**6. Introduction to Deployment Services**

**Vercel**

In the world of web development, efficiency and speed are paramount.

Developers constantly seek tools that streamline the deployment process while maintaining performance and reliability.

**Vercel:**

* Vercel is a cloud platform designed to simplify the deployment process for web applications, particularly those built with modern frameworks like React, Next, etc.
* It lets you deploy their front end stupidly easily.
* Beyond the frontend-as-a-service offering, Vercel has consistently released features to make it a one-stop solution to build your entire application:
  1. **Edge Functions: -** let you write backend code serverless to handle things like authentication etc.
  2. **Analytics: -** statistics and dashboards on your site’s performance and accessibility.
  3. **CLI / API: -** interact programmatically with Vercel to run deployments, performance checks, etc.
* Vercel operates on serverless architecture combined with a global content delivery network (CDN).

**Working of Vercel on high level:**

Deploying a website on Vercel typically follows a three-stage process at a high level:

1. Uploading the Project Files.
2. Creating the Deployment.
3. Request Phase.
4. **Uploading the Project Files:**

A POST request is made containing the project’s files to be uploaded to a scalable, secure, and highly durable data storage service.

In simple words, it takes the files from your GitHub Repository and then uploads them to something like AWS S3.

1. **Creating the deployment:**

Once the files have been uploaded successfully to the storage service, another POST request is made to start the build and deployment process.

Here’s a breakdown of the deployment process on Vercel:

1. **Authentication and Validation:** Vercel authenticates the user and checks the request’s authenticity and permissions against the vercel.json file to prevent unauthorized access.
2. **Scheduling Build:** If everything checks out, the deployment is scheduled for building.
3. **Tracking Build Progress:** While processing files, the build container pings an API endpoint to track deployment status, visible in the CLI and dashboard.
4. **Ready for CDN:** Once resources are provisioned and metadata is uploaded, the deployment is ready to be served via the Vercel CDN

In Simple words and keeping all the complex things as a black box, the Deployment Service grabs the repository code from the Storage location (like S3) and then builds that project and stores the build in the Storage.

1. **Request phase:**

After deployment, you receive a URL for your website, such as myapp.vercel.app which you can use to access it.

How Vercel accomplishes this behind the scenes:

1. **DNS Lookup:** When you type a website’s address, Vercel uses smart routing to make sure your request reaches the closest data center, which makes the website load faster.
2. **Content Serving:** Different types of content like static pages, special functions, or optimized images are served accordingly.
3. **Caching:** Vercel caches some of the stuff it sends you, so the next time you ask for it, it can give it to you faster.

**Services Vercel uses:**

1. Amazon S3
2. Amazon Simple Queue Service
3. Auto-scaling fleet of EC2 instances powered by AWS Fargate
4. Amazon Global Accelerator
5. AWS Global Network
6. Amazon EKS
7. AWS Lambda