
Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

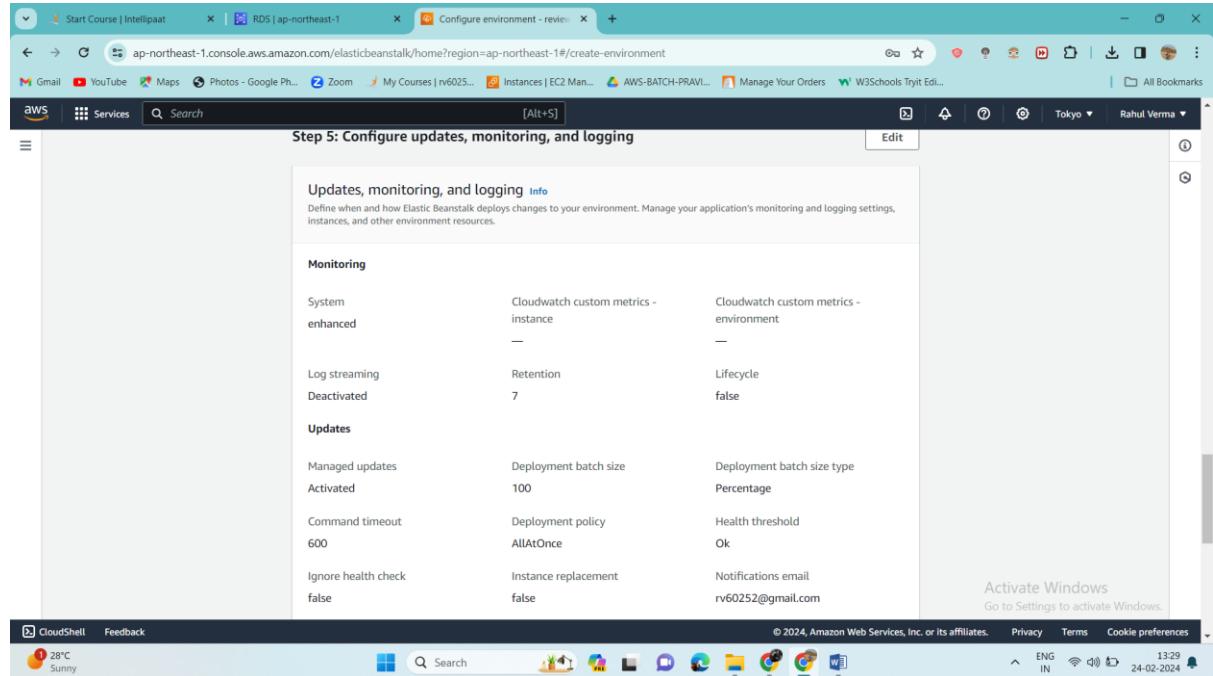
IMDSv1	EC2 Security Groups
Deactivated	sg-09b2aa96c910a7ca9

Capacity

Environment type	Fleet composition	On-demand base
Single instance	On-Demand instance	0
On-demand above base	Capacity rebalancing	Scaling cooldown
0	Deactivated	360
Processor type	Instance types	AMI ID
x86_64	t2.micro	ami-03986ae5c6779bf3f
Availability Zones	Metric	Statistic
Any	NetworkOut	Average

Activate Windows
Go to Settings to activate Windows.

5



Step 5: Configure updates, monitoring, and logging

Updates, monitoring, and logging Info
Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

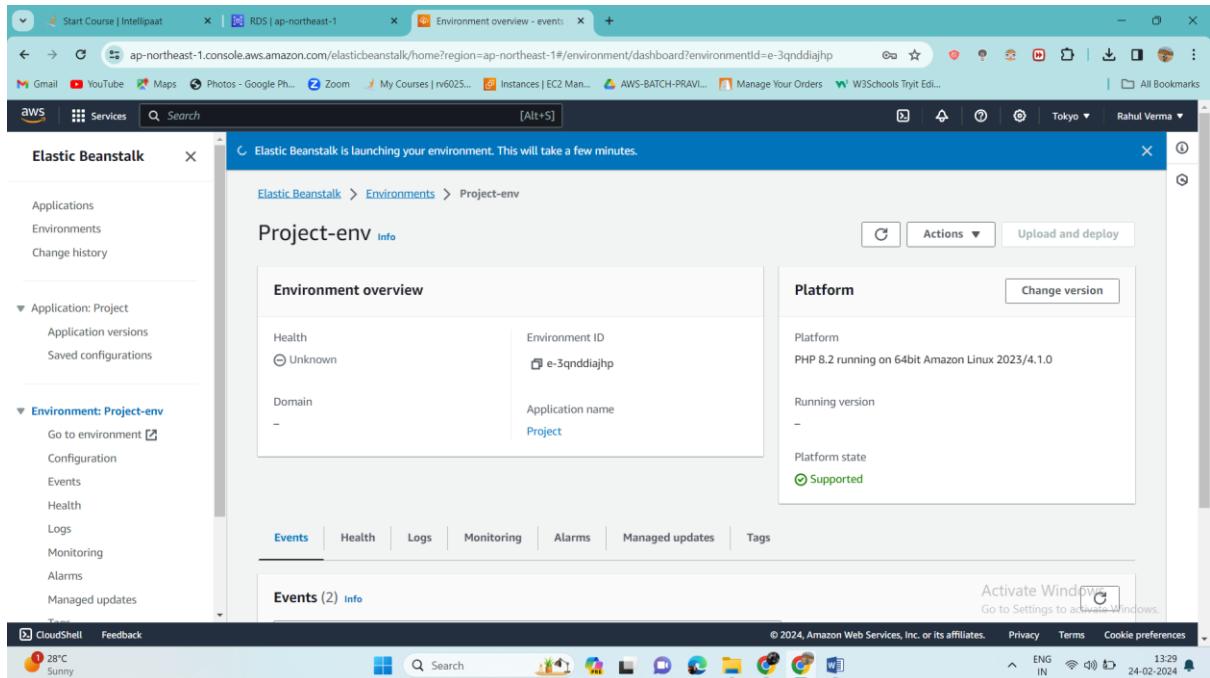
System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
enhanced	—	—

Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	Notifications email
false	false	rv60252@gmail.com

Activate Windows
Go to Settings to activate Windows.

Now we have to wait until it's getting done



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

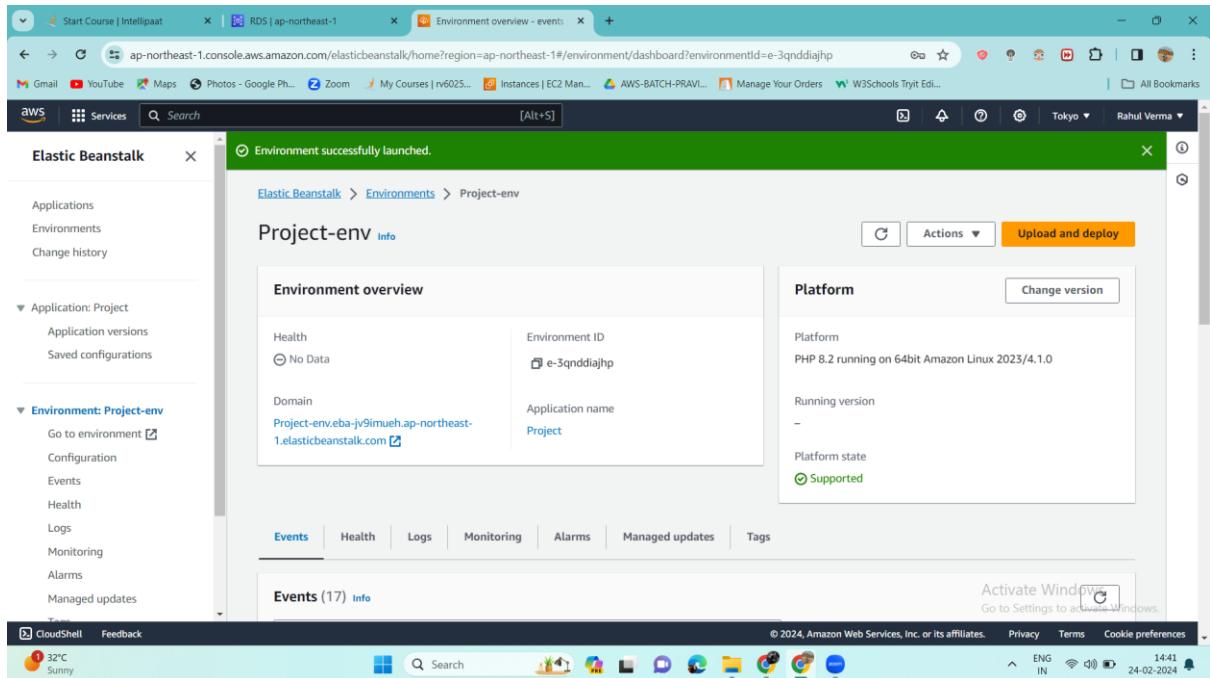
Health	Environment ID
Unknown	e-3qnnddajhp
Domain	Application name
-	Project

The 'Platform' section indicates:

- Platform: PHP 8.2 running on 64bit Amazon Linux 2023/4.1.0
- Running version: -
- Platform state: Supported

The 'Events' tab shows 2 events. The bottom of the page includes standard AWS navigation and status information.

And it's done now



The screenshot shows the AWS Elastic Beanstalk environment overview page for 'Project-env'. A message at the top says 'Environment successfully launched.' The 'Environment overview' section shows the following details:

Health	Environment ID
No Data	e-3qnnddajhp
Domain	Application name
Project-env.eba-jv9imueh.ap-northeast-1.elasticbeanstalk.com	Project

The 'Platform' section indicates:

- Platform: PHP 8.2 running on 64bit Amazon Linux 2023/4.1.0
- Running version: -
- Platform state: Supported

The 'Events' tab shows 17 events. The bottom of the page includes standard AWS navigation and status information.

DONE