BadCats🐱 - Week 9

THEORETICAL

What are the benefits of using Elastic Beanstalk over just EC2

PRACTICAL

Configure a Typescript App with the correct EB settings and build/start scripts

[1] Shimaa Adel Mohammed Elzarief

[2] Burham BadrElden Burham Soliman

[3] Mohammed Ahmed Hassen

[4] Shaymaa Hafez Ebrahiem

[5] yousef alaa adel

Shimaa Adel Elzarief

Linkedin / Github

- THEORETICAL -

Below are some benefits that AWS Elastic Beanstalk offers over other PaaS services

Offers Quicker Deployment:

Elastic Beanstalk offers developers the fastest and simplest way to deploy their application. Within minutes, the application will be ready to use without users having to deal with the underlying infrastructure or resource configuration.

Supports Multi-Tenant Architecture:

- THEORETICAL -

AWS Elastic Beanstalk makes it possible for users to share their applications across different devices with high scalability and security. It provides a detailed report of application usage and user profiles.

Simplifies Operations:

Beanstalk provisions and operates the infrastructure and manages the application stack. Developers have to just focus on developing code for their application rather than spending time on managing and

configuring servers, databases, firewalls, and networks.

Offers Complete Resource Control:

Beanstalk gives developers the freedom to select the AWS resources, like *EC2 instance* type, that are optimal for their application. It allows developers to retain full control over AWS resources and access them at any time.

- THEORETICAL -
- PRACTICAL -

Burham B. Soliman

[github@egstar | LinkedIn@Burham]

- THEORETICAL -

- Elastic Beanstalk is one layer of abstraction away from the EC2 layer.
- Elastic Beanstalk will setup an "environment" for you that can contain a number of EC2 instances, an optional database, as well as a few other AWS components such as:
  - Elastic Load Balancer
  - Auto-Scaling Group
  - Security Group.
- Then Elastic Beanstalk will manage these items for you whenever you want to update your software running in AWS.
- Elastic Beanstalk doesn't add any cost on top of these resources

By another meaning, you save the time at the dev process, so you don't have to create or install the basic packages for running your app, you just give the EB the needed information to let it start perfectly as you were going to do exactly.

- PRACTICAL -

1) Initializing and Configuring the EB:

```
    * Make sure to have the AWS configuration done using the command [ aws configure
]
      using the right IAM user with valid Admin privileges
    * Run the command [ eb init ] and follow the script
      or use eb init APP_NAME --platform Node.js --region us-state-1
    * Create the the eb instance with command [ eb create ] and follow the script
      or use eb create --sample ENV_NAME
```

2) Set the package.json script:

```
    * Set the start script "start": "node ./SERVERFILE.js"
    * Set the build script "build": "npm i . && rm -rf ./DIST_DIR/ && tsc && cp -R
.elasticbeanstalk DIST_DIR/.elasticbeanstalk && cp .npmrc DIST_DIR/.npmrc && cp
package.json DIST_DIR/package.json && zip -r DIST_DIR/FILE.zip ./DIST_DIR"
    * Set the deploy script "deploy": "npm run build && eb use ENV_NAME && eb deploy
&& eb setenv VAR_NAME1=$VAR_NAME1 VAR_NAME2=$VAR_NAME2 VAR_NAME3=$VAR_NAME3"
```

3) Identify the deploy target into the EB config file [.elasticbeanstalk/config.yml]

```
    * add the deploy command:
        deploy:
            artifact: DIST_DIR/FILE.zip
```

4) Create the .npmrc file to allow node_gyp creating files/folders:

```
    * allow unsafe permissions using this line  unsafe-perm=true
```

5) Build and run deploy:

```
npm run deploy
```

Mohammed Ahmed Hassan
mohamedlolx (Mohamed Hassan)(github.com)
Mohamed Hassan | LinkedIn

- THEORETICAL -

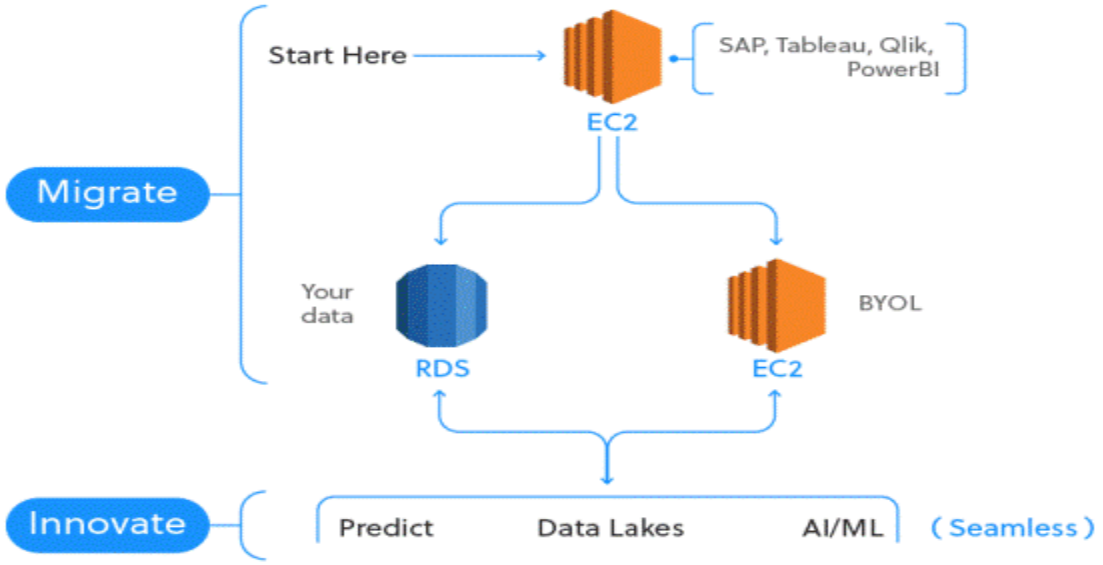
Elastic Beanstalk Vs EC2

Regardless of their similarities, there are significant contrasts between them, as well,

- The Elastic Beanstalk is essentially 1 layer of reflection away from that of the EC2 layer.
- At a point when you work with Beanstalk's service, all the backend servers will be EC2 instances, and they will be getting their configuration behind a load balancer that will open them to the outer universe.
- Truly, a Beanstalk arranged framework conceals several matters, as it sets up a situation for you that contains EC2 instances, security scaling groups, databases, and so forth.

Differences between the AWS Services of Elastic Beanstalk and EC2:

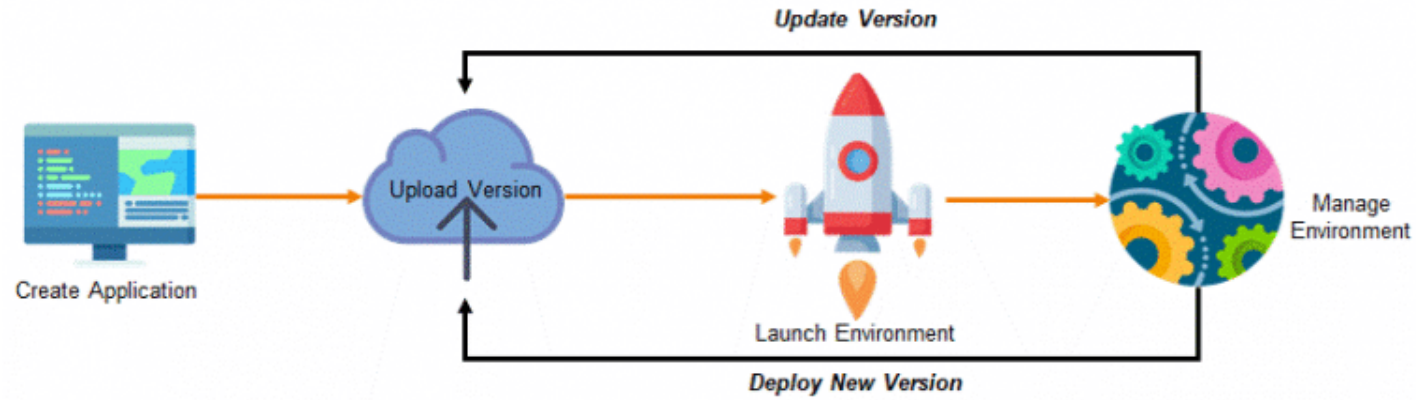
How does the EC2 actually work Vs Elastic Beanstalk?



EC2 allows users to create and launch servers in the cloud. The EC2 instances offer a total web services API for accessing the many different services that are available on AWS platform.

How does the Elastic Beanstalk actually work Vs EC2?

- THEORETICAL -

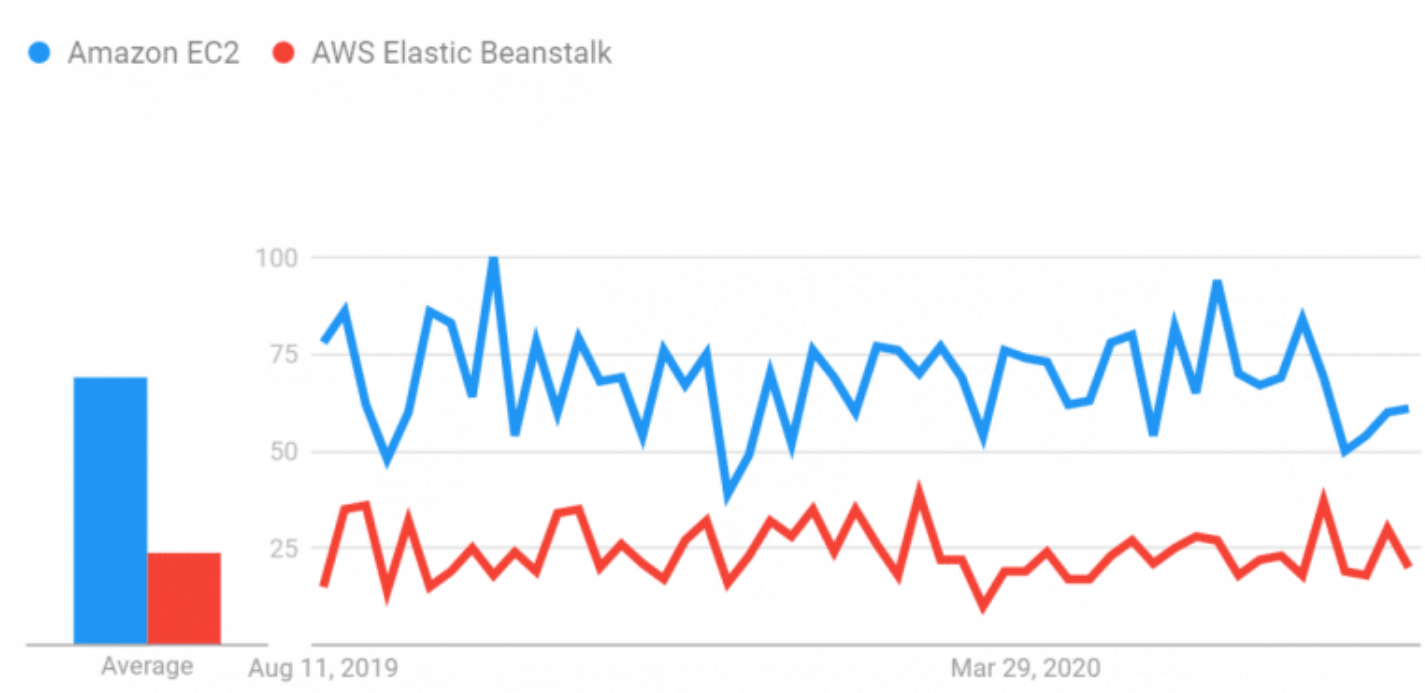


- The Elastic Beanstalk service from AWS provides developers with a platform for the deployment of apps on the AWS cloud, as well as gets them connected to other AWS services.
- This means that Elastic Beanstalk cannot be regarded as something which could be discovered as time goes by, but on the contrary it should be studied in terms of its underlying AWS services, and the way they work in concert through the help of Elastic Beanstalk.
- Elastic Beanstalk connects services such as S3, EC2 and Auto Scaling in order to deploy elastic cloud apps.
- When an environment gets launched, Elastic Beanstalk will merely use an already defined AMI which is accompanied with an operating system that is installed, then goes ahead with launching a newly created instance having the same type of your specification.
- Additionally, Beanstalk would set up an elastic load balancer making it respond to a unique URL.
- Due to the fact that Elastic Beanstalk would **orchestrate** various services, you will find extra procedures of interaction with those services.

Regardless of the fact that everything would be organized and verified by default, you will get the possibility to actually work with and change the assets managed by Elastic Beanstalk, where you've got the opportunity to either overwrite, adjust or bypass whatever Elastic Beanstalk performs. You can also get things customized based on what you require.

Comparison of Interest over time for Elastic Beanstalk Vs EC2:

- THEORETICAL -



As it can be concluded from the above graph, Amazon EC2 has a higher interest over time while

AWS Elastic Beanstalk has a low interest over time rate. This means that with the use of Amazon EC2 you can get a higher rate of interest as time goes by, while with AWS Elastic Beanstalk this rate will be way less.

- THEORETICAL -

Advantages and Disadvantages of AWS Elastic Beanstalk Vs EC2:

Check out in the table below some of the major advantages and disadvantages of AWS Beanstalk and those of Amazon EC2.

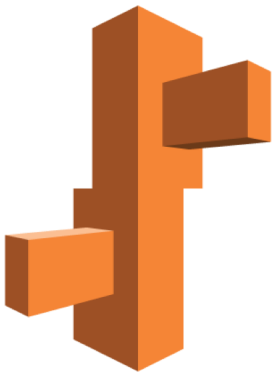
AWS Service	Advantages	Disadvantages
Elastic Beanstalk	1. Integrates with a variety of AWS services. 2. Easy deployment. 3. Quick. 4. Painless. 5.Neatly Documented.	You will get charged directly upon exceeding the free quota.
EC2	1. Fast and reliable cloud servers 2. Scalable 3. Easily managed 4. Low costing 5. Auto-scaling	1. Ui needs extra work Poor CPU performance. 2. High learning curve.

“Fast and reliable cloud servers” is the major cause for having a greater number of developers who prefer using Amazon EC2, yet a lower number of developers shed a light on the feature of “Integrates with a variety of AWS services” as the number one reason behind going with the AWS Elastic Beanstalk.

Summary

- THEORETICAL -

What is the Functionality of AWS Elastic Beanstalk?



Upon uploading an app, the AWS Elastic Beanstalk will directly takes care of the capacity provisioning's & deployment details, along with that of app health monitoring, auto-scaling & load balancing.

What is the Functionality of Amazon EC2?

Amazon EC2 is a web service which offers its users some resizable compute capacity when working with the cloud. This service helps in transforming web-scale computing into a simple and easy task for its developers.

- PRACTICAL -

- THEORETICAL -

This is the steps on how to create EB with correct setting and configuration also Build and starting the script.....
I just used the project 3 as refrence .....

1- CREATE ADMIN USER DURCH IAM CREATE GROUP AND THEN USER.

2- Save your acces-key and secret key.

3- instal awl CLI software from AWS website.

4- open any terminal and type AWC configure to configure the aws and enter the access-key and the secret-key and the region.

5-create bucket through terminal or gui.

6-write in the terminal the followoing:

aws s3api create-bucket --bucket the-name-of-the-bucket --region here-the-region

7-go to aws s3 then select the created bucket in the site and enable the static website and write the directory of html like index.html

8- go to bucket you will see in the last a link for the static bucket but it will give u 403

9-back to the bucket go to permission make it public go to policy then past this :

{

```
        "Version": "2012-10-17",
        "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::The-name-of-Bucket /*"
        }
        ]
}
```

10- now if you refresh the link you will get 404 instead of 403 this is normal cus you did not upload your site, no object in the bucket.

11- now go to build then deploy this is the comand to deploy:

deploy: npm run build && aws s3 sync name-of-file-to-be-build/ s3://name-of-Bucket

12-now after finished see the link again u will see the frontend.

13-now let's create database go to amazon RDS then click CREATE DATABASE

14- now choose the details of database:

-method: standard or easy choose standard.
-engine options: postgres.
-select the version of postgres sql.
-template: free tier
-deployment option: base on ur need.
-setting : DB instance identifier: name

- THEORETICAL -

-username: postgres
-password: postgres
-instance config: db instance class: db.t3.micro
-storage:autosacle need ? no
-connectivity: public access

15-then check that it's created successfuly and go to this database through postbird and write:
postgres://postgres:postgres@write-here-the-endpoint-ofdb:5432/postgres

16-go on the vpc and make a new traffic all with ip 0.0.0.0:0 to let anyone go on it then try to connect to db and it will be fine then u can now migrate up and see.

17- now install python then pip then eb after

18- make initialization :

eb init -i
Then choose all the suitable data to you
Like node version
SSH
And others

19- a folder of elasticbean is created now with yml config

20-now you need to create enviroment :
eb create --sample name-of-project

21- now go to file.yml and edit it: we need to write the deploy order to

- THEORETICAL -

specify where is the file
deploy:
    artifact: xxx/name-of-project.zip

22- make npm run build again

23- now check that the enviroment in created write in the terminal

    eb list
then you will see the enviroments that's are created and working.

24- to use this envriomeent as the default for the comming order use

eb use name-of-eb-enviroment

25- So now we can gather all of that and make it one script:

"deploy": "npm run build && eb list && eb use sad-api-dev && eb deploy",

After it may wait

26- if your project use .env variable
it's normal to get degraded 502 status

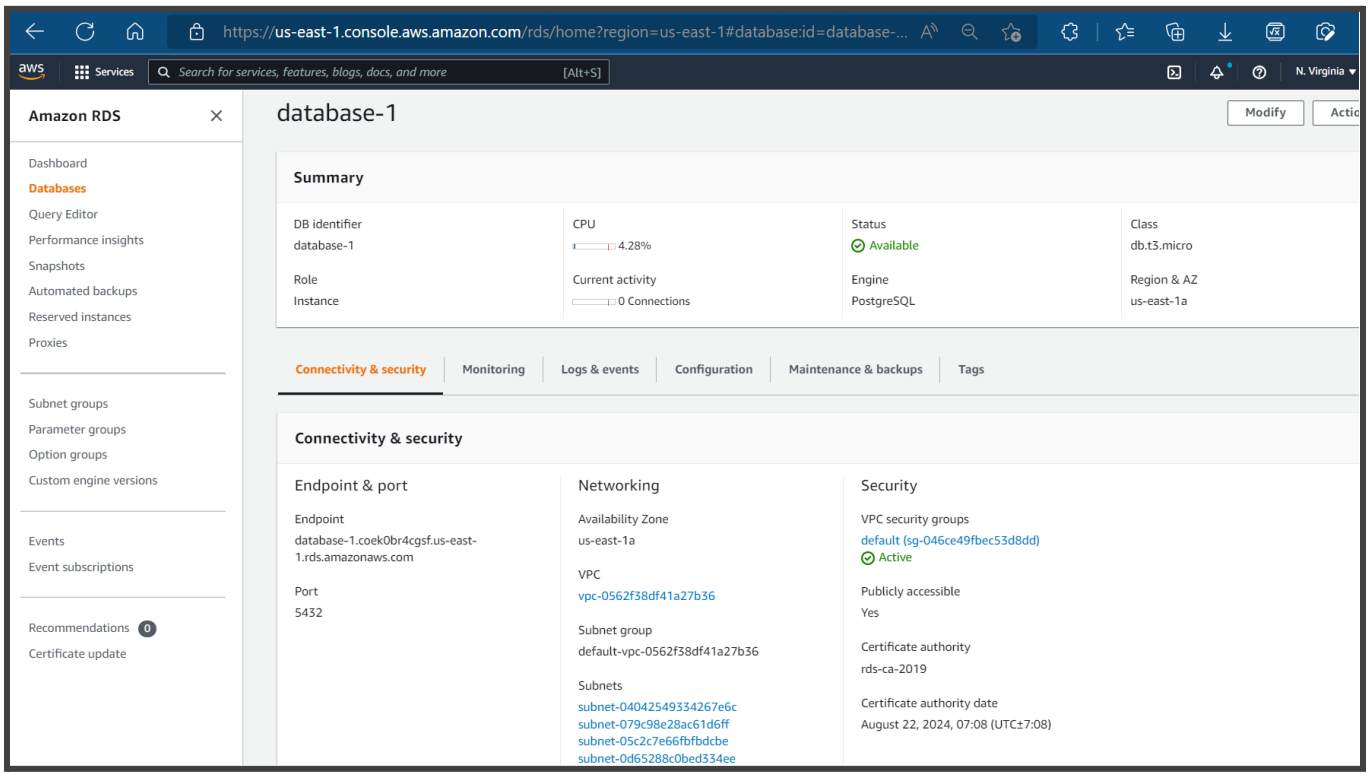
27- you should pass the enviroment variable:
eb setenv key="value" key="value" key="value" key="value"

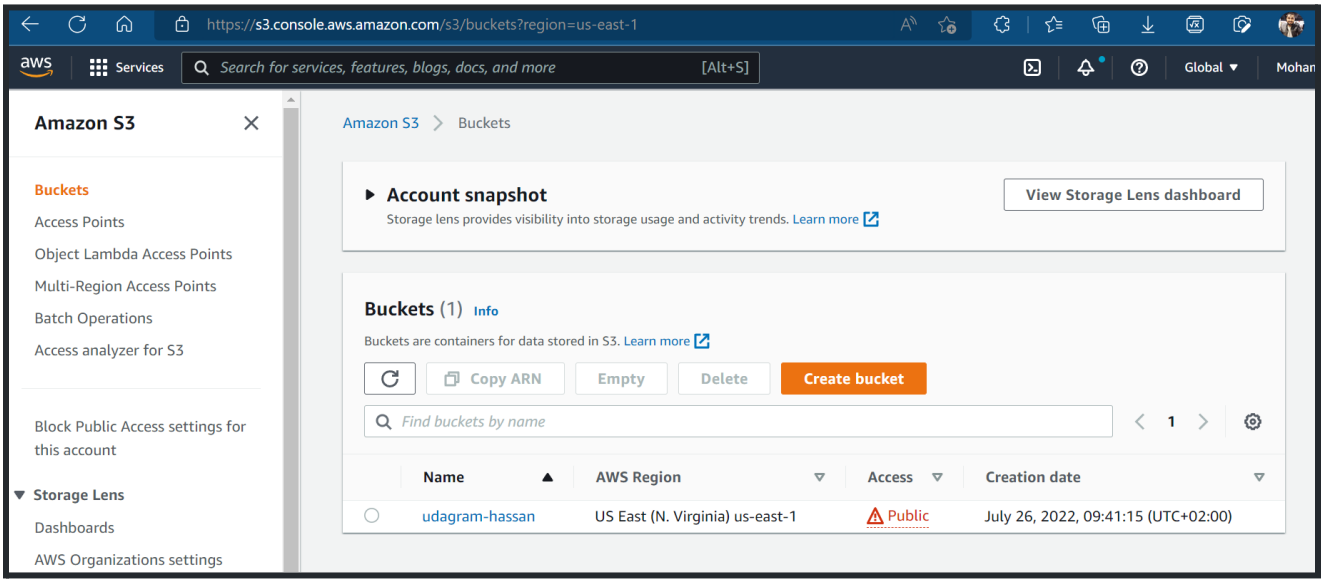
28- now your project works fine.

29-you must use : (if you are in test phases only)
eb terminate so not to get charged more money

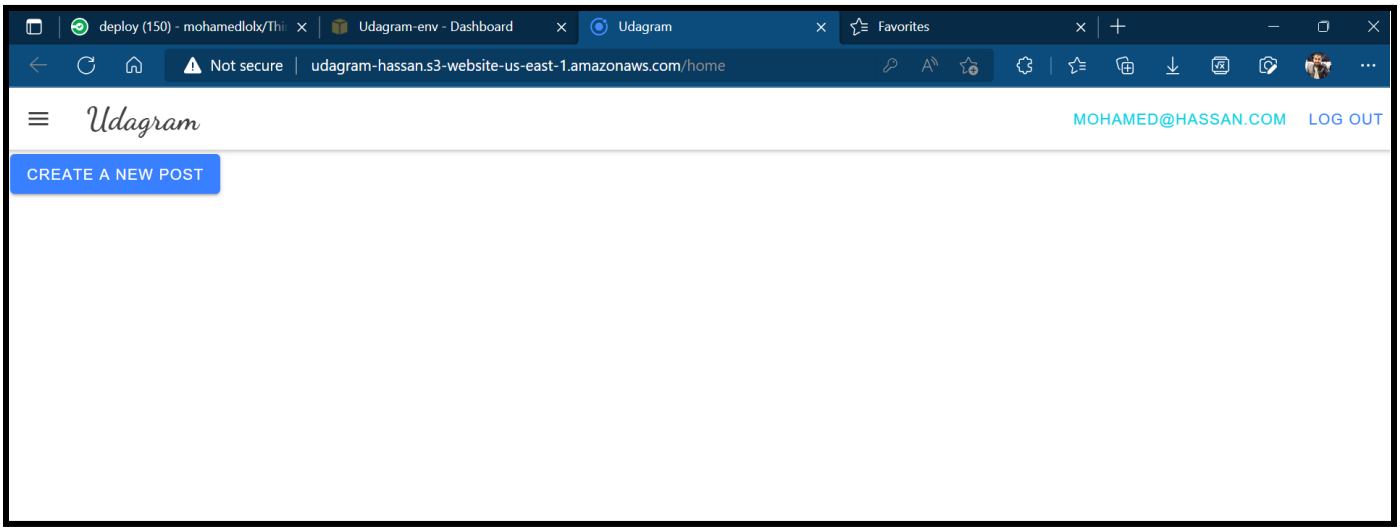
- THEORETICAL -

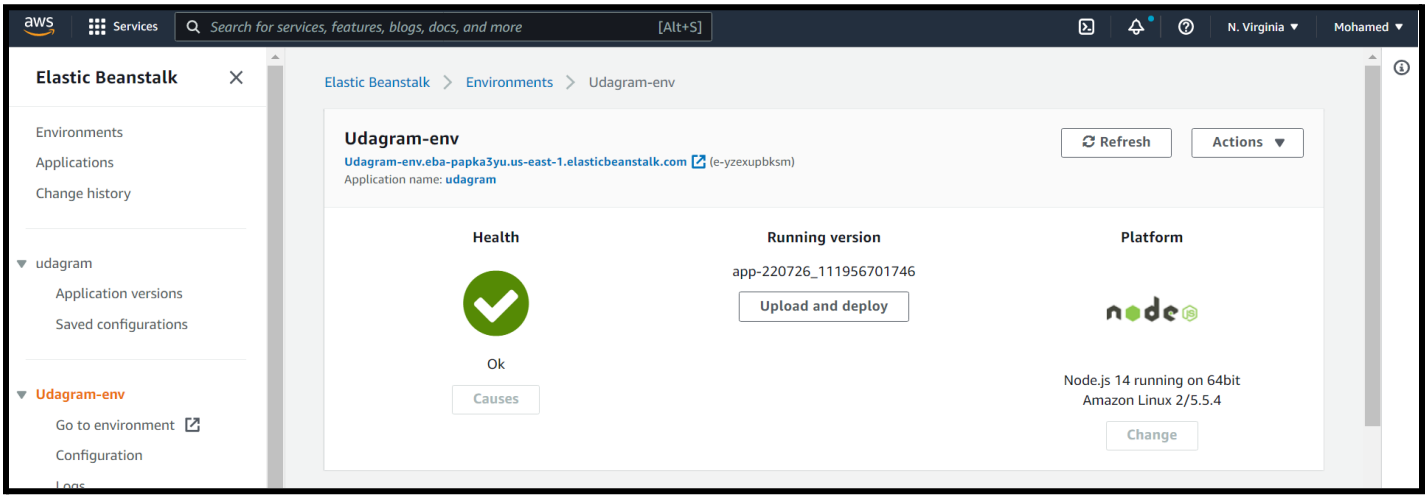
Here are some photosfor the results.



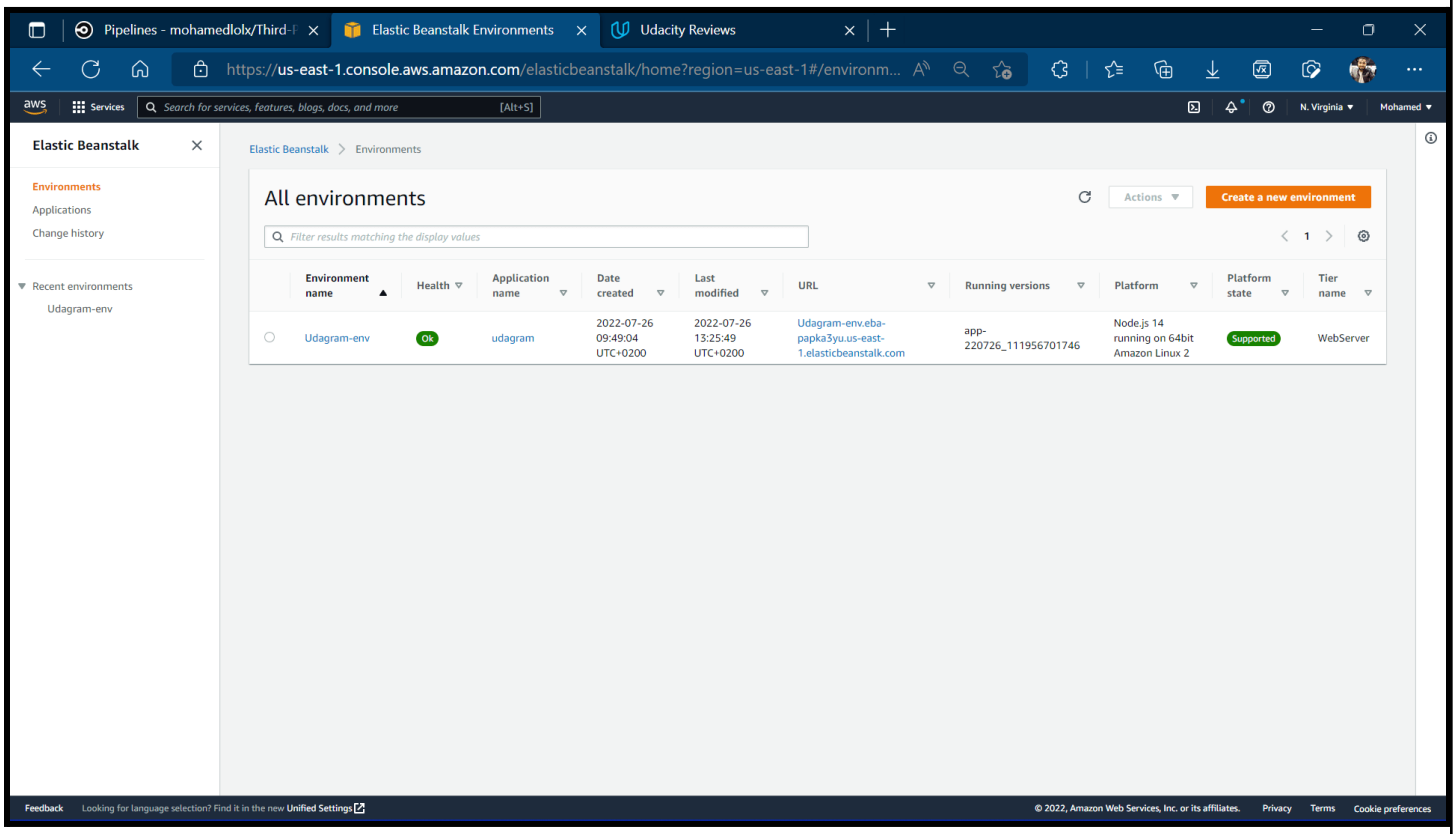


- THEORETICAL -





- THEORETICAL -



Shaymaa Hafez Ebrahiem
Github | Linkedin

- THEORETICAL -
- PRACTICAL -

Yousef alaa adel

https://www.linkedin.com/in/yousefalaa

- THEORETICAL -

There are usually 3 models of cloud services:

- Software as a service (SaaS)
- Platform as a service (PaaS)
- Infrastructure as a service (IaaS)

So if we are spotting the difference at first then the main difference here is not between elastic beanstalk and EC2, it's actually between platform as a service(elastic beanstalk), and Infrastructure as a service(EC2).

With PaaS, you typically only manage an application and its data. With Iaas, you also manage the runtime environment for the application and the operating system on which that environment runs.

With IaaS, you are so much involved in low-level decisions around network configuration, load balancing, and other 'infrastructure' components. In PaaS, these are abstracted away from you. For example,with PaaS, the decision to include load balancing is typically a checkbox. With IaaS, you would have to configure and spin up these components yourself, connect them together, and manage them.

So in small words elastic beanstalk makes it easier for you to configure your environment ,rather than do this configuration at a low-level yourself.

Even though using elastic beanstalk gives you simplicity, if the company has professionals using EC2 will give more control and will cost less.

- PRACTICAL -