

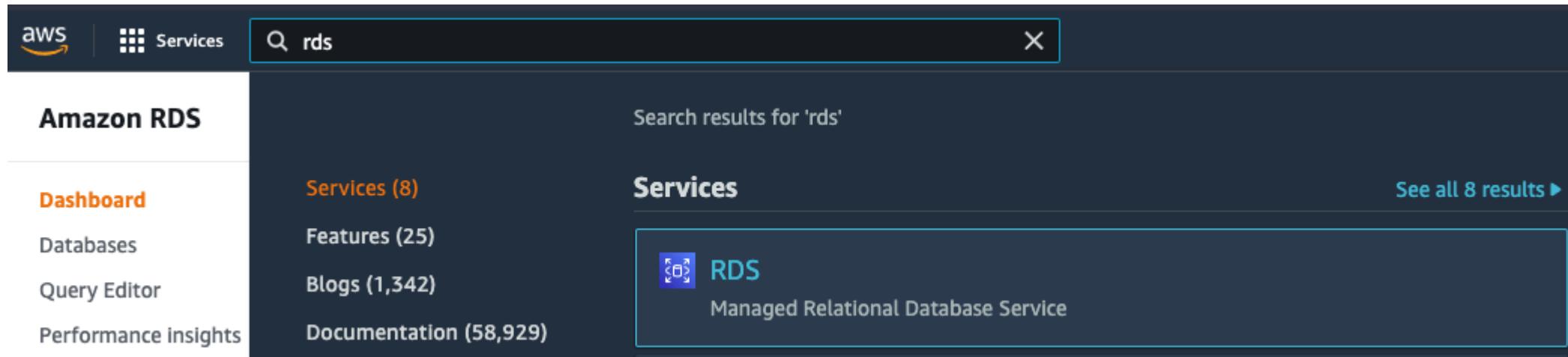
Infrastructure Description

AWS Services

- RDS Database.
- S3 Bucket.
- Elastic Beanstalk.

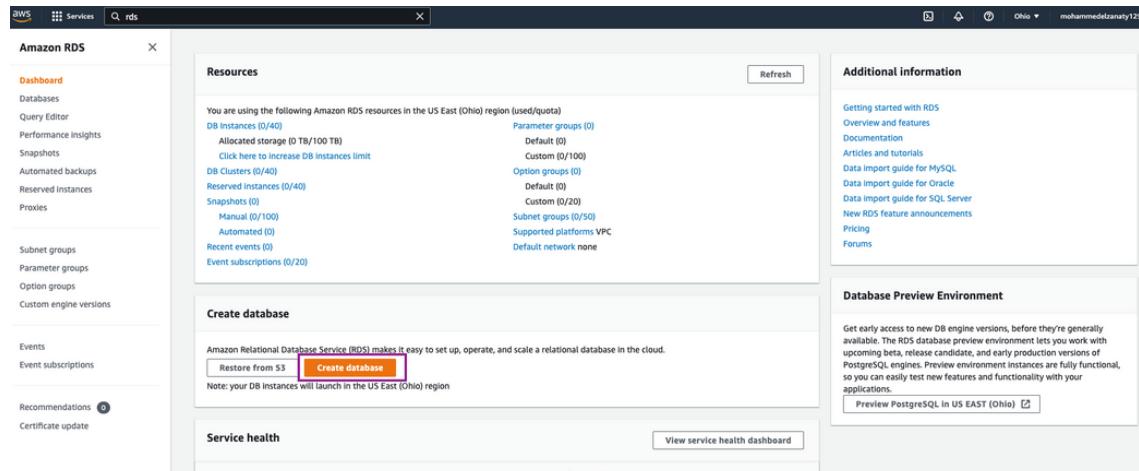
Create Database

* Navigate to RDS :

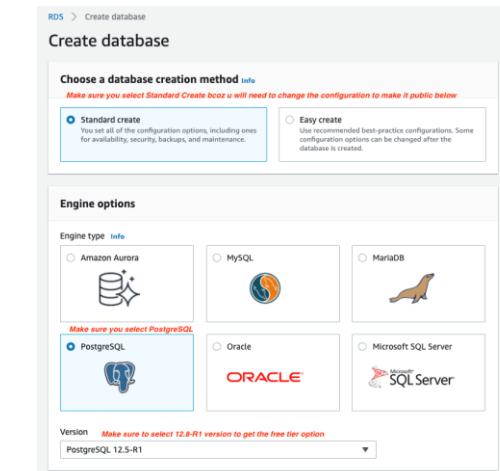


Create Database

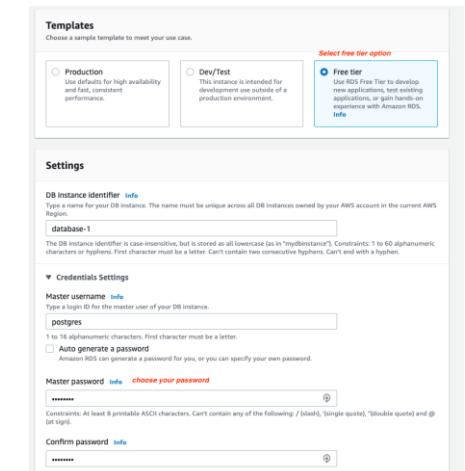
* Create Database:



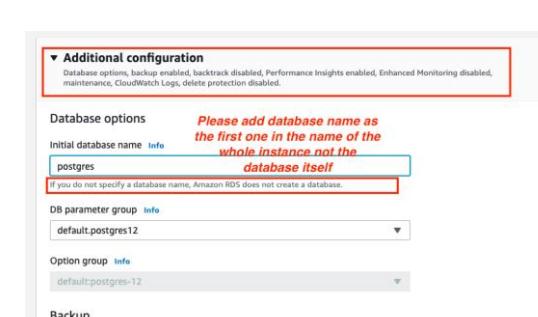
The screenshot shows the AWS RDS Dashboard. On the left sidebar, under 'Databases', there is a red box around the 'Create database' button. The main area displays 'DB Instances (0/40)' and 'DB Clusters (0/40)'. A red box highlights the 'Create database' button in the 'Create database' section.



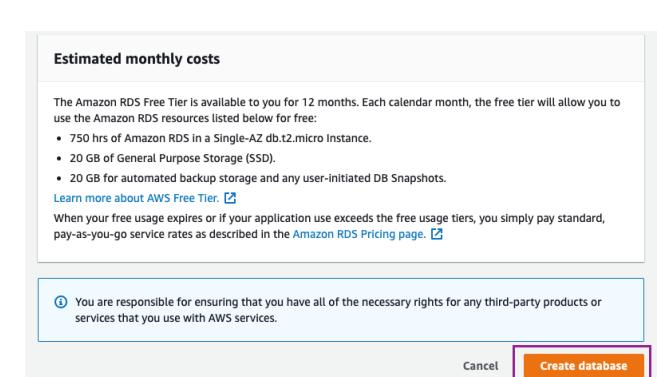
This screenshot shows the 'Create database' wizard. Step 1: Choose a database creation method. It offers two options: 'Standard create' (selected) and 'Easy create'. A red box highlights the 'Standard create' option.



This screenshot shows the 'Create database' wizard. Step 2: Engine options. It lists various engines: Amazon Aurora, MySQL, MariaDB, PostgreSQL (selected), Oracle, and Microsoft SQL Server. A red box highlights the 'PostgreSQL' engine.



This screenshot shows the 'Create database' wizard. Step 3: Additional configuration. It includes fields for 'Initial database name' (set to 'postgres'), 'DB parameter group' (set to 'default.postgres12'), 'Option group' (set to 'default.postgres12'), and 'Backup' settings. A red box highlights the 'Initial database name' field.



This screenshot shows the 'Create database' wizard. Step 4: Estimated monthly costs. It provides information about the free tier and pay-as-you-go service rates. A red box highlights the note at the bottom: 'You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.'

Create Database

* Edit Database & Check Connection:

The screenshot shows the AWS RDS Databases page. A red box highlights the 'DB Identifier' column, and another red box highlights the 'database-1' entry. A callout text 'Click on your database instance' points to the highlighted row.

The screenshot shows the AWS EC2 Security Groups page. It displays two inbound rules: one for 'All traffic' from 'Custom' source IP '0.0.0.0' and another for 'All traffic' from 'Custom' source IP 'sg-06489d6c6b7038fdd'. A red box highlights the second rule. A callout text 'this will give anyone literally anyone can access but for don't worry know you so make it 😊' is shown below the rules.

The screenshot shows the AWS RDS database instance details page for 'database-1'. It includes sections for 'Summary' and 'Connectivity & security'. In the 'Connectivity & security' section, a red box highlights the 'Endpoint & port' table, which lists the endpoint as 'database-1.2ah0mt5uy.us-east-2.rds.amazonaws.com' and port as '5432'. Another red box highlights the 'Security' section, specifically the 'VPC security groups' field containing 'default (sg-06489d6c6b7038fdd) Active'. A third red box highlights the 'Publicly accessible' field set to 'Yes'. A callout text 'Click the active security group link' points to the security group name in the VPC security groups list.

The screenshot shows the AWS EC2 Security Groups page. It displays a single security group named 'sg-06489d6c6b7038fdd' with the status 'default'. A red box highlights the 'Inbound rules' tab. A callout text 'The security group will be selected by default, if not just select it' is shown above the table.

The screenshot shows the AWS RDS Connection page for 'database-2'. It displays a connection status message 'Successfully connected!' with a green 'OK' button. A red box highlights the 'Endpoint & port' table, which lists the endpoint as 'database-2.2ah0mt5uy.us-east-2.rds.amazonaws.com' and port as '5432'. A callout text 'Make sure the instance is publicly accessible' is shown below the connection message.

Create S3 Bucket

* Create S3 Bucket:

The screenshot shows the AWS search interface. The search bar at the top contains 's3'. Below the search bar, there is a sidebar with 'Services' and 'Features' sections. The main search results area shows 'Search results for 's3'' and a list of services, with 'S3' being the first item.

The screenshot shows the Amazon S3 buckets list page. The left sidebar has 'Buckets' selected. The main area displays a table with one row: 'No buckets'. A 'Create bucket' button is visible at the bottom of the table.

The screenshot shows the 'Create bucket' wizard. The 'General configuration' step is active. It includes fields for 'Bucket name' (set to 'yonisify-udagram') and 'AWS Region' (set to 'US East (Ohio) us-east-2'). There is also a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button. The 'Object Ownership' step is shown below, with the 'ACLs disabled (recommended)' option selected. The 'Bucket owner enforced' setting is also indicated.

Create S3 Bucket

* Some Configuration in Create S3 Bucket:

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable
 Enable

Tags (0) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

Disable
 Enable

Create S3 Bucket

* Edit S3 Bucket Policy:

The screenshot shows the AWS S3 Buckets page. On the left, there's a sidebar with options like 'Access Points', 'Object Lambda Access Points', etc. The main area shows an 'Account snapshot' and a table for 'Buckets'. A red box highlights the 'Create bucket' button at the top right of the table.

The screenshot shows the 'yonisify-udagram' bucket details page. The 'Permissions' tab is selected. A red box highlights the 'Edit' button under the 'Bucket policy' section. Below it, a note says 'It is important to disable public writes to prevent people from editing our website while still enabling public reads to allow it to be visited.' and 'The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts.' At the bottom, there's a link 'add here below snippet'.

The screenshot shows the 'Edit bucket policy' page for the 'yonisify-udagram' bucket. The 'Policy' section displays the following JSON code:

```
1  {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Sid": "PublicReadGetObject",
6             "Effect": "Allow",
7             "Principal": "*",
8             "Action": [
9                 "s3:GetObject"
10            ],
11            "Resource": [
12                "arn:aws:s3:::yonisify-udagram/*"
13            ]
14        }
15    ]
16 }
```

A red box highlights the 'Edit statement' button. At the bottom right, there's a note 'Don't Forget to save changes'.

Create S3 Bucket

* Edit S3 Bucket Properties:

The screenshot shows the AWS S3 console interface. The top navigation bar includes the AWS logo, a services menu, a search bar, and user information. The left sidebar has a 'Buckets' section with options like Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3. It also includes sections for Block Public Access settings for this account, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3.

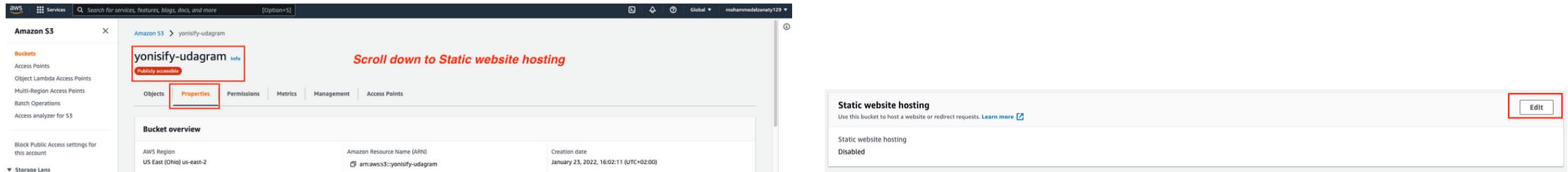
The main content area displays a success message: "Successfully edited bucket policy." Below this, the bucket name "yonisify-udagram" is shown with a "Info" link and a "Publicly accessible" status indicator. The "Permissions" tab is selected, showing a "Permissions overview" section with an "Access" card. The "Access" card displays a warning: "As you see there's warning as S3 bucket now is public".

Under the "Permissions overview" section, there is a "Block public access (bucket settings)" section. It explains that public access is granted through various methods and recommends turning on "Block all public access". A "Edit" button is available for this setting. Below this, there is a "Block all public access" section with a status of "Off" and a note about individual settings for the bucket.

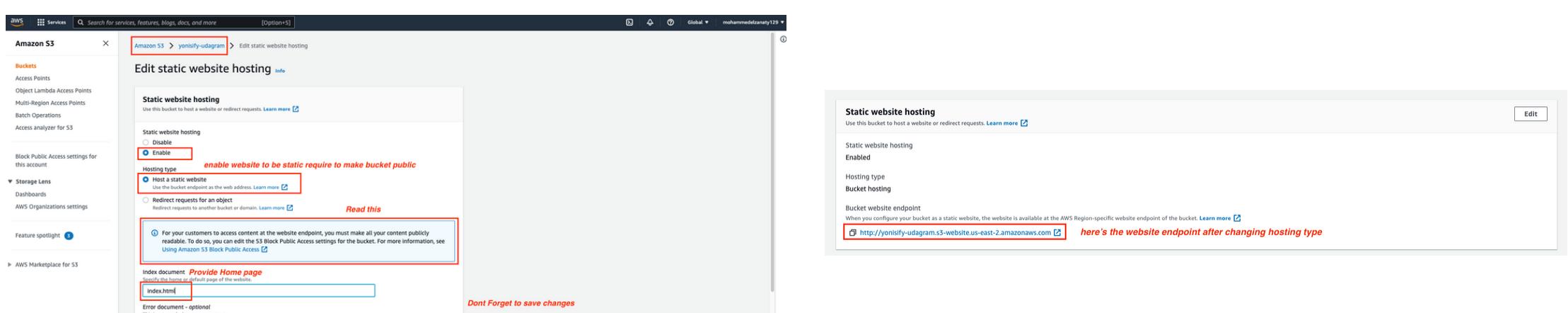
At the bottom, there is a "Bucket policy" section with an "Edit" and "Delete" button, and a "Copy" button. The policy JSON code is partially visible: { "Version": "2012-10-17",

Create S3 Bucket

* Configure S3 Bucket Endpoint:



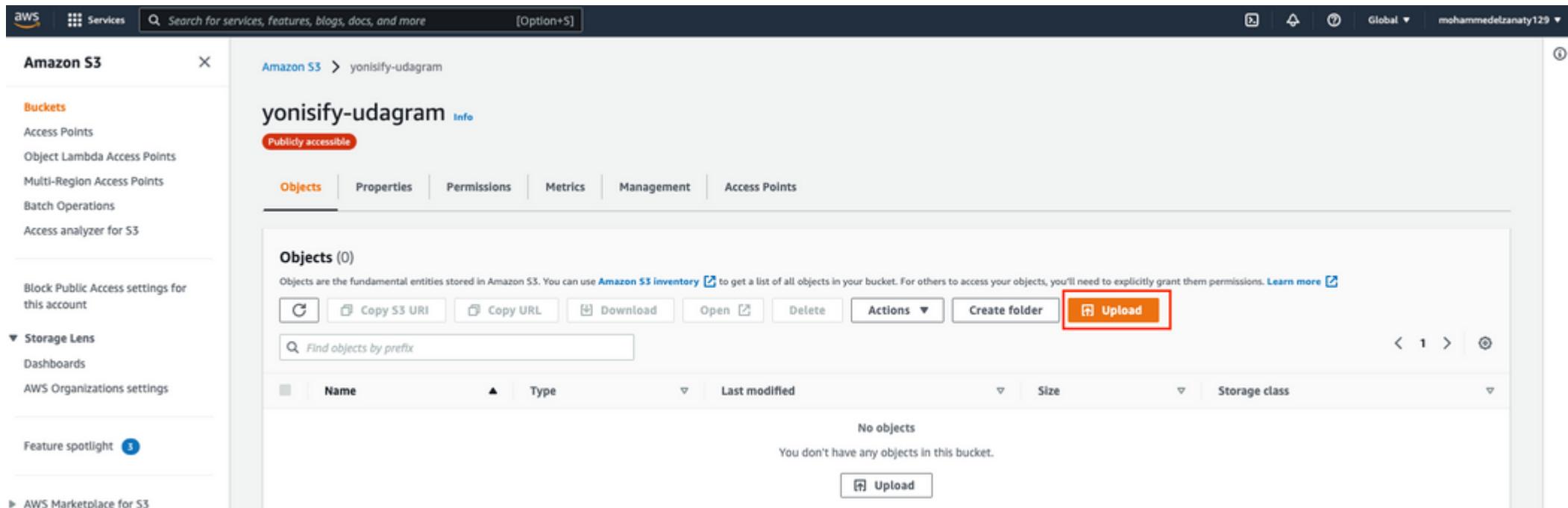
The screenshot shows the AWS S3 console. On the left, the navigation pane is visible with options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3. The main area displays a bucket named 'yonisify-udagram'. A red box highlights the 'Publicly accessible' button under the 'Properties' tab. Below it, the 'Static website hosting' section is shown with the status 'Disabled'. An 'Edit' button is highlighted with a red box.



The screenshot shows the 'Edit static website hosting' configuration page. The 'Enable' radio button is selected. Under 'Hosting type', the 'Host a static website' option is selected, with a note: 'enable website to be static require to make bucket public'. A red box highlights this note. The 'Bucket website endpoint' field contains the URL <http://yonisify-udagram.s3-website.us-east-2.amazonaws.com>. A red box highlights this URL. A note at the bottom right says 'here's the website endpoint after changing hosting type'.

Create S3 Bucket

* Upload Files:



The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various options like 'Buckets', 'Access Points', and 'Storage Lens'. The main area shows a bucket named 'yonisify-udagram' which is 'Publicly accessible'. The 'Objects' tab is selected, showing '0' objects. Below the table, it says 'No objects' and 'You don't have any objects in this bucket.' At the top of the page, there's a navigation bar with the AWS logo, a search bar, and some global settings.

* Upload Files With Terminal or Circle CI:

```
aws s3 cp --recursive --acl public-read ./www s3://yonisify-udagram/
```

Create S3 Bucket

* AWS Configurations:

```
Desktop/udagram-deploy/client 🎉 main using
❯ aws configure
AWS Access Key ID [*****TDRQ]: AKI...
AWS Secret Access Key [*****V+8H]:
```

- But First You need to create IAM user.

Create S3 Bucket

* Create IAM User:

(Create User Group)

The screenshot shows the AWS IAM dashboard. The left sidebar is collapsed, and the main area displays the 'User groups' section. A red box highlights the 'User groups' link in the sidebar. The main content area includes a 'Security recommendations' section with a note about adding MFA to the root user and a 'Root user has no active access keys' message. Below this is a table with columns for User groups, Users, Roles, Policies, and Identity providers, all showing 0. To the right is an 'AWS Account' summary with account ID, alias, sign-in URL, and an 'Add MFA' button. A 'Quick Links' section provides links to security credentials and IAM reports.

This screenshot shows the 'Create user group' step in the IAM User creation process. The left sidebar shows the 'User groups' section. A red box highlights the 'Create group' button. The main content area is titled 'Click on create-group' and shows a table with columns for Group name, Users, Permissions, and Creation time. A search bar at the top allows filtering by group name or ARN. A note states 'No resources to display'.

This screenshot shows the 'Name the group' step. The left sidebar shows the 'User groups' section. A red box highlights the 'User group name' input field. The main content area is titled 'Create user group' and contains a single input field labeled 'Name the group' with the placeholder 'User group name'. Below it is a note: 'Enter a meaningful name to identify this group.' and 'Maximum 128 characters. Use alphanumeric and '+, -, _, .' characters.'

This screenshot shows the 'Add users to the group - Optional' step. The left sidebar shows the 'User groups' section. A red box highlights the 'User groups' link. The main content area is titled 'Add users to the group - Optional (0) info'. It shows a table with columns for User name, Groups, Last activity, and Creation time. A note says 'No resources to display'. Below this is a search bar for 'User name'. A note at the bottom says 'Search for AdministratorAccess and hit create group button'.

Attach permissions policies - Optional (Selected 1/727) Info
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

| Policy name | Type | Description |
|---|----------------------------|-------------------------------|
| <input checked="" type="checkbox"/> AdministratorAccess | AWS managed - job function | Provides full access to AWS |
| <input type="checkbox"/> AdministratorAccess-Amplify | AWS managed | Grants account administrative |
| <input type="checkbox"/> AdministratorAccess-AWSMobileBackend | AWS managed | Grants account administrative |
| <input type="checkbox"/> AWSAuditManagerAdministratorAccess | AWS managed | Provides administrative acc |

Create S3 Bucket

* Create IAM User:

(Create User 01)

Screenshot of the AWS IAM Users list page. The 'User' menu is highlighted with a red box. The 'Add user' button is also highlighted with a red box.

Add user

Set user details

add name of the user User name: cli

Select AWS access type

Select programmatic access

click next

* Required

Cancel Next: Permissions

Add user

Set permissions

1. Select this

Add user to group

2. Mark admin group to get it's policies

3. Click Next

Showing 1 result

Group Attached policies

admin AdministratorAccess

Set permissions boundary

Cancel Previous Next: Tags

Create S3 Bucket

* Create IAM User:

(Create User 02)

This screenshot shows the third step of the 'Add user' wizard. It is titled 'Add tags (optional)' and includes a note about IAM tags being key-value pairs. A table allows adding up to 50 tags, with one entry shown: 'Add new key' and 'Value (optional)'. A 'Remove' button is also present. The top navigation bar shows 'aws Services' and the user 'mohammedstanzy129'.

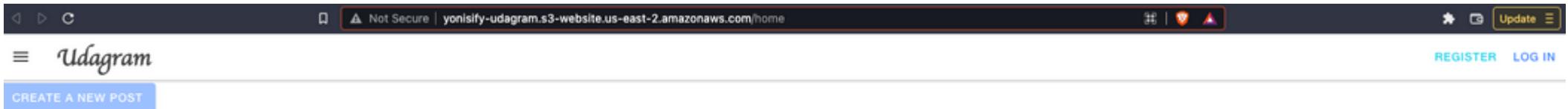
This screenshot shows the fourth step of the 'Add user' wizard, titled 'Review'. It displays the user details: 'User name: cli', 'AWS access type: Programmatic access - with an access key', and 'Permissions boundary: Permissions boundary is not set'. Below this is a 'Permissions summary' table showing a single group assignment: 'Group: admin'. The top navigation bar shows 'aws Services' and the user 'mohammedstanzy129'.

This screenshot shows the final step of the 'Add user' wizard, titled 'Success'. It confirms that the user 'cli' was successfully created and provides instructions for signing in to the AWS Management Console. It also shows the generated 'Access key ID' (AKIA4GR7JMLFR6ZJXZMF) and 'Secret access key' (hidden). A 'Download .csv' button is available to download the credentials. A note at the bottom states: 'You will not see this page again so please download .csv file which contain secret key that we will use at login from cli'. The top navigation bar shows 'aws Services' and the user 'mohammedstanzy129'.

Create S3 Bucket

* Uploading & Check Site:

```
Desktop/udagram-deploy/client 🐳 main using • v16.13.2
❯ aws s3 cp --recursive --acl public-read ./www s3://yonisify-udagram/
upload: www/16.1f758169589524f569ff.js.LICENSE.txt to s3://yonisify-udagram/16.1f758169589524f569ff.js.LICENSE.txt
upload: www/2.f5f69f369cdee4245175.js to s3://yonisify-udagram/2.f5f69f369cdee4245175.js
upload: www/17.9402a5f14a347d20862a.js to s3://yonisify-udagram/17.9402a5f14a347d20862a.js
upload: www/15.c44aea2ff4e2b6093903.js to s3://yonisify-udagram/15.c44aea2ff4e2b6093903.js
upload: www/14.a2eaa8d06cda75322318.js to s3://yonisify-udagram/14.a2eaa8d06cda75322318.js
upload: www/24.88270e18927606f4a2c0.js to s3://yonisify-udagram/24.88270e18927606f4a2c0.js
upload: www/25.178d87f4b468290d794c.js to s3://yonisify-udagram/25.178d87f4b468290d794c.js
upload: www/26.154cab7127dd0504352.js to s3://yonisify-udagram/26.154cab7127dd0504352.js
upload: www/16.1f758169589524f569ff.js to s3://yonisify-udagram/16.1f758169589524f569ff.js
```



Create Elastic Beanstalk Server

* Create Sample Environment In The Project First:

```
Desktop/udagram-deploy/server  🏡 main 🗑️ x1📝 x3📝 x1📦 v1.0.0 using • v16.13.2 took 8s
> eb init udagram-api --platform node.js --region us-east-1
Application udagram-api has been created.
```

```
Desktop/udagram-deploy/server  🏡 main 🗑️ x4📝 x1📦 v1.0.0 using • v16.13.2 took 4s
> eb create --sample udagram-api-dev
Environment details for: udagram-api-dev
  Application name: udagram-api
  Region: us-east-1
  Deployed Version: Sample Application
  Environment ID: e-9y9ap2iv8h
  Platform: arn:aws:elasticbeanstalk:us-east-1::platform/Node.js running on 64bit Amazon Linux/4.17.13
  Tier: WebServer-Standard-1.0
  CNAME: UNKNOWN
  Updated: 2022-01-24 10:07:21.257000+00:00
Alert: Your environment is using a deprecated platform branch. It might not be supported in the future.

Printing Status:
2022-01-24 10:07:19  INFO  createEnvironment is starting.
2022-01-24 10:07:21  INFO  Using elasticbeanstalk-us-east-1-838725100235 as Amazon S3 storage bucket for environment data.
2022-01-24 10:07:48  INFO  Created security group named: sg-0fdbbe234651e13168
2022-01-24 10:08:03  INFO  Created load balancer named: awseb-e-9-AWSEBLoa-10CWD0VYS9EB
2022-01-24 10:08:03  INFO  Created security group named: awseb-e-9y9ap2iv8h-stack-AWSEBSecurityGroup-GJP0B4J1LTT0
2022-01-24 10:08:03  INFO  Created Auto Scaling launch configuration named: awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingLaunchConfiguration-5Q6WV5BFWP08
2022-01-24 10:10:08  INFO  Created Auto Scaling group named: awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingGroup-1P1LK9N0473XA
2022-01-24 10:10:08  INFO  Waiting for EC2 instances to launch. This may take a few minutes.
2022-01-24 10:10:24  INFO  Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:838725100235:scalingPolicy:ff9920ed-e92b-4fa9-a6fc-d3cea2e17f98:autoScalingGroupName/awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingGroup-1P1LK9N0473XA:policyName/awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingScaleDownPolicy-NM46EJRR0000
2022-01-24 10:10:24  INFO  Created Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:838725100235:scalingPolicy:0cb3b4ad-9bc3-4f4e-8f25-342ae0e40074:autoScalingGroupName/awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingGroup-1P1LK9N0473XA:policyName/awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingScaleUpPolicy-3EQNF5YJ14JB
2022-01-24 10:10:24  INFO  Created CloudWatch alarm named: awseb-e-9y9ap2iv8h-stack-AWSEBCloudwatchAlarmLow-1QD14M2CAAI5Z
2022-01-24 10:10:24  INFO  Created CloudWatch alarm named: awseb-e-9y9ap2iv8h-stack-AWSEBCloudwatchAlarmHigh-80EW37BN3LQP
2022-01-24 10:11:18  INFO  Application available at udagram-api-dev.eba-xxprmaiui.us-east-1.elasticbeanstalk.com.
2022-01-24 10:11:19  INFO  Successfully launched environment: udagram-api-dev
```

Create Elastic Beanstalk Server

* The Sample Environment In AWS:

The screenshot shows the AWS Elastic Beanstalk Applications page. The 'Applications' tab is selected. A new application named 'udagram-api' has been created and is listed in the table. The ARN of the application is also displayed.

| Application name | Environments | Date created | Last modified | ARN |
|------------------|-----------------|------------------------------|------------------------------|---|
| udagram-api | udagram-api-dev | 2022-01-24 12:06:20 UTC+0200 | 2022-01-24 12:06:20 UTC+0200 | arn:aws:elasticbeanstalk:us-east-1:838725100235:application/udagram-api |

The screenshot shows the AWS Elastic Beanstalk Application environments page for the 'udagram-api' application. A new environment named 'udagram-api-dev' has been successfully created, as indicated by the message 'created successfully'. The URL of the application is also displayed.

| Environment name | Health | Date created | Last modified | URL | Running versions | Platform | Platform state | Tier name |
|------------------|-----------------|------------------------------|------------------------------|---|--------------------|---------------------------------------|----------------|-----------|
| udagram-api-dev | Ok | 2022-01-24 12:07:21 UTC+0200 | 2022-01-24 12:11:19 UTC+0200 | udagram-api-dev.eba-xoprmlau.us-east-1.elasticbeanstalk.com | Sample Application | Node.js running on 64bit Amazon Linux | - | WebServer |

click here

Create Elastic Beanstalk Server

* The Sample Environment In AWS:

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, a sidebar lists environments, applications, and configuration options for the 'udagram-api' application, with 'udagram-api-dev' selected. The main area displays the 'udagram-api-dev' environment details. A red box highlights the URL 'udagram-api-dev.eba-xxprmalu.us-east-1.elasticbeanstalk.com' and the application name 'udagram-api'. Another red box highlights the 'Health environment status' section, which shows a green checkmark and the word 'Ok'. A third red box highlights the 'Logs' tab under the environment configuration. The 'Recent events' section at the bottom also has a red box around it, showing a list of log entries. The top navigation bar includes the AWS logo, services menu, search bar, and user information.

from the url you can access the environment

udagram-api-dev
udagram-api-dev.eba-xxprmalu.us-east-1.elasticbeanstalk.com (e-9y9ap2iv8h)
Application name: udagram-api

Health environment status
Ok

Configuration Logs Health Monitoring Alarms Managed updates Events Tags

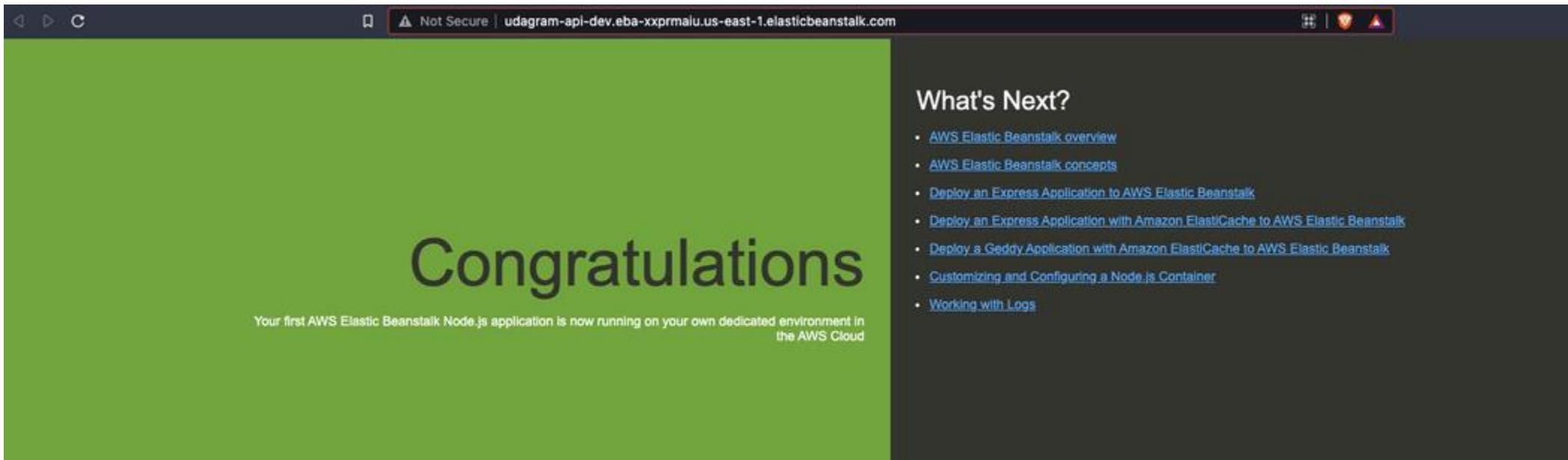
Recent events

this section has the latest activity and it will help you a lot to if anything wrong happen

| Time | Type | Details |
|------------------------------|------|---|
| 2022-01-24 12:11:25 UTC+0200 | INFO | Environment health has transitioned from Pending to Ok. Initialization completed 35 seconds ago and took 3 minutes. |
| 2022-01-24 12:11:19 UTC+0200 | INFO | Successfully launched environment: udagram-api-dev |
| 2022-01-24 12:11:18 UTC+0200 | INFO | Application available at udagram-api-dev.eba-xxprmalu.us-east-1.elasticbeanstalk.com. |
| 2022-01-24 12:10:25 UTC+0200 | INFO | Added instance [i-0d0c72916e4f0b430] to your environment. |
| 2022-01-24 12:10:24 UTC+0200 | INFO | Created CloudWatch alarm named: awseb-e-9y9ap2iv8h-stack-AWSEBCloudwatchAlarmHigh-8OEW378N3LQP |

Create Elastic Beanstalk Server

* The Sample Environment In AWS is Working:



Create Elastic Beanstalk Server

* Deploy Server To EB:

1. Use the environment in the project

```
eb use udagram-api-dev
```

2. Build Backend Files

```
Desktop/udagram-deploy/server  🏠 main ✨ x4 🎨 x1 📦 v1.0.0 using • v16.13.2
✖ yarn build
yarn run v1.22.17
$ npm run clean && tsc && cp -rf src/config www/config && cp .npmrc www/.npmrc && cp package.json www/package.json && cd www && zip -r Archive.zip .
&& cd ..

> udagram-server@1.0.0 clean
> rm -rf www || true
```

3. Specify The Archive File in Config.yml in Elastic Beanstalk Folder

```
5  deploy:
6    artifact: www/Archive.zip
```

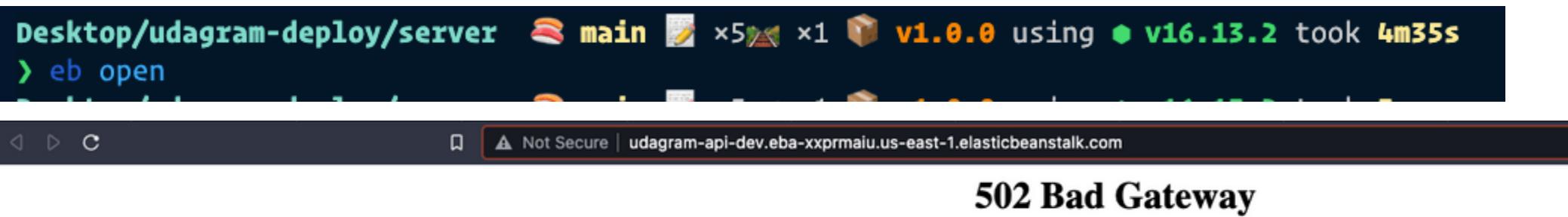
Create Elastic Beanstalk Server

4. Deploy

```
Desktop/udagram-deploy/server  🏫 main  📄 x5  🏠 x1  📦 v1.0.0 using  ● v16.13.2 took 8s
> eb deploy udagram-api-dev
Alert: Your environment is using a deprecated platform branch. It might not be supported in the future.

Creating application version archive "app-220124_122332".
Uploading: [########################################] 100% Done...
2022-01-24 10:25:59    INFO    Environment update is starting.
2022-01-24 10:26:41    INFO    Deploying new version to instance(s).
2022-01-24 10:28:04    INFO    New application version was deployed to running EC2 instances.
-- Events -- (safe to Ctrl+C) Use "eb abort" to cancel the com2022-01-24 10:28:04    INFO    Environment update completed successfully
.
-- Events -- (safe to Ctrl+C) Use "eb abort" to cancel the com
```

5. Open



The screenshot shows a terminal window at the top with the command 'eb open' and its output. Below it is a browser window. The browser's address bar shows a URL starting with 'Not Secure | udagram-api-dev.eba-xxprmai...'. The main content area of the browser displays a large '502 Bad Gateway' error message.

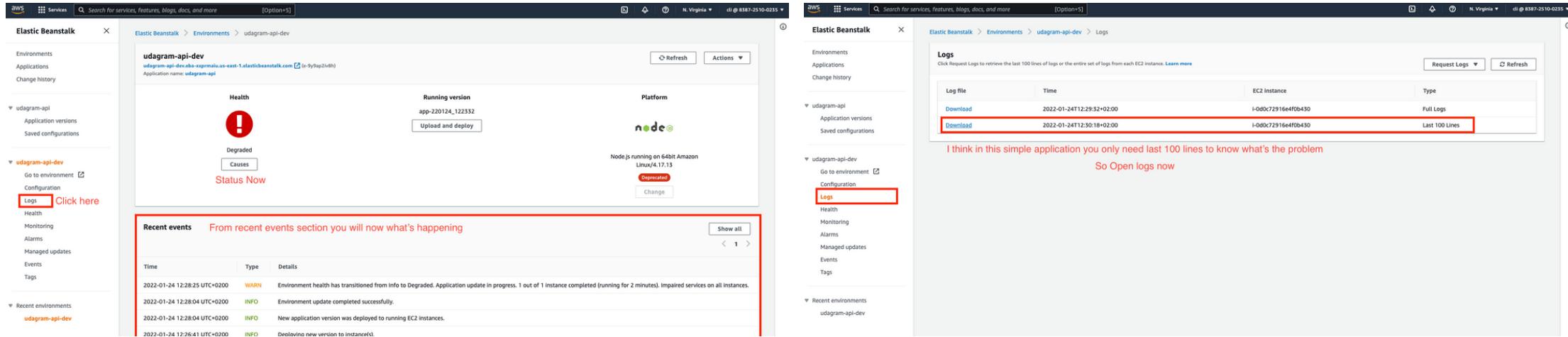
```
Desktop/udagram-deploy/server  🏫 main  📄 x5  🏠 x1  📦 v1.0.0 using  ● v16.13.2 took 4m35s
> eb open
```

Not Secure | udagram-api-dev.eba-xxprmai...us-east-1.elasticbeanstalk.com

502 Bad Gateway

Create Elastic Beanstalk Server

* Check Errors:



The image shows two side-by-side screenshots of the AWS Elastic Beanstalk console. Both screenshots are for the environment 'udagram-api-dev'.
The left screenshot shows the 'Logs' section under the 'Logs' tab. It displays a table of recent events with columns for Time, Type, and Details. One event is highlighted with a red border:

| Time | Type | Details |
|------------------------------|------|---|
| 2022-01-24 12:28:25 UTC+0200 | WARN | Environment health has transitioned from Info to Degraded. Application update in progress. 1 out of 1 instance completed (running for 2 minutes). Impaired services on all instances. |
| 2022-01-24 12:28:04 UTC+0200 | INFO | Environment update completed successfully. |
| 2022-01-24 12:28:04 UTC+0200 | INFO | New application version was deployed to running EC2 instances. |
| 2022-01-24 12:26:41 UTC+0200 | INFO | Deploying new version to instance(s). |

The right screenshot shows the 'Logs' section under the 'Logs' tab. It displays a table of log files with columns for Log file, Time, EC2 Instance, and Type. Two log entries are highlighted with a red border:

| Log file | Time | EC2 Instance | Type |
|----------|---------------------------|---------------------|----------------|
| Download | 2022-01-24T12:29:32+02:00 | I-0d0c72916e4f0b430 | Full Logs |
| Download | 2022-01-24T12:30:18+02:00 | I-0d0c72916e4f0b430 | Last 100 Lines |

* Set Environment Variables:

```
Desktop/udagram-deploy/server ➜ eb setenv PORT=3000
2022-01-24 10:31:55      INFO  Environment update is starting.
2022-01-24 10:32:05      INFO  Updating environment udagram-api-dev's configuration settings.
2022-01-24 10:33:18      INFO  Successfully deployed new configuration to environment.
```

Create Elastic Beanstalk Server

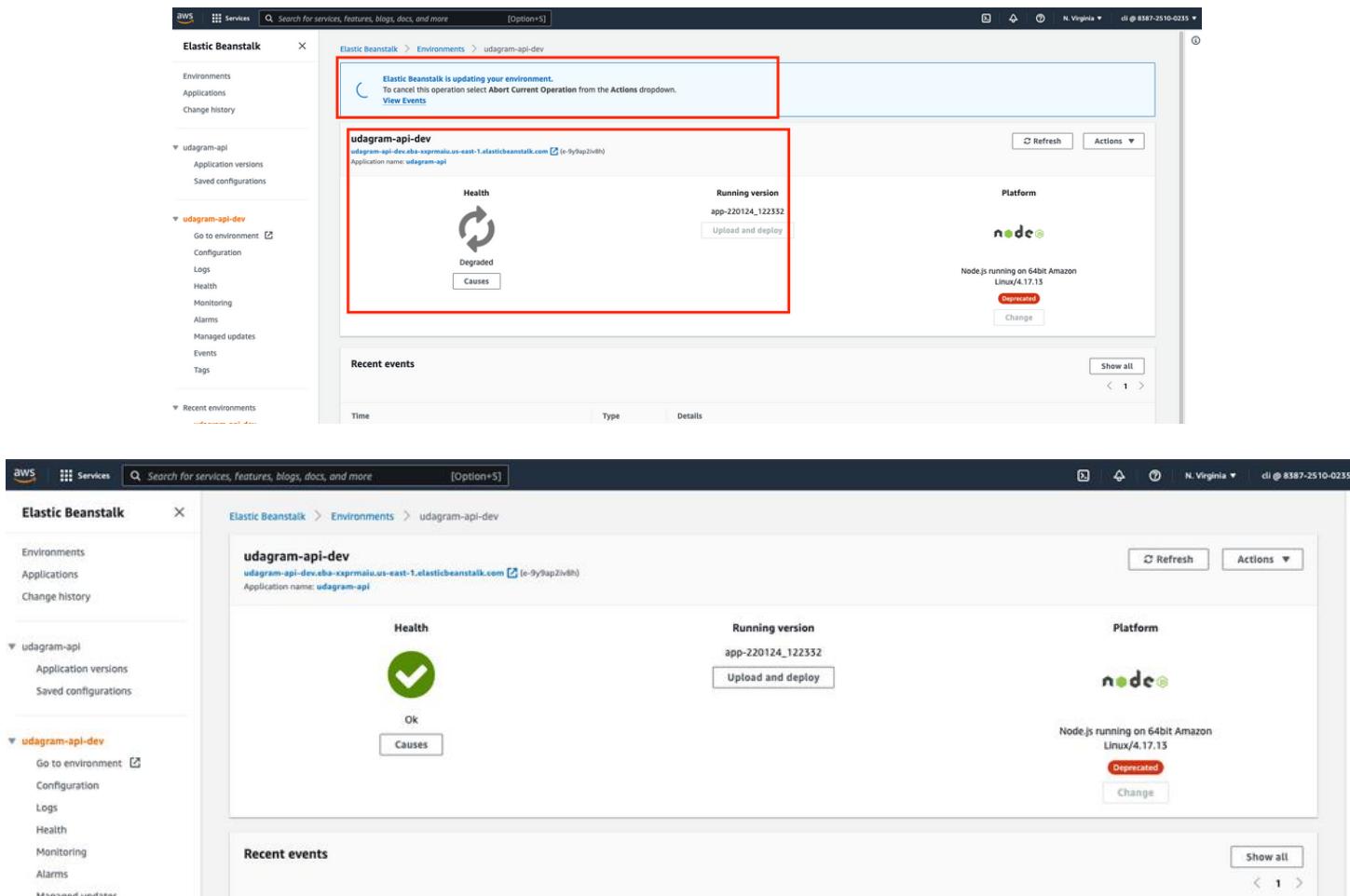
* Then Build Again & Deploy Again:

```
Desktop/udagram-deploy/server  main  x12  x5  x2  v1.0.0 using v16.13.2
> eb deploy udagram-api-dev
Alert: Your environment is using a deprecated platform branch. It might not be supported in the future.

Uploading udagram-api/app-220124_125413.zip to S3. This may take a while.
Upload Complete.
2022-01-24 10:54:17  INFO    Environment update is starting.
2022-01-24 10:54:21  INFO    Deploying new version to instance(s).
2022-01-24 10:54:56  INFO    New application version was deployed to running EC2 instances.
2022-01-24 10:54:56  INFO    Environment update completed successfully.
```

Create Elastic Beanstalk Server

* Refresh The Environment Automatically After Upload:



Create Elastic Beanstalk Server

* If You are Done >> Terminate Environment:

```
Desktop/udagram-deploy/server 🐀 main 🐀 ×12 🐀 ×5 🐀 ×2 🐀 v1.0.0 using 🐀 v16.13.2 took 2s
❯ eb terminate
The environment "udagram-api-dev" and all associated instances will be terminated.
To confirm, type the environment name: udagram-api-dev
2022-01-24 10:56:06 INFO  terminateEnvironment is starting.
2022-01-24 10:56:24 INFO  Deleted CloudWatch alarm named: awseb-e-9y9ap2iv8h-stack-AWSEBCloudwatchAlarmLow-10D14M2CAAI5Z
2022-01-24 10:56:24 INFO  Deleted CloudWatch alarm named: awseb-e-9y9ap2iv8h-stack-AWSEBCloudwatchAlarmHigh-80EW37BN3LQP
2022-01-24 10:56:24 INFO  Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:838725100235:scalingPolicy:6cb3b4
ad-9bc3-4f4e-8f25-342ae0e40074:autoScalingGroupName:awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingGroup-1P1LK9N0473XA:policyName:awseb-e-9y9
ap2iv8h-stack-AWSEBAutoScalingScaleUpPolicy-3EQNFSYJ14JB
2022-01-24 10:56:24 INFO  Deleted Auto Scaling group policy named: arn:aws:autoscaling:us-east-1:838725100235:scalingPolicy:ff9920
ed-e92b-4fa9-a6fc-d3cea2e17f98:autoScalingGroupName:awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingGroup-1P1LK9N0473XA:policyName:awseb-e-9y9
ap2iv8h-stack-AWSEBAutoScalingScaleDownPolicy-NM46EJRR00Q
2022-01-24 10:56:24 INFO  Waiting for EC2 instances to terminate. This may take a few minutes.
2022-01-24 11:00:28 INFO  Deleted Auto Scaling group named: awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingGroup-1P1LK9N0473XA
2022-01-24 11:00:29 INFO  Deleted load balancer named: awseb-e-9-AWSEBLba-10CHGDOVV59EB
2022-01-24 11:00:44 INFO  Deleted Auto Scaling launch configuration named: awseb-e-9y9ap2iv8h-stack-AWSEBAutoScalingLaunchConfigur
ation-506WVSBFWP88
2022-01-24 11:00:44 INFO  Deleted security group named: awseb-e-9y9ap2iv8h-stack-AWSEBSecurityGroup-GJP0B4J1LTTO
2022-01-24 11:01:15 INFO  Deleted security group named: sg-0fdb2e3d651e13168
2022-01-24 11:01:16 INFO  Deleting SNS topic for environment udagram-api-dev.
2022-01-24 11:01:18 INFO  terminateEnvironment completed successfully.
```

The screenshot shows the AWS Elastic Beanstalk console. At the top, there's a search bar and a navigation bar with 'Services' selected. Below that, the 'Environments' section is open, showing a warning message: 'Deprecated/Retired Platform(s)'. It says 'One or more of your environments is configured with a deprecated or retired platform.' with a 'Learn more' link. The main table, titled 'All environments', has columns for Environment name, Health, Application name, Date created, Last modified, URL, Running versions, Platform, Platform state, and Tier name. One environment, 'udagram-api-dev (terminated)', is listed. The 'Platform state' column shows 'Deprecated' in red for this row. The 'Tier name' column shows 'WebServer'.

| Environment name | Health | Application name | Date created | Last modified | URL | Running versions | Platform | Platform state | Tier name |
|------------------------------|--------|------------------|------------------------------|------------------------------|--|-------------------|---------------------------------------|----------------|-----------|
| udagram-api-dev (terminated) | - | udagram-api | 2022-01-24 12:07:21 UTC+0200 | 2022-01-24 13:01:17 UTC+0200 | udagram-api-dev.eba-xxprmlu.us-east-1.elasticbeanstalk.com | app-220124_125413 | Node.js running on 64bit Amazon Linux | Deprecated | WebServer |

APP DEPENDENCIES

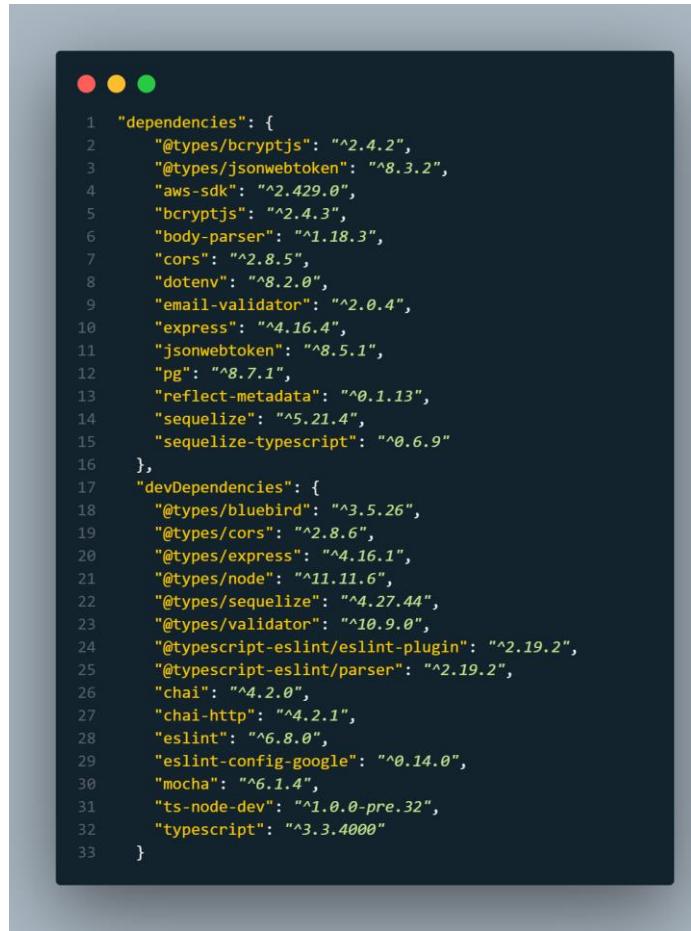
APP DEPENDENCIES

* Front End



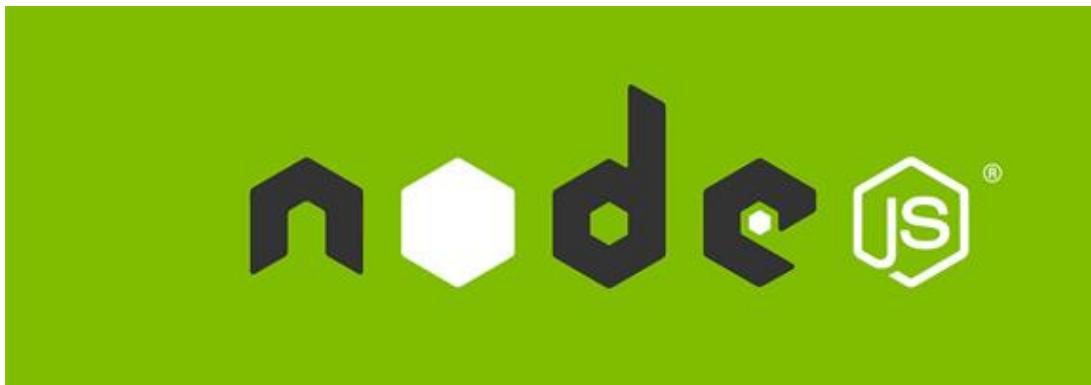
```
1 "dependencies": {
2   "@angular/common": "~8.2.14",
3   "@angular/core": "~8.2.14",
4   "@angular/forms": "~8.2.14",
5   "@angular/http": "~7.2.2",
6   "@angular/platform-browser": "~8.2.14",
7   "@angular/platform-browser-dynamic": "~8.2.14",
8   "@angular/router": "~8.2.14",
9   "@ionic-native/core": "~5.0.0",
10  "@ionic-native/splash-screen": "~5.0.0",
11  "@ionic-native/status-bar": "~5.0.0",
12  "@ionic/angular": "~4.1.0",
13  "core-js": "~2.5.4",
14  "rxjs": "~6.5.4",
15  "zone.js": "~0.9.1"
16 },
17 "devDependencies": {
18   "@angular-devkit/architect": "~0.12.3",
19   "@angular-devkit/build-angular": "~0.803.24",
20   "@angular-devkit/core": "~7.2.3",
21   "@angular-devkit/schematics": "~7.2.3",
22   "@angular/cli": "~8.3.25",
23   "@angular/compiler": "~8.2.14",
24   "@angular/compiler-cli": "~8.2.14",
25   "@angular/language-service": "~8.2.14",
26   "@ionic/angular-toolkit": "~1.4.0",
27   "@types/jasmine": "~2.8.8",
28   "@types/jasminewd2": "~2.0.3",
29   "@types/node": "~10.12.0",
30   "@typescript-eslint/eslint-plugin": "~2.20.0",
31   "@typescript-eslint/parser": "~2.20.0",
32   "codelyzer": "~4.5.0",
33   "jasmine-core": "~2.99.1",
34   "jasmine-spec-reporter": "~4.2.1",
35   "karma": "~3.1.4",
36   "karma-chrome-launcher": "~2.2.0",
37   "karma-coverage-istanbul-reporter": "~2.0.1",
38   "karma-jasmine": "~1.1.2",
39   "karma-jasmine-html-reporter": "~0.2.2",
40   "protractor": "~5.4.0",
41   "ts-node": "~8.0.0",
42   "tslint": "~5.12.0",
43   "typescript": "~3.5.3"
44 },
```

* Back End



```
1 "dependencies": {
2   "@types/bcryptjs": "^2.4.2",
3   "@types/jsonwebtoken": "^8.3.2",
4   "aws-sdk": "^2.429.0",
5   "bcryptjs": "^2.4.3",
6   "body-parser": "^1.18.3",
7   "cors": "^2.8.5",
8   "dotenv": "^8.2.0",
9   "email-validator": "^2.0.4",
10  "express": "^4.16.4",
11  "jsonwebtoken": "^8.5.1",
12  "pg": "^8.7.1",
13  "reflect-metadata": "^0.1.13",
14  "sequelize": "^5.21.4",
15  "sequelize-typescript": "^0.6.9"
16 },
17 "devDependencies": {
18   "@types/bluebird": "^3.5.26",
19   "@types/cors": "^2.8.6",
20   "@types/express": "^4.16.1",
21   "@types/node": "^11.11.6",
22   "@types/sequelize": "^4.27.44",
23   "@types/validator": "^10.9.0",
24   "@typescript-eslint/eslint-plugin": "^2.19.2",
25   "@typescript-eslint/parser": "^2.19.2",
26   "chai": "4.2.0",
27   "chai-http": "4.2.1",
28   "eslint": "6.8.0",
29   "eslint-config-google": "0.14.0",
30   "mocha": "6.1.4",
31   "ts-node-dev": "1.0.0-pre.32",
32   "typescript": "3.3.4000"
33 }
```

APP DEPENDENCIES



PIPELINE PROCESS

PIPELINE PROCESS

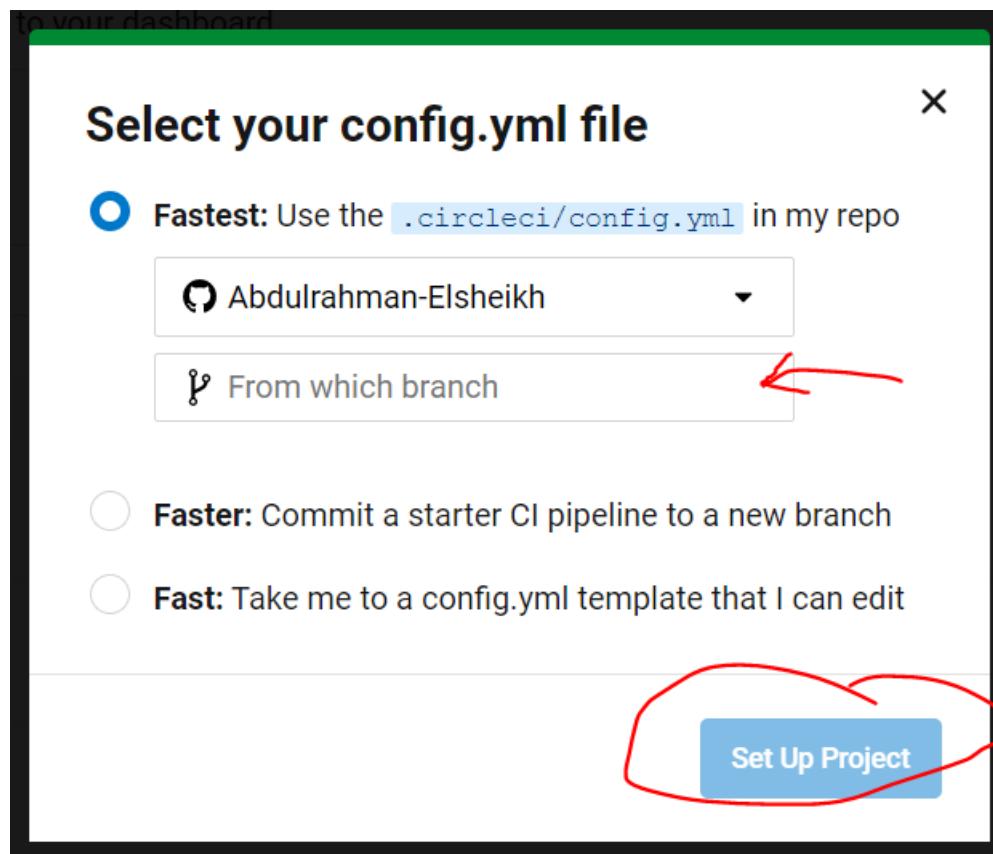
* Create Account on Circle CI and Connect it with your GitHub Account:

The screenshot shows the CircleCI dashboard interface. On the left is a dark sidebar with navigation links: Dashboard, Projects (which is selected and highlighted in grey), Insights, Organization Settings, Plan, and a search bar for 'Can't find an organization?'. Below these are Notifications (1), Status (OPERATIONAL), Docs, and Orbs. The main area is titled 'Projects' and contains a search bar for 'Repo name'. A modal window at the top right provides instructions: 'Set up new projects or follow projects already building on CircleCI.' and 'Following a project adds it to your dashboard.' Below the modal is a list of repositories: 'Hosting-Full-Stack-Application', 'Deploy-Fullstack-Website', 'Abdulrahman-Elsheikh', 'StoreFront-Backend', 'resizing-images-api', 'weather-journal-app', and 'Landing-Page'. To the right of each repository are three buttons: 'Unfollow Project', 'Follow All', and a three-dot menu icon. The 'Deploy-Fullstack-Website' row has its 'Set Up Project' button circled in red.

| Repository | Action | More Options |
|--------------------------------|------------------|--------------|
| Hosting-Full-Stack-Application | Unfollow Project | ... |
| Deploy-Fullstack-Website | Unfollow Project | ... |
| Abdulrahman-Elsheikh | Set Up Project | ... |
| StoreFront-Backend | Set Up Project | ... |
| resizing-images-api | Set Up Project | ... |
| weather-journal-app | Set Up Project | ... |
| Landing-Page | Set Up Project | ... |

PIPELINE PROCESS

* Specify The Config.yml File From Which Branch:



PIPELINE PROCESS

* Project Dashboard:

The screenshot shows the CircleCI Project Dashboard for the 'Hosting-Full-Stack-Application' project. The dashboard features a dark sidebar on the left with navigation links: Dashboard, Projects, Insights, Organization Settings, Plan (with an Upgrade button), and a section for finding organizations. The main area displays the project name 'Hosting-Full-Stack-Application' with an 'Add team members' link. It includes filters for 'Everyone's Pipelines', 'Hosting-Full-Stack-Application', 'All Branches', and a dropdown for 'Auto-expand'. A table lists the pipeline runs, showing one successful run on the 'main' branch started 24m ago, which took 4m 26s. The table columns are Pipeline, Status, Workflow, Branch / Commit, Start, Duration, and Actions. The 'Actions' column contains icons for viewing logs, triggering a new build, and deleting the pipeline. Below the table, there are sections for 'Jobs' and 'Builds', with 8 builds listed.

| Pipeline | Status | Workflow | Branch / Commit | Start | Duration | Actions |
|--------------------------------|----------------------|----------|--|---------|----------|---|
| Hosting-Full-Stack-Application | Success | workflow |  main c73e50b Added Circle CI Pipeline Diagram | 24m ago | 4m 26s |    |
| | | | | | 4m 24s | |

PIPELINE PROCESS

* Setup Environment Variables:

The screenshot shows the CircleCI web interface for managing pipelines. At the top, there's a header with the pipeline name "Hosting-Full-Stack-Application", an "Add team members" button, and three action buttons: "Edit Config", "Trigger Pipeline", and "Project Settings" (the last one is circled in red). Below the header are several filter options: "Filters", "Everyone's Pipelines" dropdown, "Hosting-Full-Stack-Ap..." dropdown, "All Branches" dropdown, a search icon, and an "Auto-expand" toggle switch. The main area displays a table with columns: Pipeline, Status, Workflow, Branch / Commit, Start, Duration, and Actions. A single row is shown for the "Hosting-Full-Stack-Application" pipeline, which is in "Success" status, triggered by a "workflow" job, and started 25m ago. The commit information is "main c73e50b Added Circle CI Pipeline Diagram". The "Actions" column contains icons for viewing logs, cloning the repository, and deleting the pipeline. Below the table, there's a summary section for the "Jobs" step, showing 8 builds completed in 4m 24s. At the bottom, there are navigation links for "Creating Full", "Workflow", "Main", and "Logs".

| Pipeline | Status | Workflow | Branch / Commit | Start | Duration | Actions |
|--------------------------------|------------------------|----------|--|---------|----------|---|
| Hosting-Full-Stack-Application | ✓ Success | workflow |  main c73e50b Added Circle CI Pipeline Diagram | 25m ago | 4m 26s |    ... |

Jobs

✓ build 8

4m 24s

PIPELINE PROCESS

* Setup Environment Variables:

The screenshot shows a sidebar with various pipeline settings and a main content area titled 'Environment Variables'. The sidebar includes options like Overview, Triggers, Advanced, Environment Variables (which is highlighted with a red oval), SSH Keys, API Permissions, Jira Integration, Slack Integration, Status Badges, and Webhooks. The main content area has a descriptive paragraph about environment variables and a table listing three variables: AWS_ACCESS_KEY_ID, AWS_DEFAULT_REGION, and AWS_SECRET_ACCESS_KEY. A blue 'Add Environment Variable' button is also highlighted with a red oval.

Overview

Triggers

Advanced

Environment Variables

SSH Keys

API Permissions

Jira Integration

Slack Integration

Status Badges

Webhooks

Environment Variables

Environment variables let you add sensitive data (e.g. API keys) to your jobs rather than placing them in the repository. The value of the variables cannot be read or edited in the app once they are set.

If you're looking to share environment variables across projects, try [Contexts](#).

| Name | Value | X |
|-----------------------|----------|---|
| AWS_ACCESS_KEY_ID | xxxx6IGM | X |
| AWS_DEFAULT_REGION | xxxxst-1 | X |
| AWS_SECRET_ACCESS_KEY | xxxxbqSS | X |

Add Environment Variable

Import Variables

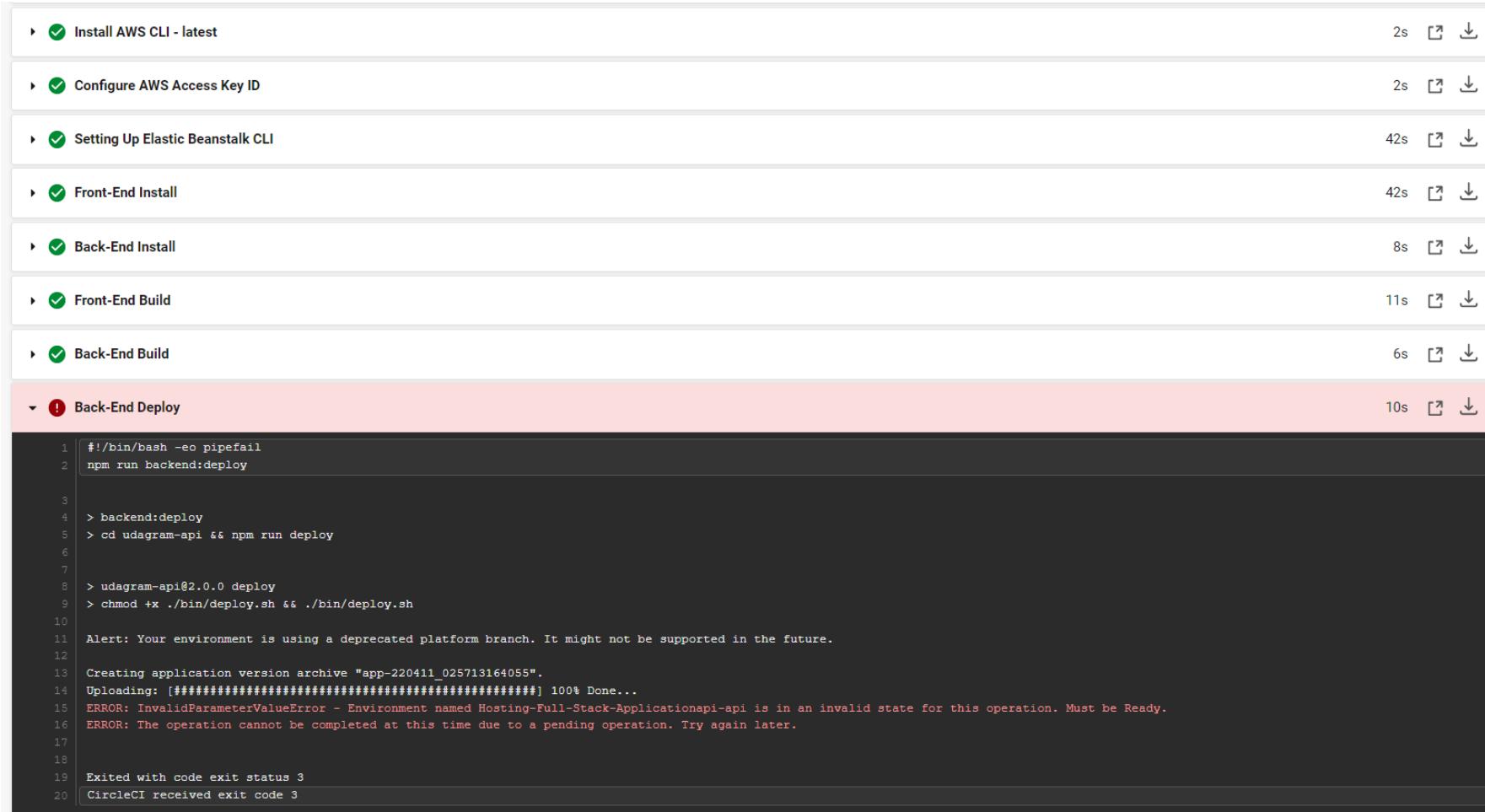
PIPELINE PROCESS

* Deployment Process Status:

| Pipeline | Status | Workflow | Branch / Commit | Start | Duration | Actions |
|-------------------------------------|---------|----------|--|---------------------------------------|----------|---------|
| Hosting-Full-Stack-Application 7 | Success | workflow | main c73e50b Added Circle CI Pipeline Diagram | 27m ago | 4m 26s | ... |
| | Jobs | build 8 | | | 4m 24s | |
| Hosting-Full-Stack-Application 6 | Success | workflow | main 002b779 Added Project Diagram | 53m ago | 4m 43s | ... |
| | Jobs | build 7 | | | 4m 41s | |
| | | Failed | workflow | main 002b779 Added Project Diagram | 59m ago | 2m 20s |
| | Jobs | build 6 | | | 2m 17s | |

PIPELINE PROCESS

* Error in Deployment Process:



The screenshot shows a CI pipeline interface with a list of steps and a detailed log for the final step.

| Step | Description | Duration | Icon |
|------|----------------------------------|----------|---------|
| 1 | Install AWS CLI - latest | 2s | Success |
| 2 | Configure AWS Access Key ID | 2s | Success |
| 3 | Setting Up Elastic Beanstalk CLI | 42s | Success |
| 4 | Front-End Install | 42s | Success |
| 5 | Back-End Install | 8s | Success |
| 6 | Front-End Build | 11s | Success |
| 7 | Back-End Build | 6s | Success |
| 8 | Back-End Deploy | 10s | Error |

Back-End Deploy Step Details:

```
#!/bin/bash -eo pipefail
npm run backend:deploy

> backend:deploy
> cd udagram-api && npm run deploy

> udagram-api@2.0.0 deploy
> chmod +x ./bin/deploy.sh && ./bin/deploy.sh

Alert: Your environment is using a deprecated platform branch. It might not be supported in the future.

Creating application version archive "app-220411_025713164055".
Uploading: [########################################] 100% Done...
ERROR: InvalidParameterValueError - Environment named Hosting-Full-Stack-Applicationapi-api is in an invalid state for this operation. Must be Ready.
ERROR: The operation cannot be completed at this time due to a pending operation. Try again later.

Exited with code exit status 3
CircleCI received exit code 3
```

PIPELINE PROCESS

* Deployment Process Succeed:

| | | | |
|--------------------------------------|-------|--|--|
| ▶ ✓ Spin up environment | 1s | | |
| ▶ ✓ Preparing environment variables | 0s | | |
| ▶ ✓ Install Node.js 16.13 | 3s | | |
| ▶ ✓ Install NPM | 3s | | |
| ▶ ✓ Checkout code | 0s | | |
| ▶ ✓ Install AWS CLI - latest | 2s | | |
| ▶ ✓ Configure AWS Access Key ID | 2s | | |
| ▶ ✓ Setting Up Elastic Beanstalk CLI | 37s | | |
| ▶ ✓ Front-End Install | 37s | | |
| ▶ ✓ Back-End Install | 7s | | |
| ▶ ✓ Front-End Build | 10s | | |
| ▶ ✓ Back-End Build | 6s | | |
| ▶ ✓ Back-End Deploy | 2m 6s | | |
| ▶ ✓ Front-End Deploy | 23s | | |