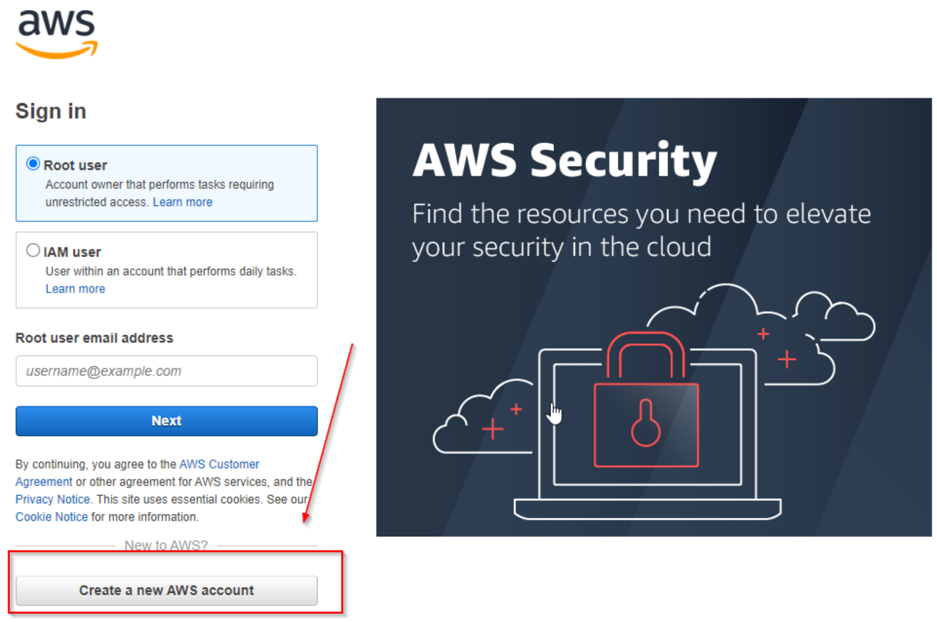
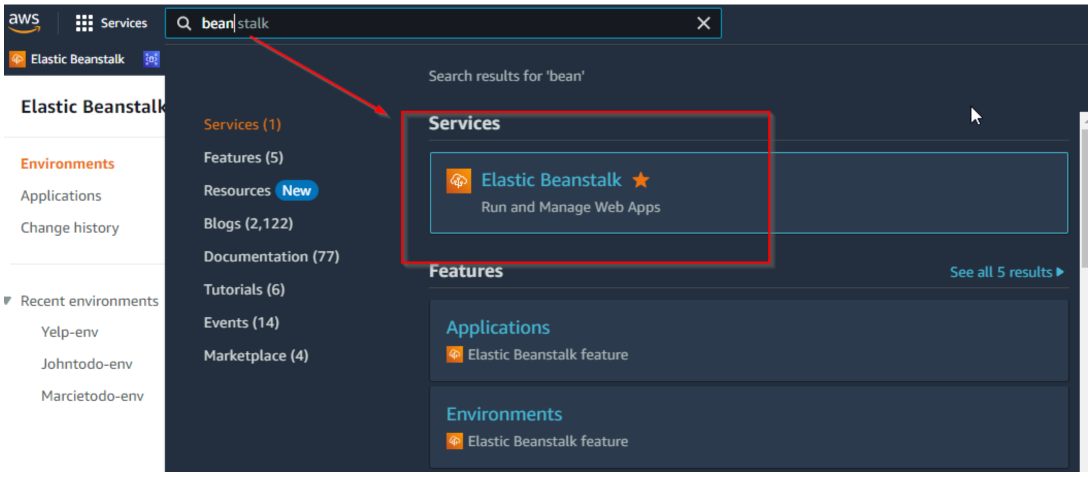
AWS Bean Stalk Deploy MONO PERN Stack

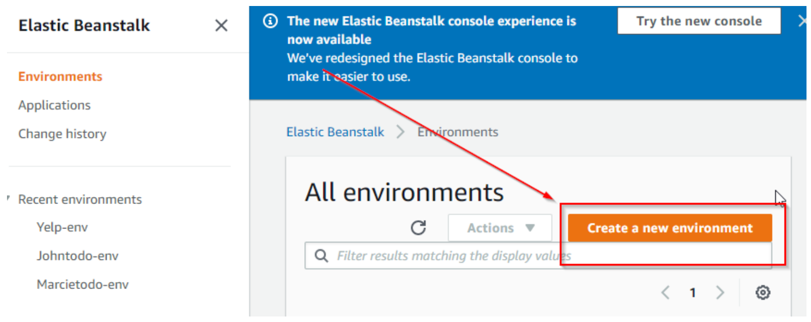
1. Login to AWS Console create a free account if you don't have one already


aws
Sign in
Root user
Account owner that performs tasks requiring unrestricted access. Learn more
IAM user
User within an account that performs daily tasks. Learn more
Root user email address
username@example.com
Next
By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.
New to AWS?
Create a new AWS account
AWS Security
Find the resources you need to elevate your security in the cloud

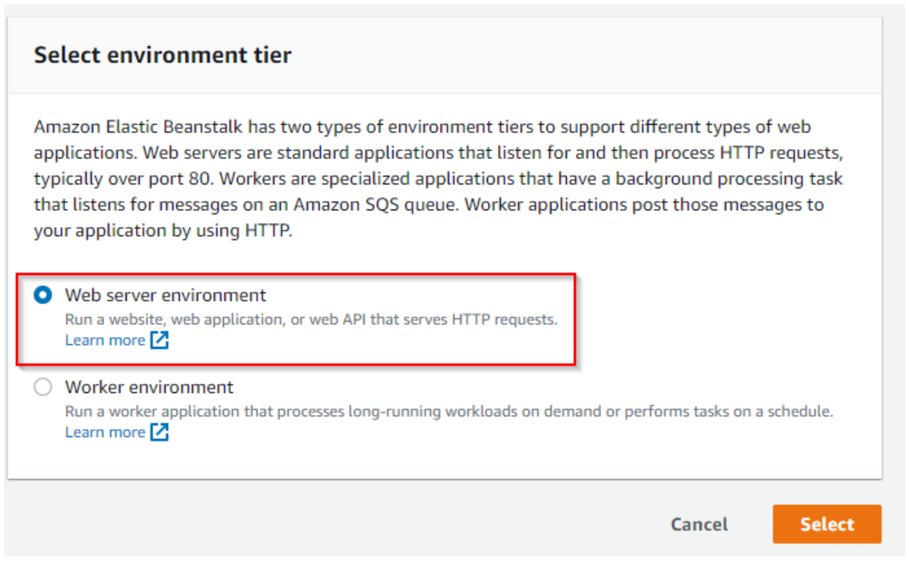
2. Once logged in, you want to search and select the "Elastic Beanstalk" service.



3. Select **"Create a new environment "**


Elastic Beanstalk
Environments
Applications
Change history
Recent environments
Yelp-env
Johntodo-env
Marcietodo-env
The new Elastic Beanstalk console experience is now available
We've redesigned the Elastic Beanstalk console to make it easier to use.
Try the new console
Elastic Beanstalk > Environments
All environments
Actions
Create a new environment
Filter results matching the display values
1

4. Select **“Web Server Environment”**



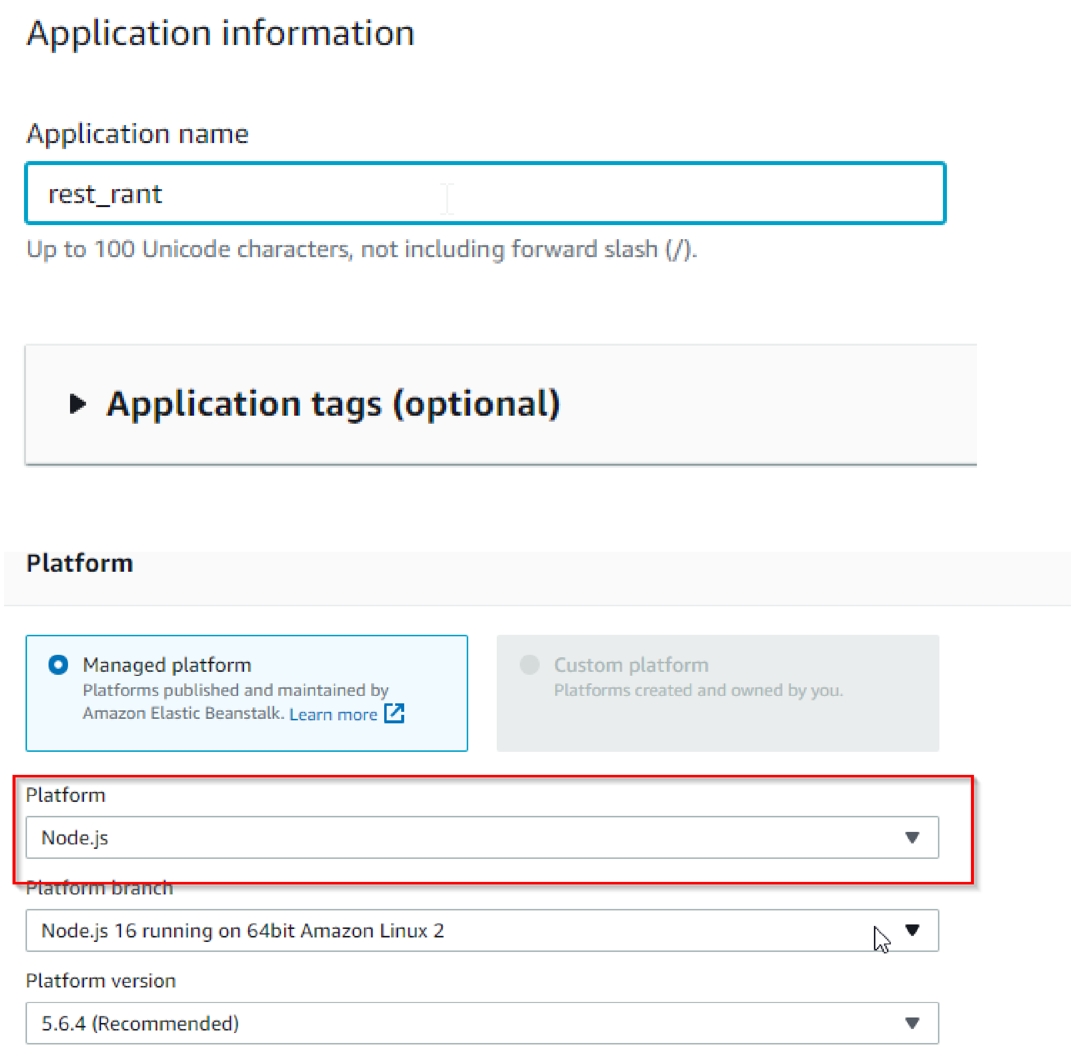
Select environment tier

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web servers are standard applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue. Worker applications post those messages to your application by using HTTP.

[✓ 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

5. Give your application a name then scroll down and select **“Node.js”** as a platform
   Accept the other prefilled defaults for Node



Application information

Application name

rest_rant

Up to 100 Unicode characters, not including forward slash (/).

Platform

[✓ Managed platform]
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

Custom platform
Platforms created and owned by you.

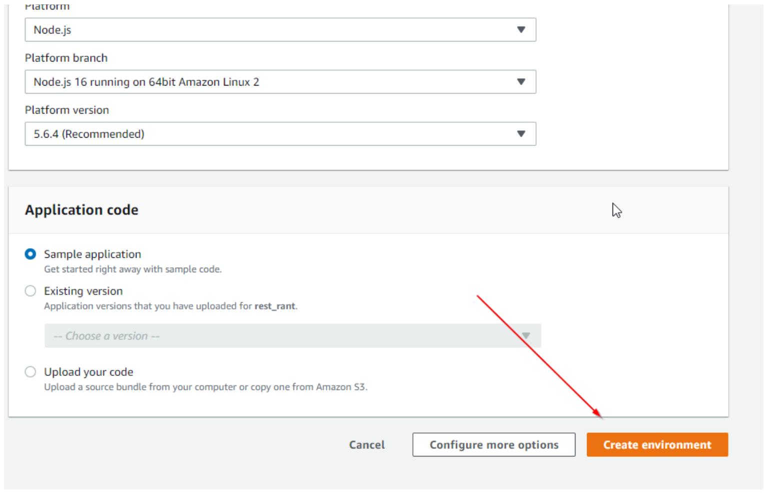
Platform: Node.js

Platform branch: Node.js 16 running on 64bit Amazon Linux 2

Platform version: 5.6.4 (Recommended)

6. Scroll down and select **“Create environment”**

This may take several minutes to build out now.



Platform: Node.js

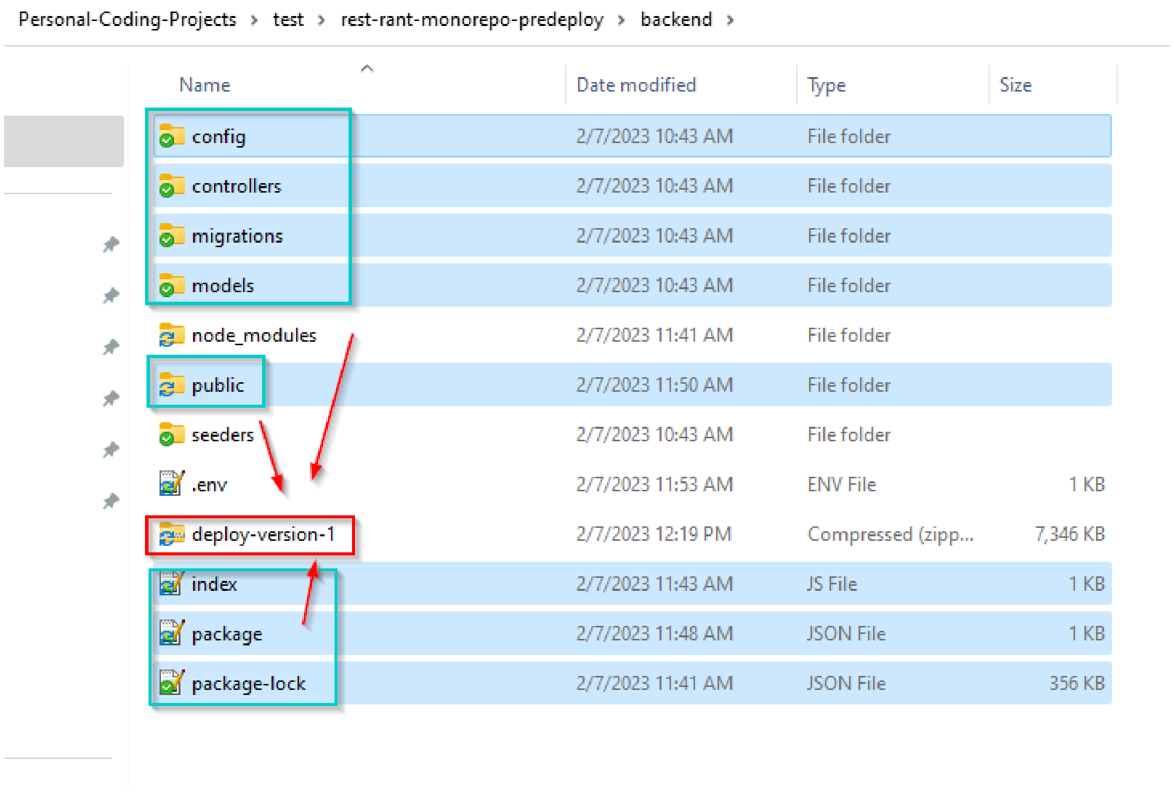
Platform branch: Node.js 16 running on 64bit Amazon Linux 2

Platform version: 5.6.4 (Recommended)

Application code

- [✓ Sample application] — Get started right away with sample code.
- Existing version — Application versions that you have uploaded for rest_rant. -- Choose a version --
- Upload your code — Upload a source bundle from your computer or copy one from Amazon S3.

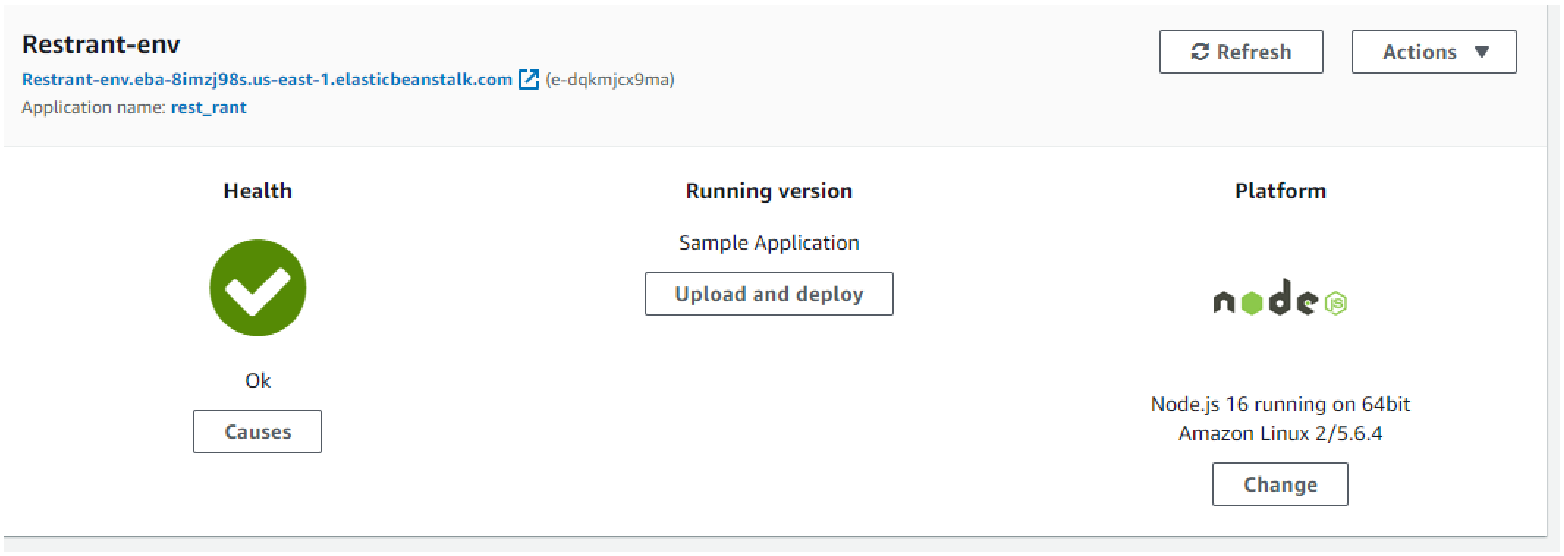
7. At this point you must prepare your local code to upload.
   - Zip up all the code shown here in your backend folder.
   - You can name the zip whatever you like.
   - We will get back to that zip file later



Personal-Coding-Projects › test › rest-rant-monorepo-predeploy › backend ›

Name	Date modified	Type	Size
config	2/7/2023 10:43 AM	File folder	
controllers	2/7/2023 10:43 AM	File folder	
migrations	2/7/2023 10:43 AM	File folder	
models	2/7/2023 10:43 AM	File folder	
node_modules	2/7/2023 11:41 AM	File folder	
public	2/7/2023 11:50 AM	File folder	
seeders	2/7/2023 10:43 AM	File folder	
.env	2/7/2023 11:53 AM	ENV File	1 KB
deploy-version-1	2/7/2023 12:19 PM	Compressed (zipp...	7,346 KB
index	2/7/2023 11:43 AM	JS File	1 KB
package	2/7/2023 11:48 AM	JSON File	1 KB
package-lock	2/7/2023 11:41 AM	JSON File	356 KB

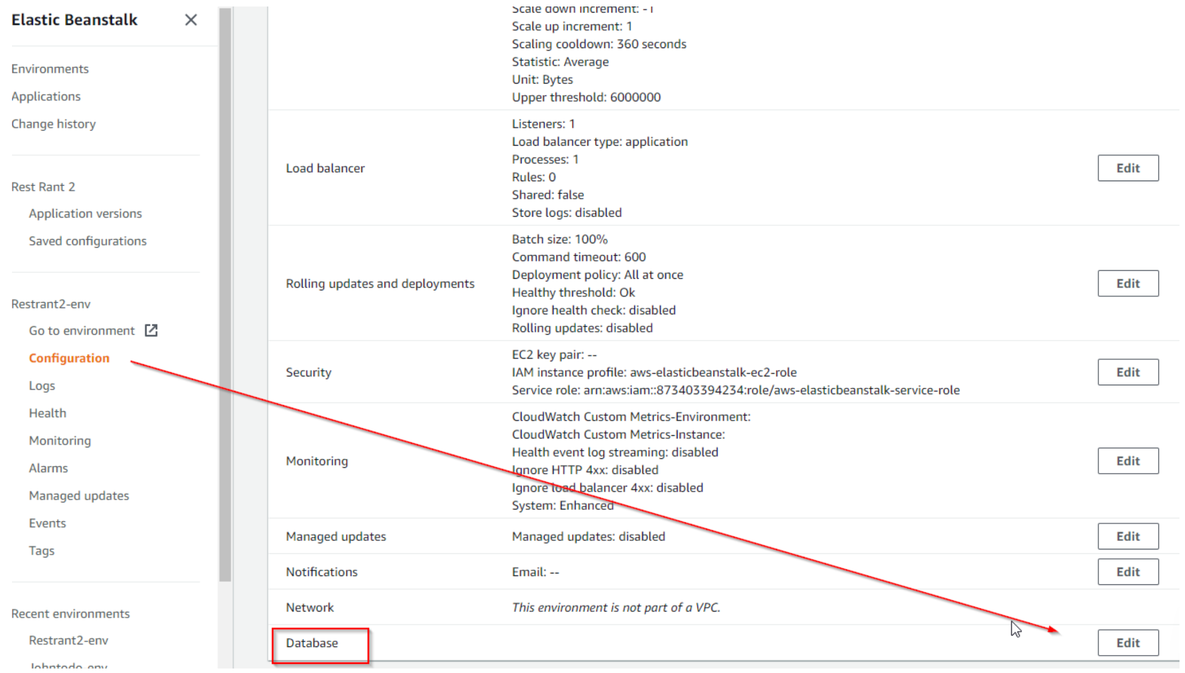
8. Once it’s complete you should see the screen below
   You still have to setup your database and environment variables.



Restrant-env
Restrant-env.eba-8imzj98s.us-east-1.elasticbeanstalk.com (e-dqkmjcx9ma)
Application name: rest_rant

Health	Running version	Platform
Ok	Sample Application	Node.js 16 running on 64bit Amazon Linux 2/5.6.4

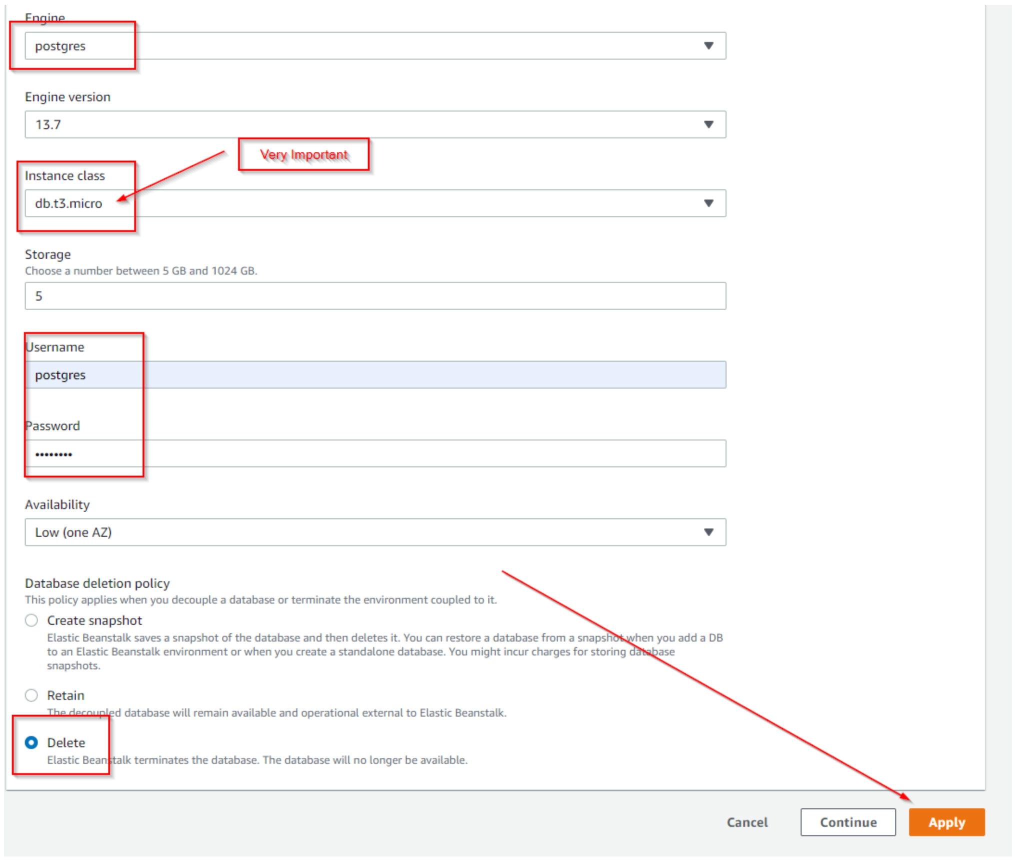
9. Now create Your AWS database
   Select Configuration and scroll down to **“Database”** and select **edit**



	Scale down increment: -1 Scale up increment: 1 Scaling cooldown: 360 seconds Statistic: Average Unit: Bytes Upper threshold: 6000000	
Load balancer	Listeners: 1 Load balancer type: application Processes: 1 Rules: 0 Shared: false Store logs: disabled	Edit
Rolling updates and deployments	Batch size: 100% Command timeout: 600 Deployment policy: All at once Healthy threshold: Ok Ignore health check: disabled Rolling updates: disabled	Edit
Security	EC2 key pair: -- IAM instance profile: aws-elasticbeanstalk-ec2-role Service role: arn:aws:iam::873403394234:role/aws-elasticbeanstalk-service-role	Edit
Monitoring	CloudWatch Custom Metrics-Environment: CloudWatch Custom Metrics-Instance: Health event log streaming: disabled Ignore HTTP 4xx: disabled Ignore load balancer 4xx: disabled System: Enhanced	Edit
Managed updates	Managed updates: disabled	Edit
Notifications	Email: --	Edit
Network	*This environment is not part of a VPC.*	
Database		Edit

- Select **“Postgres”** as the engine
- Select **“db.t3.micro”** as the instance
- (Very important being if you don’t select this instance type you may see some high charges on your AWS bill. This is the “free level” instance)

- Select a **username** and **password** for your AWS database
- Select **"delete"** as a detention policy. (again, to save on possible charges not included in the "free tier")
- Select **"Apply"** this will take some time to re-build again
- Wait for the update to build again, **this may take some time**



Engine
postgres

Engine version
13.7

Instance class
db.t3.micro

Storage
Choose a number between 5 GB and 1024 GB.
5

Username
postgres

Password
••••••••

Availability
Low (one AZ)

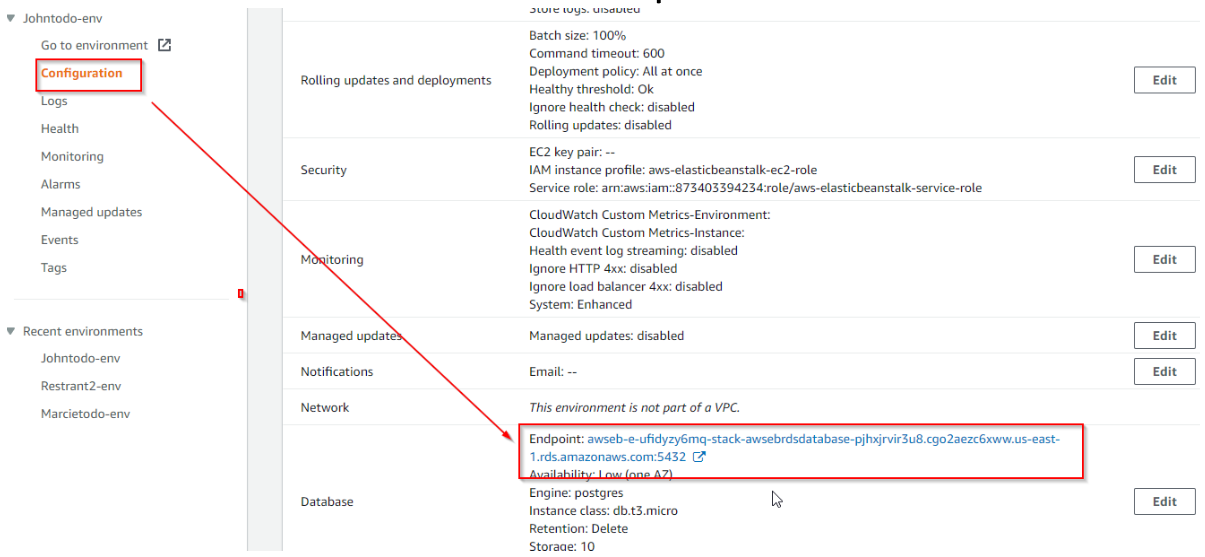
Database deletion policy
This policy applies when you decouple a database or terminate the environment coupled to it.

○ Create snapshot
Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.

○ Retain
The decoupled database will remain available and operational external to Elastic Beanstalk.

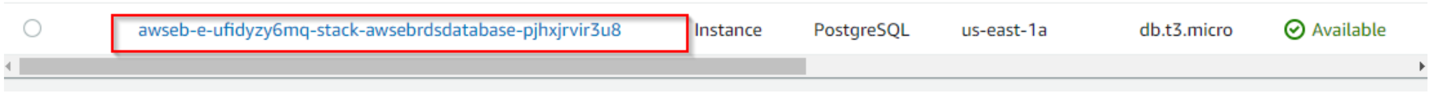
● Delete
Elastic Beanstalk terminates the database. The database will no longer be available.

10. Now you have to setup access to the database from any IP address
    - Click **Configuration** again and scroll down to the database section.
    - Click on the link to the new database **"Endpoint"**



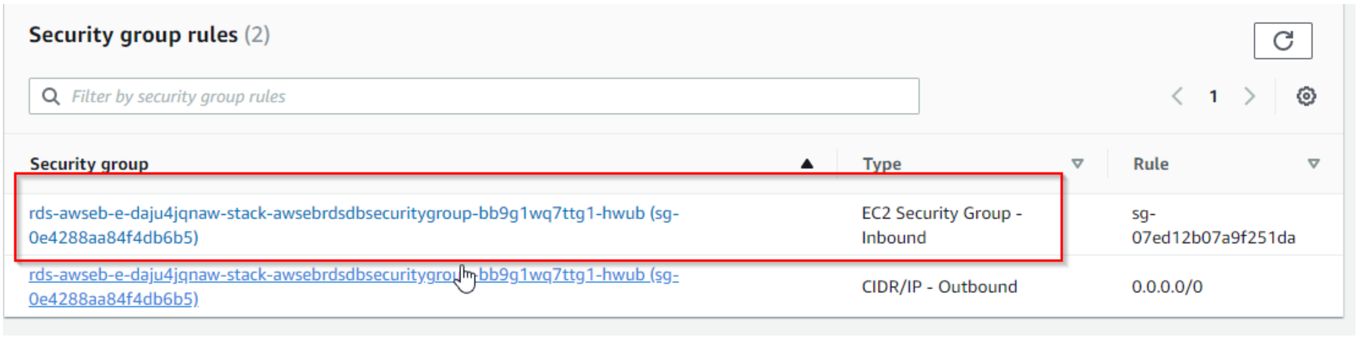
	Store logs: disabled	
Rolling updates and deployments	Batch size: 100% Command timeout: 600 Deployment policy: All at once Healthy threshold: Ok Ignore health check: disabled Rolling updates: disabled	Edit
Security	EC2 key pair: -- IAM instance profile: aws-elasticbeanstalk-ec2-role Service role: arn:aws:iam::873403394234:role/aws-elasticbeanstalk-service-role	Edit
Monitoring	CloudWatch Custom Metrics-Environment: CloudWatch Custom Metrics-Instance: Health event log streaming: disabled Ignore HTTP 4xx: disabled Ignore load balancer 4xx: disabled System: Enhanced	Edit
Managed updates	Managed updates: disabled	Edit
Notifications	Email: --	Edit
Network	*This environment is not part of a VPC.*	
Database	Endpoint: awseb-e-ufidyzy6mq-stack-awsebrdsdatabase-pjhxjrvir3u8.cgo2aezc6xww.us-east-1.rds.amazonaws.com:5432 Availability: Low (one AZ) Engine: postgres Instance class: db.t3.micro Retention: Delete Storage: 10	Edit

Next you will see your list of databases. Click on the one you just created



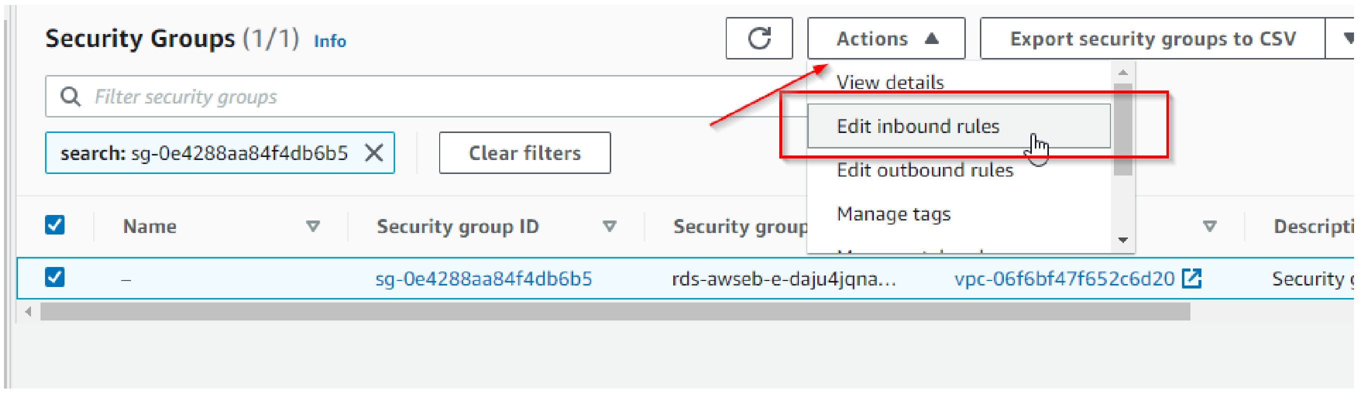
○	awseb-e-ufidyzy6mq-stack-awsebrdsdatabase-pjhxjrvir3u8	Instance	PostgreSQL	us-east-1a	db.t3.micro	✓ Available

- Scroll down to you see **security groups** and click on the **“inbound”** one



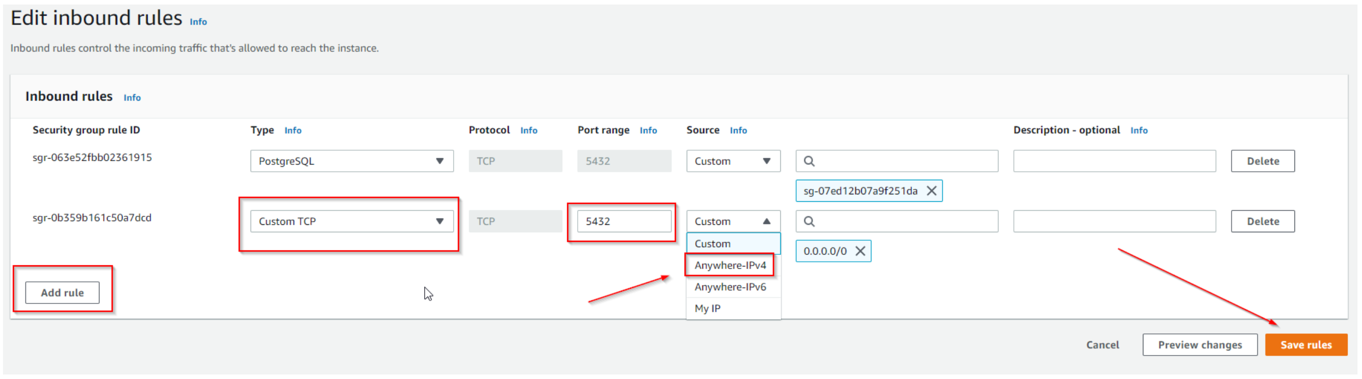
Security group	Type	Rule
rds-awseb-e-daju4jqnaw-stack-awsebrdsdbsecuritygroup-bb9g1wq7ttg1-hwub (sg-0e4288aa84f4db6b5)	EC2 Security Group - Inbound	sg-07ed12b07a9f251da
rds-awseb-e-daju4jqnaw-stack-awsebrdsdbsecuritygroup-bb9g1wq7ttg1-hwub (sg-0e4288aa84f4db6b5)	CIDR/IP - Outbound	0.0.0.0/0

- In the **“actions”** drop down select **“Edit inbound rules”**



Name	Security group ID	Security group		Descripti
–	sg-0e4288aa84f4db6b5	rds-awseb-e-daju4jqna…	vpc-06f6bf47f652c6d20	Security

- Select **“add rule”**, add **“custom TCP”** set the **port** as needed, set it to **“Anywhere ipv4”** and select **“Save Rules”**



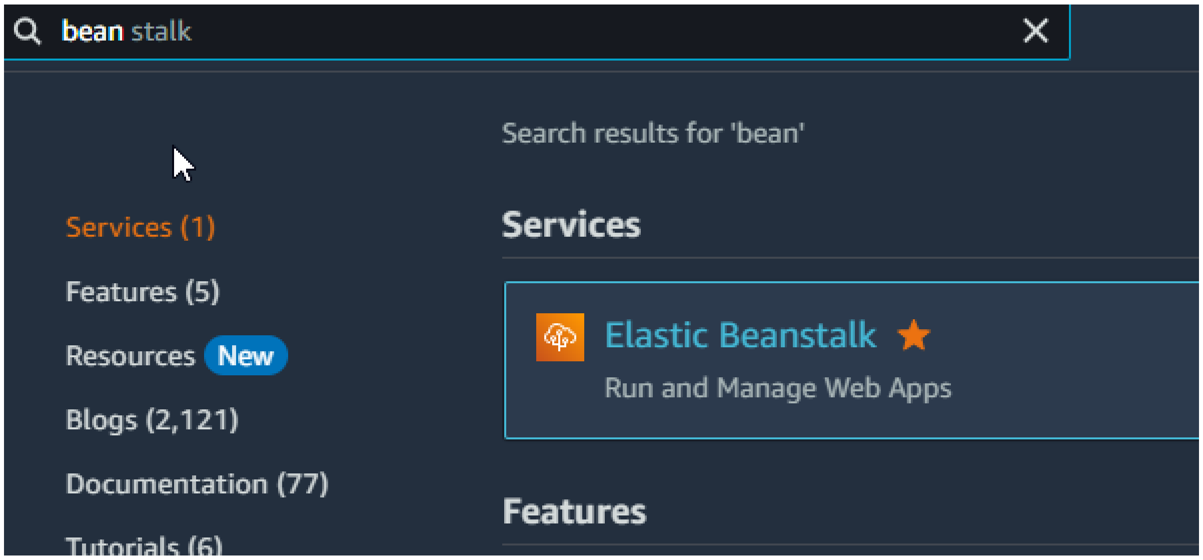
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

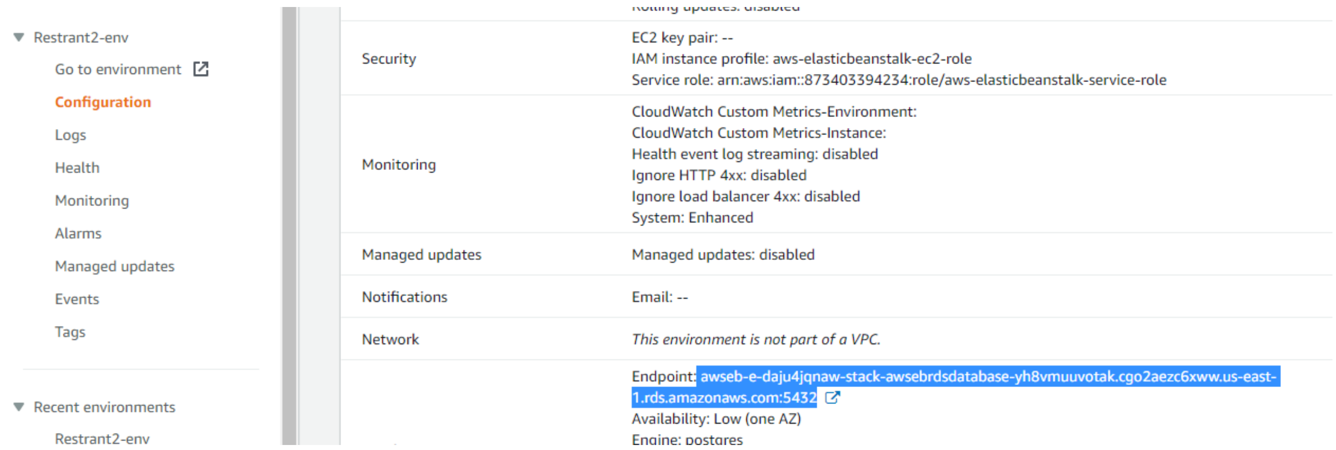
Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-063e52fbb02361915	PostgreSQL	TCP	5432	Custom	sg-07ed12b07a9f251da		Delete
sgr-0b359b161c50a7dcd	Custom TCP	TCP	5432	Custom	0.0.0.0/0		Delete

- Now search and go back to the “beanstalk” settings.

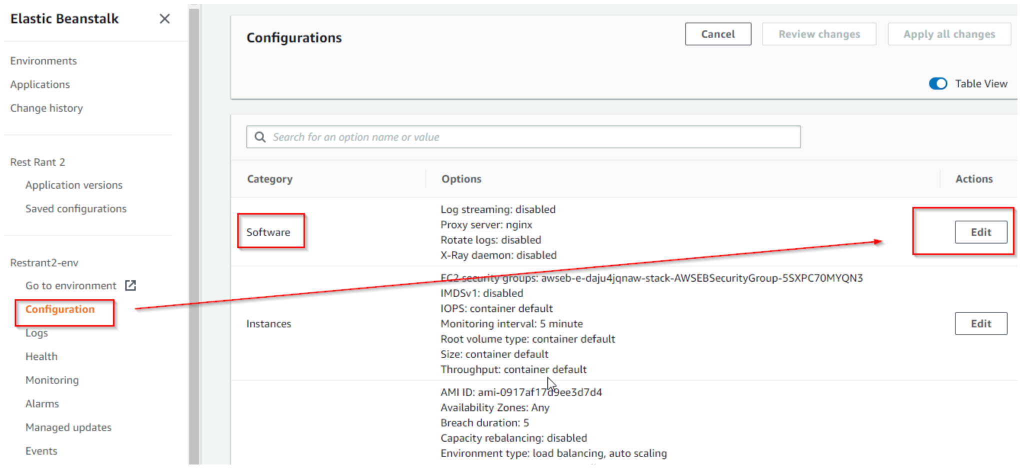


11. Time to add your system environment variables
    - First you need to copy that database endpoint that you created under **configuration** (You will need this link to enter as a variable but **remove the :5432 port number from the end**)



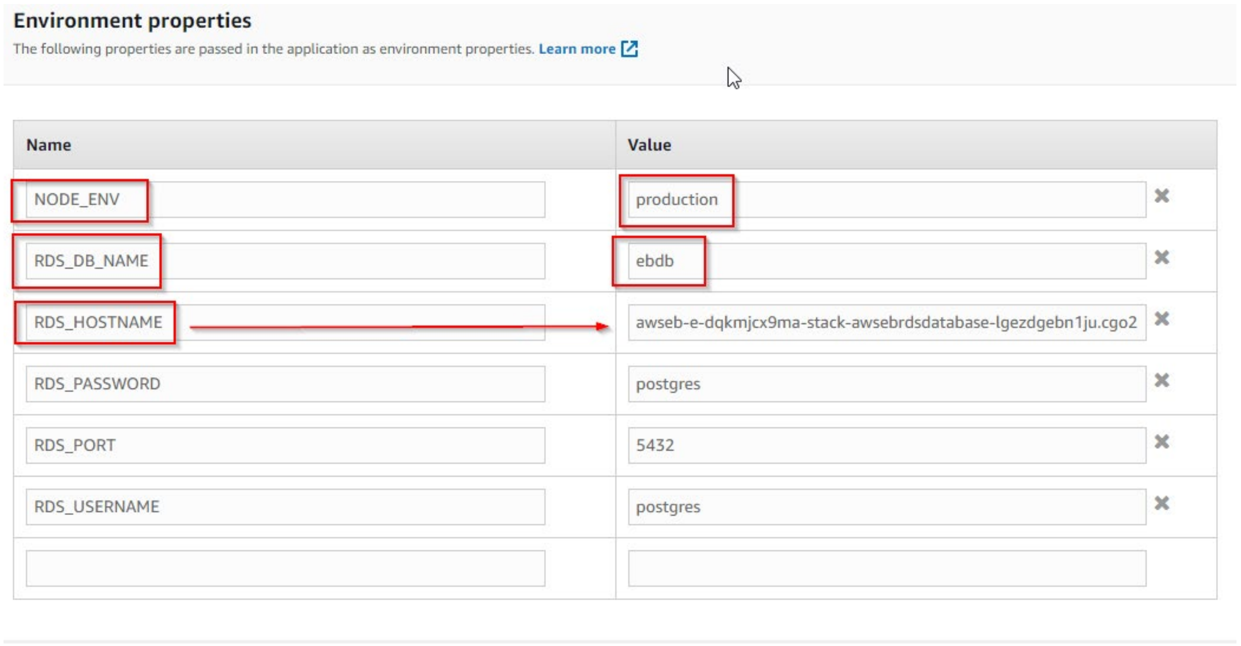
Security	EC2 key pair: -- IAM instance profile: aws-elasticbeanstalk-ec2-role Service role: arn:aws:iam::873403394234:role/aws-elasticbeanstalk-service-role
Monitoring	CloudWatch Custom Metrics-Environment: CloudWatch Custom Metrics-Instance: Health event log streaming: disabled Ignore HTTP 4xx: disabled Ignore load balancer 4xx: disabled System: Enhanced
Managed updates	Managed updates: disabled
Notifications	Email: --
Network	*This environment is not part of a VPC.*
	Endpoint: awseb-e-daju4jqnaw-stack-awsebrdsdatabase-yh8vmuuvotak.cgo2aezc6xww.us-east-1.rds.amazonaws.com:5432 Availability: Low (one AZ) Engine: postgres

- Next go to **configuration** “software” and select **“edit”**



Category	Options	Actions
Software	Log streaming: disabled Proxy server: nginx Rotate logs: disabled X-Ray daemon: disabled	Edit
Instances	EC2 security groups: awseb-e-daju4jqnaw-stack-AWSEBSecurityGroup-5SXPC70MYQN3 IMDSv1: disabled IOPS: container default Monitoring interval: 5 minute Root volume type: container default Size: container default Throughput: container default	Edit
	AMI ID: ami-0917af17d9ee3d7d4 Availability Zones: Any Breach duration: 5 Capacity rebalancing: disabled Environment type: load balancing, auto scaling	

- Scroll down to Environment properties and enter your env variables for production from your local .env file.
- It's important that you include that NODE_ENV=**production** setting
  Note that is case sensitive match what you used in your code
- *Here is where you will need that newly created database end point minus the " **:port#"** at the end for the RDS_HOSTNAME*
- The other settings should match what you used previously when creating the DB on AWS
- The DB name on the AWS side (RDS_DB_NAME should be named **"ebdb"**
  We will get back to that later

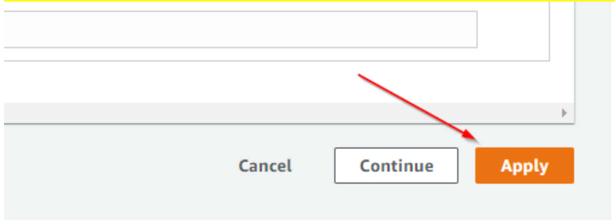


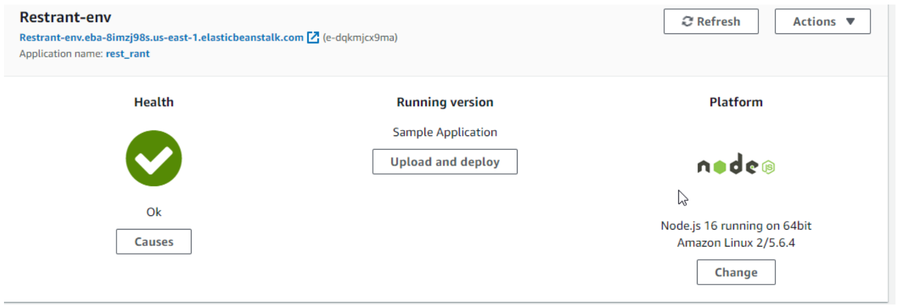
Environment properties

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

Name	Value
NODE_ENV	production
RDS_DB_NAME	ebdb
RDS_HOSTNAME	awseb-e-dqkmjcx9ma-stack-awsebrdsdatabase-lgezdgebn1ju.cgo2
RDS_PASSWORD	postgres
RDS_PORT	5432
RDS_USERNAME	postgres

- Click Apply
  This will now trigger another Beanstalk rebuild over several minutes





Restrant-env

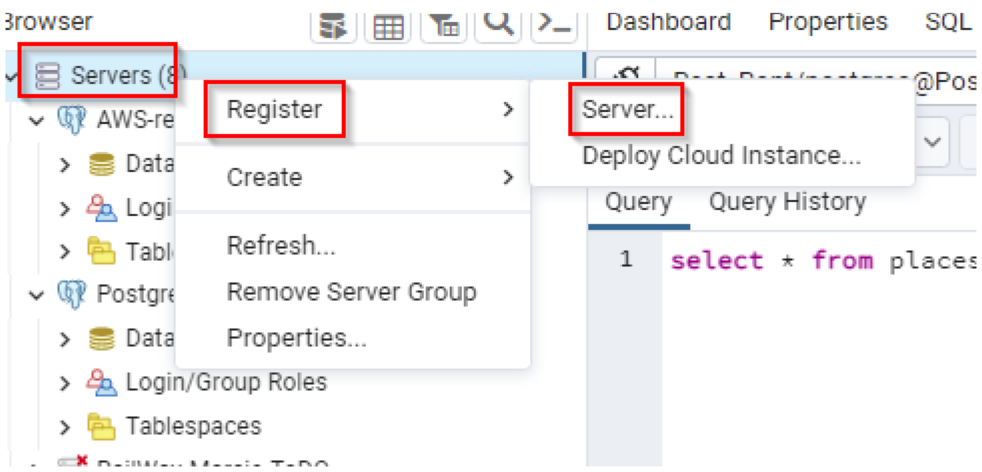
Restrant-env.eba-8imzj98s.us-east-1.elasticbeanstalk.com (e-dqkmjcx9ma)

Application name: rest_rant

Next setup the Database on AWS with your backup file

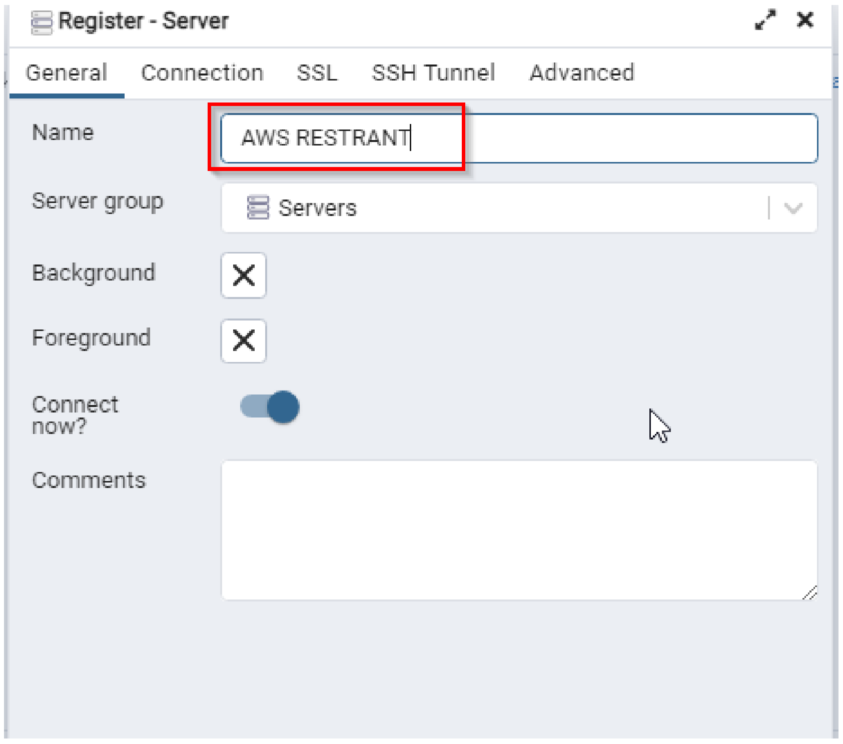
In the local PGADMIN tool connect to your AWS database

- First Right click on servers and select **"add new server"**



```
select * from places
```

- Give the server a name and select **"Connection"**

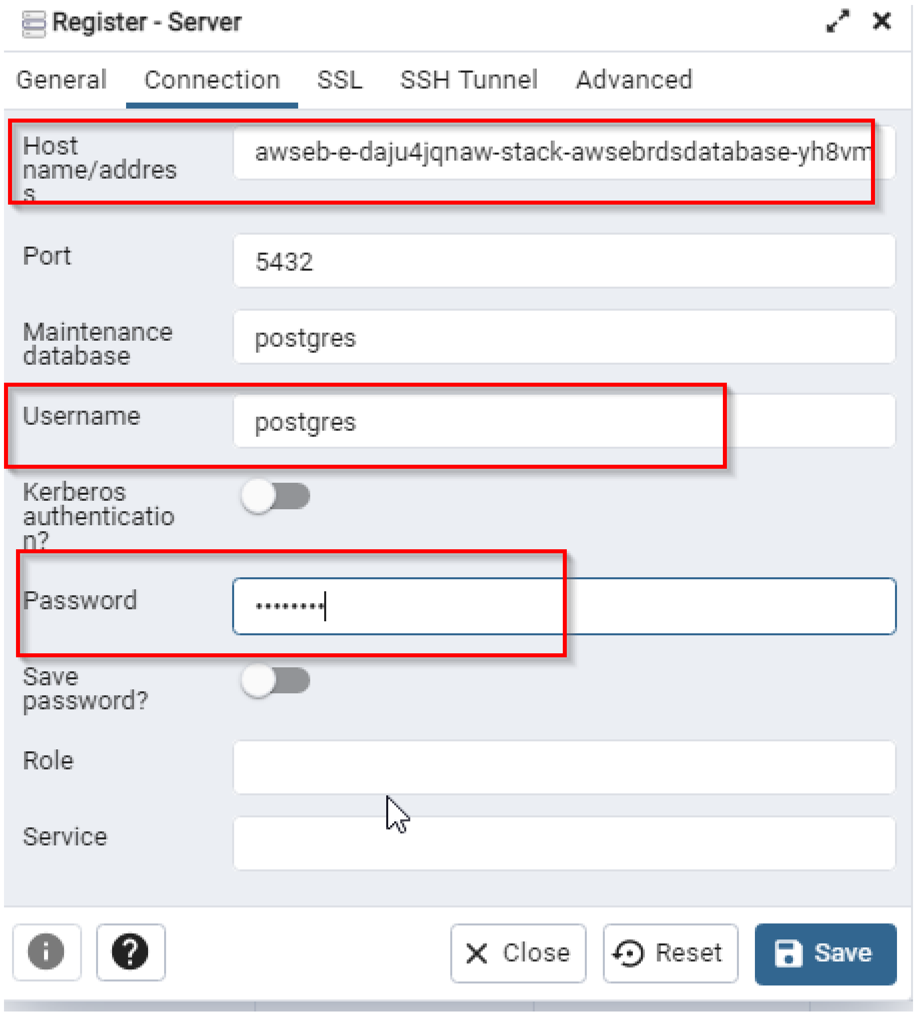


Name: AWS RESTRANT

Server group: Servers

Connect now? ●

- Click **"Connection"** and enter The Server Details to connect remember to remove the port from the end again (:5432)
- Don't forget to enter the password you selected on the AWS side for the database
- Click **"Save"**

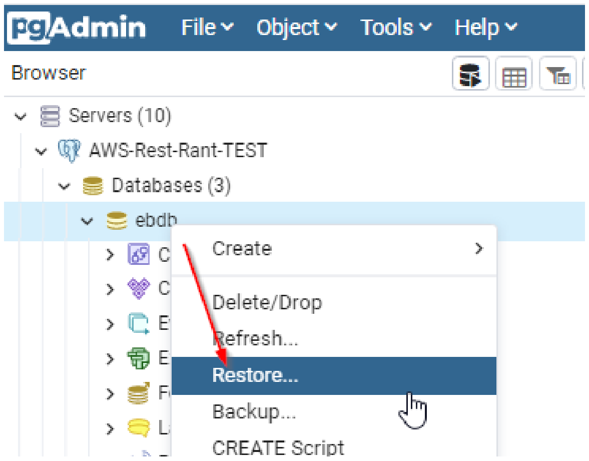


Register - Server

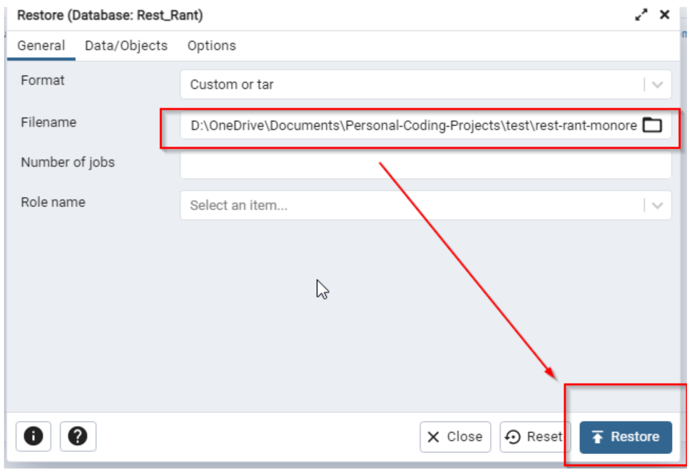
General | Connection | SSL | SSH Tunnel | Advanced

Host name/address	awseb-e-daju4jqnaw-stack-awsebrdsdatabase-yh8vm
Port	5432
Maintenance database	postgres
Username	postgres
Kerberos authentication?	
Password	••••••••
Save password?	
Role	
Service	

- Right Click on the **ebdb** database on that AWS Server and select **Restore**



- Browse to that same backup slq file that we used earlier when creating the local database



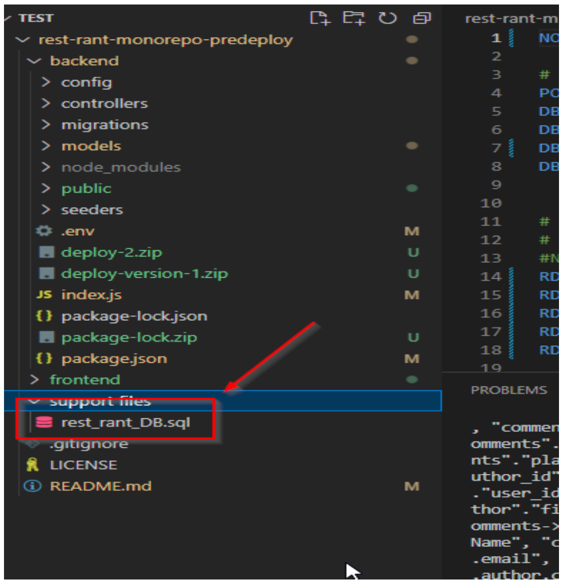
Restore (Database: Rest_Rant)

Format: Custom or tar

Filename: D:\OneDrive\Documents\Personal-Coding-Projects\test\rest-rant-monore

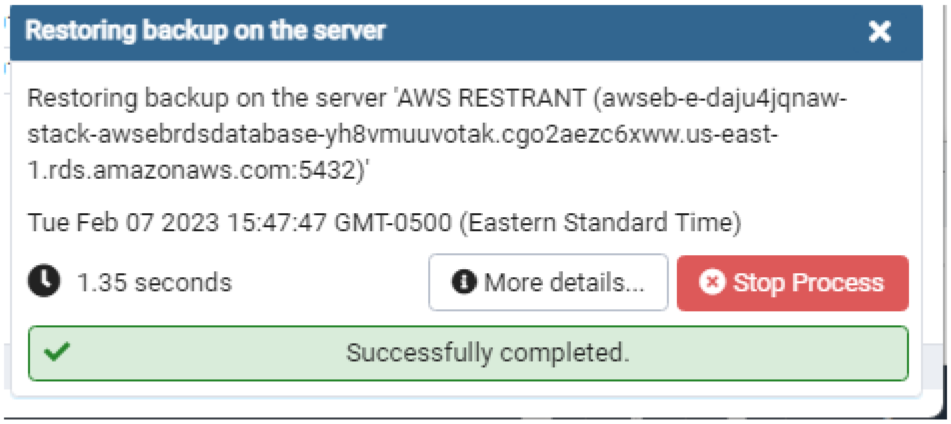
Number of jobs:

Role name: Select an item...



```
TEST
rest-rant-monorepo-predeploy
  backend
    config
    controllers
    migrations
    models
    node_modules
    public
    seeders
    .env                    M
    deploy-2.zip            U
    deploy-version-1.zip    U
    index.js                M
    package-lock.json
    package-lock.zip        U
    package.json            M
  frontend
  support files
    rest_rant_DB.sql
  .gitignore
  LICENSE
  README.md                 M
```

- Your Database should now be created with the data from your development side



Restoring backup on the server

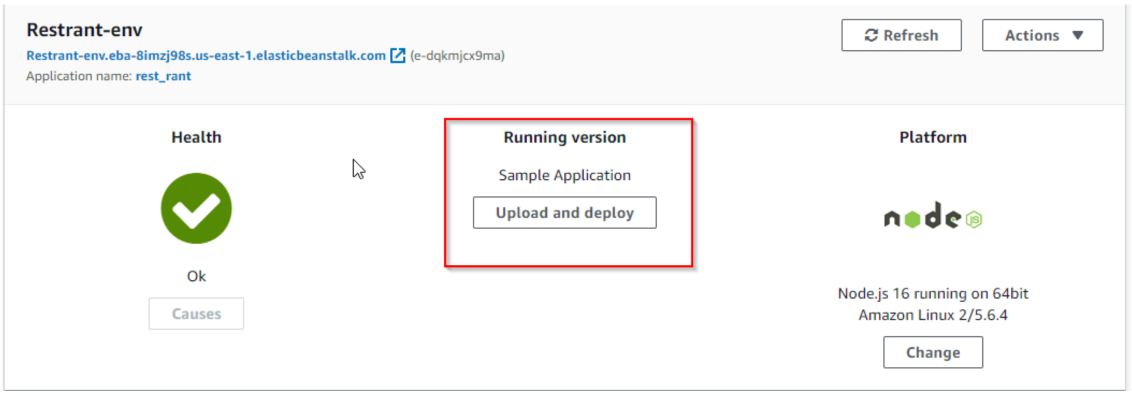
Restoring backup on the server 'AWS RESTRANT (awseb-e-daju4jqnaw-stack-awsebrdsdatabase-yh8vmuuvotak.cgo2aezc6xww.us-east-1.rds.amazonaws.com:5432)'

Tue Feb 07 2023 15:47:47 GMT-0500 (Eastern Standard Time)

1.35 seconds

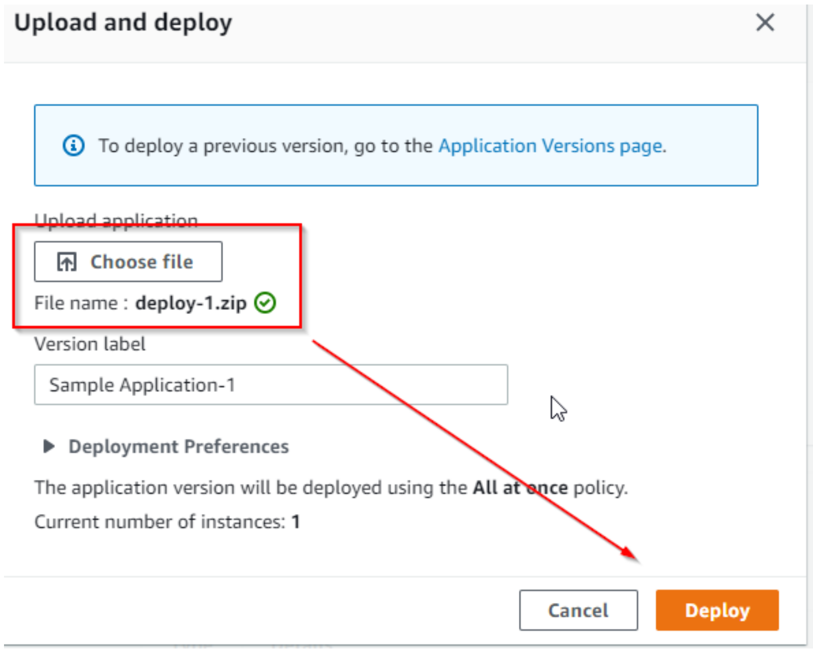
Successfully completed.

- Now we need to circle back to the Zip file you created earlier
  From your beanstalk main screen select **"Upload and Deploy"**



Restrant-env
Restrant-env.eba-8imzj98s.us-east-1.elasticbeanstalk.com (e-dqkmjcx9ma)
Application name: rest_rant

- Click **Choose file** and browse to the zip file you created earlier then select it.
  At this point click **Deploy** and wait one last time.



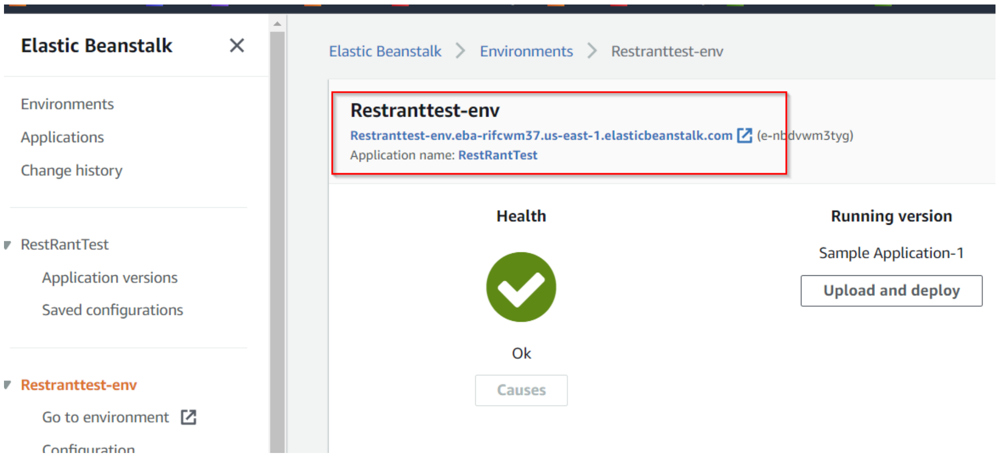
Upload and deploy

To deploy a previous version, go to the Application Versions page.

Upload application
File name : **deploy-1.zip**
Version label: Sample Application-1

The application version will be deployed using the **All at once** policy.
Current number of instances: **1**

- Once the deploy is complete
  Click the link in AWS Beanstalk and test your app



Elastic Beanstalk > Environments > Restranttest-env

Restranttest-env
Restranttest-env.eba-rifcwm37.us-east-1.elasticbeanstalk.com (e-nbdvwm3tyg)
Application name: RestRantTest