IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Modern Web Apps using Full Stack Development and Containerization

Mrudula Deore[1], Mayuri Kambli[2], Chinmayi Kulkarni[3], Asst. Prof. Sunil Chaudhari[4]

B.E. Student, Computer Department, Fr. CRCE, Mumbai University, India[1,2,3]

Assistant Professor, Computer Department, Fr. CRCE, Mumbai University, India[4]

Abstract: This paper presents a web application created using technologies like the MEAN stack, Docker, Amazon S3 and Amazon EC2. Where everyone is creating web applications using Java, JavaScript and HTML, MEAN is a fresh and emerging concept that helps building dynamic web applications using only JavaScript and HTML. Docker helps in binding all the dependencies of an application enabling you to build, ship and test applications easily. This paper weighs all the advantages provided by these technologies and presents an architecture formed by creating a Docker image of a MEAN web application which is deployed on EC2 server provided by Amazon.

Index Terms: Web Application, MEAN Stack, Docker, AWS.

I. INTRODUCTION

In today's time, there is no shortage of products and applications available for the customers to choose from. The current tech-savvy generation, has abundant options to choose from and in case they are not satisfied with the same, they can ea sy switch to something better.

Hence to keep up with the needs of the today's clients it is highly essential to develop applications that are fully functional, easy-to-use and fast. Thus the developers can choose from the different frameworks available to create their applications.

Being a new technology MEAN Stack is still gaining its popularity rapidly due to the various advantages that come along with the features provided by the existing frameworks.

The MEAN Stack actually refers to Mongo DB. Express JS, Angular JS and Node JS, thus giving the developer an advantage of writing the entire application in one language.

MEAN Stack uses Java script for frontend as well as backend operations making it easier to develop and maintain code in a simple, readable format.

II. RELATED WORK

MEAN stack is a young technology and there have been several different frameworks that are still being used. One of these frameworks includes LAMP, an open source web development platform. LAMP is referred as Lamp Stack as it consists of different layers.

A. Architecture of Lamp

The Lamp stack consists of four layers and each of the layer plays a crucial role in the entire system stack:

Fig. 1. LAMP Architecture

1) Linux: Linux is the bottom layer that provides the operating system. It is basically used to run all the other components. But it does not necessarily have to be Linux as we can use other operating systems like Windows (WAMP), Macintosh (MAMP), Solaris (SAMP) etc, as required.

2) Apache: Next is the web server layer that provides the mechanisms to get the Web page to the user. Being a stable, critical-mission-capable server it is used to run more than 65 percent of the web pages on the Internet. The PHP component is actually a part of apache which can be used together to create dynamic web pages.

3) MySQL: We are already familiar with MySQL database and we use the same for data storage in LAMP system. MySQL can provide a suitable database for running large and complex sites. The data can be stored in a systematic way making it easy to query with SQL language.

DOI 10.17148/IJARCCE.2016.5518

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

4) PHP: PHP is an efficient programming language that binds together all the other parts of LAMP. It can be used to make dynamic content that makes it possible to access data from the MySQL database and for some other features provided by the operating system.

B. Features of LAMP Stack

- The main idea behind LAMP is to use these technologies together as they are readily and freely available making

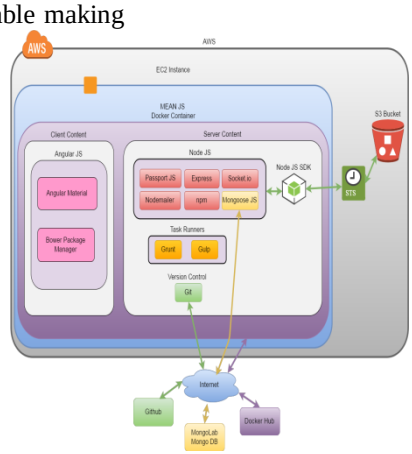

Fig. 2. Project Overview

it a development standard to make a powerful web application platform.

- It is flexible, secure and easy to deploy.
- It is easy to develop applications with LAMP and can be customized as needed.
- Also it comes with a huge support community which is highly beneficial for all the developers including beginners and experts.
- And one of the most important benefits includes open source nature of LAMP stack.

C. Advantages of MEAN over LAMP

- Node JS is much faster as well as scalable than Apache server, because it provides non-blocking I/O architecture. It does not restricts JavaScript to Browser and utilizes it at server side too.
- Angular JS is a powerful JavaScript framework because of its dependency injection and data binding, which makes it reusable and easily testable.

MongoDB supports JSON-like documents which is compatible with JavaScript code at client as well as server, reducing need for ORMs in an object oriented language such as JavaScript. You can store unstructured data and that too in a single document, thus a single read operation can return much of the data. This removes the need for complex joins which are required in MySQL.

III. PROPOSED SYSTEM

A. Architecture Overview

The architecture consists of a MEAN(Mongo DB, Exress JS, Angular JS, Node JS) Stack application deployed on EC2 service of AWS inside a Docker container. The EC2 helps to easily manage our instance. The instance is located in a Virtual Private Cloud (VPC). We can decide which instance are exposed to the Internet and which remain private. The Docker container is run by using a MEAN.JS Docker image pulled from Docker Hub for better resource utilization and allowing flexibility for any modifications. The MongoDB database is hosted as a separate instance for the security concerns in Mongo Lab. MongoLab is a cloud database service that hosts MongoDB. For storage of files we are using AWS S3 bucket. Amazon S3 (Simple Storage Service) is an online file storage web service. Amazon S3 provides storage through web services interfaces (like REST, SOAP). The AWS-SDK for Node JS enables us to access AWS services from Node JS code running in the server. With the help of this sdk, we upload the files to S3 using REST API. The language used is JavaScript for the entire application. The user requests are handled by Angular JS, these requests are forwarded to Express JS through REST API calls. Express JS is a framework for Node JS. Using Mongoose, the Express JS can talk to MongoDB database residing in MongoLab Cloud using JSON data.

B. Proposed Architecture

Full Stack Development provides a developer an opportunity to contribute in front-end as well as back-end development. A Full Stack developer needs to have functional knowledge of all stages in software development. In MEAN Stack as the language used is JavaScript only, it becomes easier for the developer to master all the layers of stack. This enables the programmer who is working as web designer to work on back-end whenever necessary which reduces time spent on learning a new back-end language.

1) MEAN Stack

The base of this architecture is the MEAN stack which enables you to use one language, JavaScript, for server-side as well as client-side programming. It is free and is used to develop dynamic web applications. The components of the MEAN stack are MongoDB, a no SQL database, Express.js, a framework that runs on Node.js, Angular.js, a JavaScript MVC framework used in the frontend and Node.js, the execution environment for event driven server side applications.

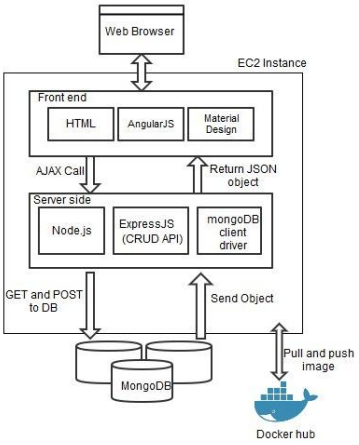

Fig. 3. Project Architecture

DOI 10.17148/IJARCCE.2016.5518

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

1.1) Front End
The front end comprises of
a) Angular.js
Angular.js is a structural framework that lets you use HTML as a template language and extend its syntax to express your applications components clearly. Angular to reduce the gap between document centric HTML and what an application actually needs by letting you create new HTML constructs. The main features of Angular.js are :

i) Model View ViewModel
AngularJS does not implement MVC (Model-View-Controller) in the traditional sense, but in the form of MVVM (Model- View-ViewModel). MVVM supports two-way data binding between view and View model. This enables automatic propagation of changes, within the state of view model to the View. The view model uses the observer pattern to notify changes in the view model to model.

ii) Data-binding and Dependency Injection
The MVVM pattern helps in communicating everything automatically across the UI whenever anything changes. Thus eliminating the need for wrappers, getters or setters or class declarations. AngularJS handles all these dependencies automatically and hence you can set your dependencies as parameters in AngularJS service functions, rather than one giant main() call to execute your code.

iii) Directives
Directives let you bring extra functionalities to HTML. They enable us to create new HTML elements depending on the requirement of the application. It eliminates the DOM manipulation code from the MVC and hence the MVC is only responsible for updating the view with new data and how the view behaves is taken care of by the directives.

iv) Deep Linking for dynamic web pages
Deep linking enables you to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.

v) UI Router
It is a library provided by AngularUI that enables to organize interfaces by state machines rather than a simple URL route. It allows creation of nested views, using multiple views on the same page, have multiples views that control a single view, etc.

b) Angular Material
Angular Material is UI library for Angular.js development. It helps in building functional, attractive and consistent web applications.

It enables the application to provide responsive interaction. Angular Material Provides the following features:

- Browser Portability
- Device Independence
- Specialized features like cards, toolbar, speed dial, side navigation, swipe, and so on.
- Standard CSS with minimal footprint
- Responsive Design

c) Bower Package Manager
Web Applications are a collection of frameworks, libraries, utilities and assets. Keeping a track of all the packages and making sure that they are up to date is a tedious task. Bower makes this task simple, it installs the right versions of the packages and their dependencies.

1.2) Back End
The back end comprises of
a) Node JS
Node.js is a powerful platform built on Google chromes javascript V8 engine that takes care of the server side code in a MEAN application. It is an open source, cross platform run time environment to develop server side web applications. Node.js uses rich libraries of Javascript modules making development of web applications easier. Node.js has proved to be the perfect technology for I/O bound applications, data streaming applications, JSON API based applications, single page applications and many more.

As mentioned in the official documentation[1] of node.js:
”Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.”

i) Sending mail using Nodemailer
Nodemailer is a simple easy-to-use module that can be installed with npm which is used to send e-mails with Node.js. It uses SMTP or sendmail or Amazon SES. It is Unicode friendly that means we can use any characters. Nodemailer supports not just HTML content but also plain text alternatives. It supports embedded images in HTML , SSL for secure e-mail delivery, uses preconfigured services for SMTP with Gmail, Hotmail etc.

ii) Node Package Manager - npm
npm is a default package manager for JavaScript. It helps us to find, share and use existing packages which can be assembled in our own applications. npm provides online repositories for node.js packages and command line utility for downloading node packages, version and dependency management of node.js packages. It can be used for searching, updating, creating and uninstalling packages.

iii) Features of node.js:
- Node.js is asynchronous and event driven. It is said to be non-blocking because node.js based server doesnt wait for the data to return. It moves to the next API and the events of node.js using a notification mechanism get the response of the earlier API call.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

- As it is built on Google chromes V8 JavaScript engine code execution is faster with node.js.
- Unlike other multithreaded frameworks, node.js is single threaded but provides high scalability. The server depicts a non-blocking behaviour due to event mechanism mak- ing it highly scalable. Using a single threaded program it can give service to higher number of requests compared to traditional servers.
- As node.js applications output the data in form of chunks they never buffer any data.

b) Express JS
Express is a node.js web application framework which gives a robust set of features to develop applications. It helps in rapid development of web applications based on node.js. We can build REST interface using express on top of node.js which is used to perform CRUD operations on the database.

Features of express:
- To respond to HTTP Requests it allows to set up middle- wares.
- To decide the different actions to be performed based on HTTP method and URL it defines a routing table
- It allows to dynamically render HTML Pages based on passing arguments to templates.

c) MongoDB
MongoDB and Node.js both make use of JavaScript and JSON which is becoming the standard data format for web APIs. MongoDB is an Open Source Document Database which can handle unstructured data. It is one of the leading NoSQL Database as shown in figure given below. MongoDB is written in C++ and it is a cross platform database. It works on the concept of collection and documents. A collection is a group of documents. A single MongoDB server has multiple databases. A Database is a physical container of collections. Data can be retrieved using MongoDB query language. MongoDB utilizes the innovations of NoSQL while maintaining the foundation of RDBMS. MongoLab is a Platform as a Service(PasS) which allows to host MongoDB remotely.

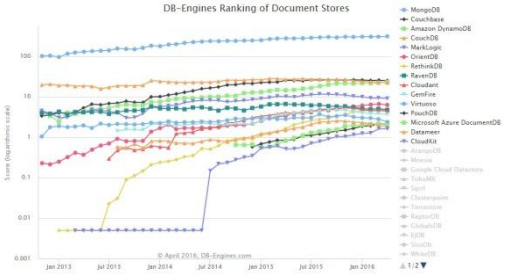

Fig. 4. DB-Engines Ranking-Document Databases

i) Querying MongoDB using Node JS
Mongoose is a Node.js Object Document Mapper(ODM). ODMs are the MongoDB equivalent of Object Relational Mappers (ORMs). Mongoose is written on top of the native Node.js MongoDB driver. Mongoose defines models and fires queries on them. The native Node.js MongoDB provides a more traditional querying interface. MongoDB, Mongoose and Node.js all use JSON. Mongoose schemas are enforced at the ODM layer by Mongoose. Mongoose.js has utilities for simplifying Nodes callback patterns which make it easier to work. Thus Mongoose makes it easier to use MongoDB with Node.js.

ii) Benefits of Using MongoDB
MongoDB helps application to use Diverse Data Types at backend, generate schemas dynamically. It also handles replication and provides auto-sharding. It removes the need for complex object-relational mapping layer that translates objects in code to relational tables. Using MongoDB making changes in data model becomes easy which helps in building application faster.

2) Authentication using Passport.js
Passport JS is Express based authentication middleware for Node.js. It authenticates request, using strategies. The Local Strategy authenticates users by looking up their data in the application's database. Passport JS maximizes flexibility by allowing application-level decisions to be made by developer. Give Passport a request to authenticate, then it provides ways for controlling what occurs when authentication succeeds or fails.

2.1) Strategies
Passport uses different strategies to authenticate requests. Strategies can be verifying username and password credentials, authentication using OAuth, authentication using OpenID. Before authenticating requests, the strategy used by an application must be configured. Passport has around 300 authentication strategies covering social networking, enterprise integration, API services etc.

2.2) Sessions
Passport maintains persistent login sessions. In persistent sessions, the authenticated user must be serialized to the session, and de-serialized when subsequent requests are made. Passport does not put any restrictions on how your user records are stored. You provide functions to Passport which implements serialization and de-serialization logic.

3) Version Control System - git
It is a free open sourced version control system that helps to manage large projects with speed and efficiency. It is used for Source Code Management which emphasizes on data integrity and support for distributed, non-linear workflows.

IV. TASK RUNNERS

4.1) Grunt
Grunt is a task-based command line build tool for JavaScript projects. It helps you perform repetitive tasks like minifying files, compiling, unit testing, etc easily. It allows you to perform mundane work effortlessly.

4.2) Gulp
Gulp is a streaming build system, by using nodes streams file manipulation is all done in memory, and a file is not

written until you tell it to do so. Gulp provides tools to create tasks that fit the need of the application by preferring code over configuration.

V. TESTING

JavaScript helps you build dynamic applications and provides a capability to express but to do all this it provides no help from the compiler. Thus making it essential to test applications written in JavaScript.

5.1) Karma Unit Testing
Karma, a Node.js application, is a test runner for JavaScript. It is a command line tool that is used to spawn a web server which loads the source code and runs the test case. Karma lets you run tests on any browser on any platform (desktop, phone, tablet, etc.)

5.2) Protractor End to End Testing
Protractor is a Node.js application that uses WebDriver to control browsers and simulate user actions. It is used to check whether the flow of the application is performing as designed from start to end.

VI. CONTAINERIZATION

Containers are innovating software lifecycle. The contents of each container are kept isolated from that of others. To make container image we need to define what needs to be there for your application to work such as OS, libraries, configuration files, application binaries etc. The container image is used to create containers that run in any environment. Containers are very efficient. They can run on same machine and can also allow full use of all available resources.

6.1) Containers vs Virtual Machines
Virtual Machine(VM) are heavier than containers and uses lot of system resources. Each VM runs an entire copy of an operating system along with virtual copy of all the hardware that operating system needs to run. This increases RAM and CPU cycles. But, a container requires part of an operating system, supporting programs, libraries, and system resources to run a specific program. Thus you can can run 2-3 times as many as applications on a single server with containers as compared to a VM. Containers help in craeting a portable operating environment for development, testing as well as deployment.

6.2) Docker - Running application in a container
Docker is an open-source project. It automates the deployment of applications inside software containers. Along with that it provides a layer of abstraction and automation of OS level virtualization.

Docker is designed to benefit both developers and operations teams which makes it a part of many DevOps tool chains. Docker gives flexibility and reduces the number of systems needed. The reason is it has small footprint and lower overhead.

6.3) Pulling image from Docker Hub
The Docker Hub is a registry service for building and shipping containers. It provides container image discovery, image distribution as well as change management. In Docker Hub we can push our committed Docker images, which can be pulled anywhere, anytime.

VII. AMAZON WEB SERVICES(AWS)

7.1) AWS EC2 (Elastic Compute Cloud)
Amazon provides cloud computing services called as Amazon Cloud Services(AWS). It provides on-demand computing platform on pay-per-use billing. EC2 is Infrastructure as a Service which acts as processing part of AWS.In EC2 you can create Virtual Machine instances to run or develop your applications. Customers can lease computing resources such as virtualized servers and applications on EC2. It is called Elastic because it can scale as per application's need automatically.

7.2) Uploading files on S3
Uploading files such as images, videos using web application is an integral part of modern web applications. Amazon's Simple Storage Service(S3) is a popular and reliable storage service, which helps developers to upload the files remotely instead of on local disk. S3 is comprised of a set of buckets. Each bucket has given a unique name which helps to identify it among other buckets. This bucket can store individual files (known as objects) and directories. The files can be uploaded to S3 using an Access Key ID and a Secret Access Key. These act as authentication credentials for user of the bucket. Even though you have Access Key, you must have sufficient privileges to the bucket, otherwise you will not be able to upload the files. Using AWS SDK for Node JS, file can be uploaded on S3 in a node.js application.

a) S3s Cross-Origin Resource Sharing (CORS) support
If you want to access resources of other domain from the current domain, then what you need is Cross-origin resource sharing (CORS). It makes the resource sharing secure and prevents hacking. With CORS support in Amazon S3, the files uploaded on S3 remain secure and can be accessed from other domains.

b) Benefits of using S3 instead of local disk
You don't have to worry about free disk space when you use S3. If you use PaaS services, you should not save your files locally because those locally saved files will be gone when you deploy newer version of your app. S3, holds more than 52 billion objects and regularly peaks at 80,000 requests per second, according to the Amazon company.

c) Security Token Service in AWS
Using STS, we can request temporary and limited permissions for AWS Identity and Access Management (IAM) users or for users that you authenticate. Security Token Service is designed to have limited access to Amazon Services. STS and Security Groups both together make your application secure from all sides.

VIII. CONCLUSION

Full Stack Development using MEAN Stack is a robust, flexible, scalable solution for the modern web applications. It provides good isolation among the 4 technologies of which it consists of, by presenting it in a hierarchical folder structure such as MEAN.JS and MEAN.IO. The Full Stack Development does not restrict a developer at one or few layers of the software development, but allows developer to explore all the layers of the stack with the power of only one programming language i.e. JavaScript.

Angular JS provides great data binding and it is used for developing single page applications because of its powerful routing techniques. Node JS plays a vital role in today's server family. It is a non-blocking I/O server, thus serves more requests as compared to the present web servers. Express makes the server-side operations more easier for Node. The MongoDB allows programmers to store unstructured data in a single document, thus with just a single read operation we can get all data of that records without performing any complex join operations. Also the services at the server side can be utilized directly in a mobile application such as android application as the data returned is in JSON format. Thus it reduces the need for coding web services for mobile applications separately. Many startups prefer MEAN stack to speed up the development of their application as it is programmer-friendly.

Containerization using Docker helps in resource utilization and AWS makes it easier to scale up your system and make it available all the time. The MEAN Stack and Docker have great community support. Amazon Web Services, Google Cloud Foundry and other cloud service providers have started docker support in their services because of rising community and also it helps them to manage their own resources, because virtual machines consume more space than docker containers.

REFERENCES

[1]. Node JS Documentation (https://nodejs.org/en/docs/)
[2]. Angular JS Documentation (https://docs.angularjs.org/guide/concepts)
[3]. Angular Material Documentation (https://material.angularjs.org/latest/)
[4]. Express JS API Reference (http://expressjs.com/en/api.html)
[5]. MongoDB Manual (https://docs.mongodb.org/manual/)
[6]. Passport JS Documentation (http://passportjs.org/docs)
[7]. Passport JS Github (https://github.com/jaredhanson/passport)
[8]. Docker User Guide (https://docs.docker.com/engine/userguide)
[9]. AWS Documentation (https://aws.amazon.com/documentation/)

DOI 10.17148/IJARCCE.2016.5518