

[THEORETICAL]

According to

[:https://www.cloudysave.com/aws/services/elastic-beanstalk/elastic-beanstalk-vs-ec2/](https://www.cloudysave.com/aws/services/elastic-beanstalk/elastic-beanstalk-vs-ec2/)

- **Functionality**

- 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

- 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.

- **Advantages and Disadvantages**

- AWS Elastic Beanstalk

- ❖ Advantages

1. Integrates with a variety of AWS services
2. Easy deployment
3. Quick
4. Painless
5. Neatly Documented

❖ Disadvantages

1. You will get charged directly upon exceeding the free quota

- Amazon EC2

❖ Advantages

1. Fast and reliable cloud servers
2. Scalable
3. Easily managed
4. Low costing
5. Auto-scaling

❖ Disadvantages

1. UI needs extra work
2. Poor CPU performance
3. High learning curve

● The Summary

- Amazon EC2

EC2 is Amazon's service that allows you to create a *server* (AWS calls these *instances*) in the AWS cloud. You pay by the hour and only what you use. You can do whatever you want with this instance as well as launch n number of instances.

- AWS Elastic Beanstalk

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 a 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 that it creates for you. If you have 10 hours of EC2 usage, then all you pay is 10 compute hours.

[PRACTICAL]

1. install the ElasticBeanstalk Cli.
2. initialize the EB Environment & Application using {eb init <APP_NAME> --platform Node.js --region <REGION>}.
3. A new folder will be created inside the udagram-api directory called `_.elasticbeanstalk_`
4. Create the Elastic environment using the following command { eb create --sample <ENVIRONMENT_NAME> }.
5. add the Environments variables to the EB
6. Create script **build** : **npm install . && npm run clean && tsc && cp -R .elasticbeanstalk dest/.elasticbeanstalk && cp .npmrc dest/.npmrc && cp package.json dest/package.json && cd dest && zip -r Archive.zip . && cd .**
7. build your API using the following command inside the App using this command {npm run build}.
8. After the build process is finished you need to deploy your app to the EB by using the following commands{
eb use ENVIRONMENT_NAME.
eb deploy ENVIRONMENT_NAME}