CIS PROJECT

1. AWS deployment to backend:

[**Elastic Compute (EC2)**](https://aws.amazon.com/ec2/)

I'd recommend looking into this for the more IO centered workloads. There's a number of available options including provisioned IOPS, Instance Store, high IO instance types, etc.

Advantages

* The most control over the system, and a setup that may be easier for those used to traditional VPS hosting
* A lot easier when you need to deal with custom libraries, native code, various languages
* IAM roles attached to an EC2 instance can be used to allow underlying code easy access to AWS resources
* A number of instance types available for high IO, high bandwidth, and other needs
* A system can be customized, then turned into an AMI to allow for reproducible systems
* May be easier to work with for some than Docker/Container solutions
* Can easily work with load balancers/auto scaling groups to scale up and down as needed, while being able to point to a single endpoint (the load balancer) and not having to worry about

Disadvantages

* Cost is going to be an issue
* Constant monitoring to make sure that your instance types meet your needs and you're not underutilizing what you're paying for
* The pricing model can be a little odd to work with since it's by per hour versus per month. There are however cost estimate calculators that will let you know your monthly estimated cost.
* If you use load balancing (which is ideal) there's additional costs
* Have to consider availability zones in where you place your servers, though Auto Scaling Groups can help with automating that. Still, it's another cost point to consider.
* You get to manage the underlying OS, including updatet
* To **Deploy NodeJS Application on AWS**
* https://youtu.be/\_U0pTlpyMGg?si=J89FxHY1IRBQFYKq

--> Cloud provides port number at runtime. So, port number can’t be hardcoded

--> We need to assign a dynamic variable/value for the port. Which is known as envirnoment variables

Learnt how to setup two/more services on a port



Need to download putty to genrate aws key for Ec2 instances

Need key to authenticate

For complete tut of putty <https://youtu.be/1UAkjMJLwy8?si=_ZJuW9g_ltZ_SYl1>

[](https://youtu.be/1UAkjMJLwy8?si=_ZJuW9g_ltZ_SYl1)

1. DynamoDB

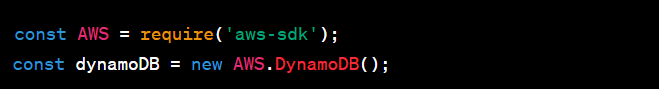
DynamoDB is a fully managed NoSQL database service provided by Amazon Web Services (AWS), and integrating it with a backend built in Node.js is a common and efficient choice.

**Installation of AWS SDK:**

* Start by installing the AWS SDK for JavaScript in Node.js (**aws-sdk** package) using npm

**Setting Up AWS Credentials:**

* Ensure that AWS credentials are configured either through environment variables, AWS CLI, or IAM roles associated with your EC2 instance (if running on AWS)



**Middleware or ORM:**

* You may choose to use middleware libraries or Object-Relational Mapping (ORM) libraries designed for DynamoDB and Node.js, such as **dynamodb-data-mapper** or **dynogels**, to simplify interactions.

S3:

In AWS everything is an API call

Amazon Simple Storage Service (S3) is a scalable object storage service provided by Amazon Web Services (AWS). It is commonly used for storing and retrieving files, including images.

Step to perform

1) **Create an S3 Bucket**

2) **Set Bucket Permissions**

3) **Generate AWS Access Credentials**

4) **Use AWS SDK in Backend**

npm install aws-sdk

5) **Frontend Integration**

6) **Retrieve Images (profile)**

**Working of COMPLETE S3 TUTORIALS**

<https://youtu.be/DOUxRYi2Fwg?si=xY97vydbYfP3IGvf>

[](https://youtu.be/DOUxRYi2Fwg?si=xY97vydbYfP3IGvf)

# **Upload Images Directly to S3 from Front End**

<https://youtu.be/yGYeYJpRWPM?si=SHloVhbGOGKOu7Vx>

[](https://youtu.be/yGYeYJpRWPM?si=SHloVhbGOGKOu7Vx)

# **Storing Images in S3 from Node Server**

<https://youtu.be/eQAIojcArRY?si=g-U_ZtHeWNX_OsCf>

[](https://youtu.be/eQAIojcArRY?si=g-U_ZtHeWNX_OsCf)