International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 2325-2332 www.ijaem.net ISSN: 2395-5252

Deploying and Setting up Ci/Cd Pipeline for Web Development Project on Aws Using Jenkins

Kavya N, Smitha P
Department of Information Science and Engineering East West Institute of Technology Bangalore, India
Department of Information Science and Engineering East West Institute of Technology Bangalore, India

Submitted: 15-06-2022 Revised: 25-06-2022 Accepted: 27-06-2022

ABSTRACT—The objective of this project is to gain a better knowledge of how to create a website using AWS and various tools. in order to use cloud computing services obtain excellent performance at a low cost . Since Amazon EC2 is so popular, it's been used here. well-known for providing excellent cloud services in computing. In this paper, I'll follow you through the process step by step,how to use the Amazon Web Services (AWS) Provisioning and producing a market.AWS cloud server has been added. Once website created we need to deploy the CI/CD.This might be accomplished using the DevOps methodology. To achieve high performance and quality assurance products, DevOps approach extends agile to swiftly build software and automatically deploy it across many platforms/environments.The backbone of a DevOps infrastructure is continuous integration/continuous deployment (CI/CD). CI/CD bridges the gap between development and operations teams by automating the building, testing, and deployment of software. DevOps tools such as Git, Maven, Jenkins, Terraform, Docker, and Kuberentes are used to automate the complete environment.

Keywords—AWS, Agile, CICD, DevOps, Docker, Git, Jenkins, Kuberenetes, Maven, Terraform.

I. INTRODUCTION

Amazon Web Services, or AWS, is one of the most well-known and reliable cloud computing platforms in the world.Because it is very safe and secure, deploying on AWS can save you money, manpower, and time as compared to constructing and maintaining more traditional systems. AWS's Amazon EC2 is a computing online service that delivers instances that are scalable.Virtual machines are all there is. It is extremely dependable, scalable, secure, adaptable, and simple to use.to put to use It is also linked to other websites.AWS's services It enables you to pay for anything you choose.only the amount of computational power you actually need use.

Amazon ELASTIC COMPUTE CLOUD(EC2) introduces a fresh perspective and standard for web hosting. It gives developers the power to scale their many computers up or down in minutes, allowing them to create distributed and scalable applications that run on the cloud. EC2 is safe, dependable, adaptable, and, most importantly, inexpensive.

You only pay for the assets that you actually use, and you can sell your multi-server application to the general public for a far lower price than you could at any other time while maintaining an extraordinarily high degree of value and accessibility.

DevOps is a term that refers to the synchronization of automation and programmable software development, as well as infrastructure deployment and maintenance. Continuous integration, continuous deployment, and continuous delivery are significant characteristics in the software business that allow firms provide new features and products that are reliable on a regular basis. Continuous integration focuses on merging each developer's work many times per day in order to make mistake troubleshooting simple.

Continuous delivery aims to eliminate inconsistencies in the deployment or release process and automate the build stage so that code can be delivered safely at any time.In the software delivery lifecycle, the CI/CD pipeline delivers the following benefits: immediate customer feedback, rapid and steady release leads to customer satisfaction and a quality assured product, and CD helps to automate processes that were previously done manually.

Developers at our company push their code to a common repository every time they write it, so that other developers in various locations may access it and learn about the project's current progress. Now is the time to build the code in the shared repository.

The building tool's job is to download and convert the code's dependencies into a package. As a result, the testing team does not need to download the dependencies again. If there is an error, they are returned to the developer. This is a manual procedure. In order to automate the entire process, continuous integration process can be used. The CI mechanism checks the shared repository often for new code arrivals. If a new code is released,,build it. This entire process contributes to continuous integration.After building the code, it should deploy to a web server hosted with container technology. This process contributes to continuous deployment.

II. LITERATURE SURVEY

Development, testing, and production are the three phases of project development. The software industry initially used the Waterfall technique. If a project must be completed in 12 months, the time necessary is divided into three phases: development takes six months, testing takes three months, and production takes three months. During the six-month development period, the developer writes all of the project's code. The code is subsequently sent to the testing step after it has been developed. The entire code is checked for errors during the testing step.If an error occurs, the testing team will notify the development team. Now the developer begins debugging the mistake and re-sends the modified code to the testing team.

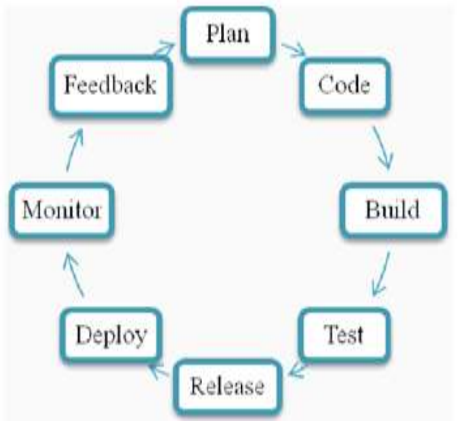

Fig.1. DevOps lifecycle

The testing team should go over the entire code again for errors. As a result, it is a time-consuming procedure. As a result, the project will not be completed within the stipulated time frame, such as 12 months. Many new technologies will develop during the next 12 months, and they will be unable to access them. As a result, adding new technologies in the middle of a project is impossible. There is also a lack of communication between the developer and the rest of the team.

To address the aforementioned issues, the industry began to use agile methodologies. The project is broken down into modules here called Sprints.It encourages the adoption of innovative technology.In this technique, as well as the phases of development and testing, are collaborated. Each sprint has its own code, which is written before and after the sprint.After each sprint is completed, it is moved to the testing phase. Thust is simple to identify and correct errors in a single sprint.It's simple and time-consuming to make a mistake. After you've completed all of the necessary repairs,The code is then pushed to the production phase with no errors.Agile methodology is faster than the waterfall model.There are certain disadvantages to this approach. They are: (i) it fails to produce the product on time; (ii) the project deviates due to a lack of documentation during talks; and (iii) the task is completed in a short amount of time. As a result, these are the limits of the software industry's previous technique. The DevOps approach era began in order to address these limitations.

III. DATASET & METHODOLOGY

A. Steps to follow to create a website

Step1: Register With Amazon Web Services (AWS) :
To access AWS's services, you must first create an account with the company. You must follow the procedures outlined below to create an Amazon account:
1. Go to http://aws.amazon.com and select "Create an AWS account" from the drop-down menu at the top of the screen.
2. Fill in your email address, password, and AWS account name on the next screen. Then, to begin the registration process, select "Continue."
3. Once that's done, sign up for AWS by choosing an account type and entering your basic contact information and phone number.
4. Once you've completed that, proceed to the next step by entering your credit card information. To proceed with the account creation, click the "Secure Submit" button.
5. You don't have to be concerned about overcharging for services because you only pay for what you utilise. However, in order to avoid service misuse and for security reasons, Amazon still wants your credit card information to validate your identity.
6. Amazon will then make an automated phone call to the number you provided, prompting you to type the PIN number displayed on the screen to verify your identification.
7. Once your identification has been verified, select the "Basic" support package and confirm your account. You'll then receive an email confirming your account, indicating that you've successfully registered with AWS.

Step2: Generate An AWS Key Pair
Follow the procedures below to generate an SSH key pair that will be required to log in to your EC2 instances:
1. Open the AWS Console and log in.
2. Select the EC2 service from the Amazon Web Services menu.
3. From the Amazon EC2 dashboard, go to the "Network & Security" menu and select "Key Pairs."
4. Select "Create Key Pair" from the drop-down menu.Enter a name for the new key pair in the resulting dialogue box and click the "Create" button.
5. A new key pair will be created, consisting of an SSH public and private key. The private SSH key will be downloaded to your computer and you will be prompted to do so.
6. Only one copy of the private SSH key will be available for download. It's important to keep it safe because you won't be able to access your AWS servers without it.

Step 3: Install WordPress on an Amazon Web Services (AWS) Cloud Server
The next step is to set up a cloud server that will run WordPress. You can do this with only a few clicks on the AWS Marketplace. The following are the steps:
1. Go to the AWS Marketplace and sign in.
2. Enter the search keyword "bitnami wordpress" in the top search field to find the Bitnami WordPress Stack.
3. From the list of search results, select the stack.
4. Review the description and hourly fees for various server sizes on the resultant detail page.To proceed, select the location in which you want to launch your server and click "Continue."
5. Review the details in the "1-Click Launch" tab on the subsequent page. This includes details on the server's configuration (CPU, memory, and storage), as well as an estimate of your monthly costs. You can choose between a "micro" server with one virtual CPU and a "large" server with up to 40 virtual CPUs.
6. On the same page, double-check that the server's key pair (generated in Step 2) is valid.
7. After you've confirmed your selection, click the "Launch with 1-Click" option.
8. The new server will now be spun up by the AWS Marketplace.
9. The procedure usually takes a few minutes, and you can monitor the server's status using the EC2 Dashboard. You can get the server's public IP address from the EC2 Dashboard once it's up and running, as illustrated below:
10. Now that you've done that, you should be able to browse to the cloud server by typing the cloud server's IP address directly into your browser's address bar.You should now see a sample post on the homepage of your WordPress blog.

Step 4: Obtain WordPress Login Information
You're almost done, but before you can log in to WordPress, you'll need administrator credentials. The steps to obtain the administrator password and username are as follows:
1. Open the AWS Cloud Console and log in.
2. Select the "Instances -> Instances" menu item from the left navigation bar.
3. From the dashboard, choose your instance.
4. Select the "Get System Log" menu item from the "Actions" drop-down menu.
5. Examine the system log by opening it. till you've discovered the application's password.

Step 5: Login and Get Started WordPress
To access the WordPress dashboard, you must first log in.will need to take the following steps:
1. Accessing the WordPress dashboard, which is generally accessed using the URL http://SERVER-IP/wpadmin.
2. Administrate your account credentials obtained in the previous stage
3. You should now be able to access the WordPress dashboard, which allows you to manage pages, comments, and posts, as well as personalise your website with various plugins and themes, add and delete new user accounts, control navigation menus, import and export material, and much more.
4. Now it's time to get WordPress up and running .Create your first website, update it with new posts and pages, and so on.
5. Use the WordPress update feature to keep your WordPress installations up to date.

B. System Architecture

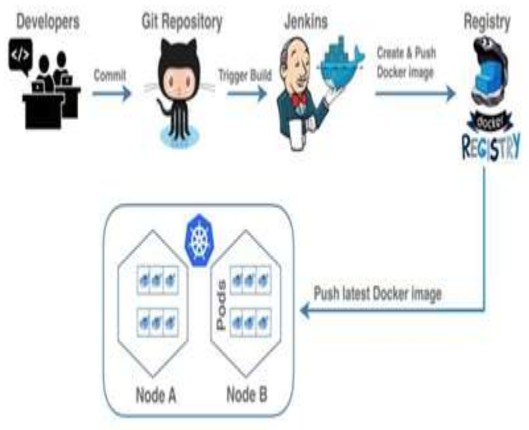

Fig. 2.System Architecture.

1. Different developers in various regions push their code from their local repository to Github, a remote repository.
2. The code is then built with Maven, resulting in the construction of a jar or war package.
3. Jenkins is used to automate the entire procedure. In Jenkins, a job is established with the responsibility of monitoring Github for new code and building it into the system.
4. With the help of S3, this jar file is released in a shared place called S3.
5. The testing team checks for errors (in Jenkins) at this stage, and if any are found, an error notice is displayed. The development team investigates the issue and re-pushes their work to github.
6. The testing team authenticates the production environment for quality assurance before distributing the jar file.
7. The production environment is built up with the Terraform tool, which creates a virtual private network (VPN), subnets, a route table, and an internet gateway.
8. The jar file is deployed in the servers after authentication.
9. Either a virtual machine or docker images could be used to deploy the code. We're going to use Docker images for this. Docker is used to create containers. Docker is used to solve the challenge of environmental change. Requirements are captured as images, which can then be transported to all phases.
10. A micro service is a Docker container that contains a certain number of containers. Kubernetes is a tool that is used to manage a large number of containers.Kubernetes is a container orchestration technology that allows containers to work together. Kuberentes is also used for resource scaling, containerized application management, and deployment automation.

C. Devops Tools

a) **Terraform :**

Terraform aids in the safe and effective building, maintenance, and configuration management of infrastructure. It's built to support and manage a large number of resources, including physical servers and SaaS products. It concentrates on the infrastructure's automation. It's also used to keep code from being duplicated.

b) **Git :**

During the software development lifecycle, Git is a version control technology that is used to submit code into a remote repository, such as Github.com. It's also used to keep track of file set modifications. Using git commands, developers push their code to a Github.com repository.

c) **Maven :**

Maven is a project management and comprehension tool that gives developers a comprehensive build lifecycle framework. The Project Object Tool (POM) file is the foundation for Maven. For project builds, dependency management, and documentation, POM is used. POM is an XML file found in the project's base directory as pom.xml. The POM file contains all of the project's relevant metadata and configuration details.

d) Jenkins :

Jenkins is used to carry out the continuous integration (CI) procedure. Jenkins is an open source automation server that aids in the automation of the software development lifecycle's manual tasks.

e) Docker :

Docker is a containerization technology that is used to bundle a programme and all of its dependencies in the form of a docker container to ensure that the application runs smoothly in all environments. A Docker Container is a pre-configured container that may be used to install a specific application or environment on the fly. Consider the following scenario. where code that runs on one machine does not run on another. This is related to changes in the environment. Docker is utilised to solve this problem. It's time to make a Docker image.

f) Kubernetes

Kubernetes is a container deployment and management platform. It's an open-source, production-ready architecture for deploying, scaling, managing, and composing application containers across clusters of servers. It is primarily intended for applications that are made up of several containers. It's a collection of containers organised into tightly and loosely connected forms utilising pods and labels for simple management and discovery.

D. Working Procedure:

Install the Git programme (for windows). Using php, create a single container website called "Pet Mitra." Launch two instances at first. Now use the "connect" option to connect to server 2. Terraform and Kubernetes must be installed. Create a directory called "terraform" after installing terraform, then move to that directory and create two files called "main.tf" and "variable.tf." Terraform init, terraform plan, and terraform apply are commands that can be used to run the terraform script.It generates VPCs, Subnets, Internet gateways, and Route tables. The IP address is provided externally by invoking the function "variable," and the VPC name, DNS support, and DNS host name are all provided externally.

Then construct four subnets, two of which are public (subnet1 and subnet2) and two of which are private (subnet3 and subnet4) (subnet3 and subnet4). Within the function, provide IP addresses for subnets, but call function "variable" to provide availability zone and vpc id.Make an internet gateway and connect it to the VPC. Create a route table and connect it to your internet gateway. Assign two public subnets to the public route table and two private route tables to the private route table now. As a result, the entire infrastructure is automated. As a result, vpc, subnets, route tables, and an internet gateway are created. Leave the server. Make a github account and make a public repository (say Pet-Mainproject-Kubernetes). Using the "connect" option, connect to server 1. Install the following software: git, maven, Jenkins, and Docker. Make a directory called "sourcecode." Using the "scp" command, copy all of the code from the website pet mitra into that folder.

To push code to a git repository, run all of the git commands (such init, add, and push) on the folder "sourcecode."Create two buckets in S3 (AWS), each with a different name, and enable versioning as well as all rights. The output of Jenkins is stored in one bucket, and the state of Kubernetes is stored in the other. Create a user with programmatic access in IAM (AWS) and provide them administrative access. Jenkins will create a secret key and an access key, which will be used to gain access to AWS. Open Jenkins server with that server's IP address and port 8080 accessible. Unlock Jenkins pops up and asks for the "initialadminpassword."

The initial admin password path will be displayed in the window. Copy the password from that location and paste it into the Unlock Jenkins window. Another window displays next, allowing you to install recommended or specific plugins. "Install suggested plugins" is selected here. The sign up page comes after the plugins have been installed. Use the needed credentials to register. Then select "start using Jenkins" from the drop-down menu. Jenkins dashboard then shows. Jobs can be created here. In the Manage Plugins area, download the git, maven, and S3 plugins. Create a new job (say "CI-petproject") by clicking on create new job. Then fill in the required information. It is optional to include a project description. Select "Git" for source code management and input the repository URL and "master" branch. Choose the required build trigger. Enter the command "mvn clean package" in the Build field. Select Publish Artefacts to S3 from the Post Build Action menu.

Fill in the S3 profile information collected during the creation of the S3 bucket and IAM user. The job will now build the code and upload it to the S3 bucket. In AWS, look into the S3 service. (Code has been deployed.) Create a repository (say "petproject") in dockerhub using your docker ID, password, and email address. To construct a Docker image, write a Dockerfile.

This Dockerfile contains docker image creation commands. It utilises the 16.04 image, updates the dependencies, and instals Apache. PHP and MySQL files should be downloaded. To make an image, download the contents of the Github repository and unzip it. Allow this website or code to operate in the foreground by exposing port 80. After creating the Dockerfile, run "sudo docker build –t xyz/petproject: 10.0" to build it. This command creates a Dockerfile and stores it in the url supplied Dockerhub repository. –t specifies the tag.Use the command "docker run –d –p 80:80 b596f2c49e16" to run the Dockerfile.After that, use the command "sudo docker push petproject:10.0" to push the docker image to the repository. As a result, the code is dockerized. To see the docker images built, run the command "docker images." To see the container that was built, use the command "docker ps." Exit from the server and search the dockerhub repository for a docker image with the tag 10.0.

Now use the "connect" option to connect to server 2. Create a bucket on S3 called project-kubernetescluster to set environmental variables and store kubernetes status information. Give the cluster the name "petproject.k8s.local." Now type "kops create cluster —node-count=1 —node-size=t2.small —master-size=t2.small —zones=us-west-1a —name=$KOPS CLUSTER NAME" to create a cluster. In zone us-west-1a, it builds one worker node of size t2.small and one master node of size t2.small.Then run the following command to build a Kubernetes cluster. The clustering tool "kops" is utilised, and the deployment tool "kubectl" is employed.The command "kubectl cluster info" displays detailed information about the formed cluster. Now, inside Kubernetes, deploy the dockerized image. Use the command "kubectl run petproject010 image=xyz/petproject:10.0 replicas=1 port=80" to launch the dockerized image. "petproject010" is a pod in this case. The pod is where the Docker image will run. As a result, pod will be created after you run this command. Execute the command "kubectl get pods" to learn about the pod details. A dockerized image is executing within the pod. Execute the command "kubectl get deployment" to learn more about the deployment.The load balancer should now be able to see the code deployment.

The command "kubectl expose deployment petproject010 type=LoadBalancername=my-project010" is used to accomplish this. Execute the command "kubectl get service" to acquire information about the load balancer. This command returns the load balancer's external IP address (a648de64b5e9a11e9a11906195251c80-138107180.us-west-1.elb.amazonaws.com). This is the last destination. If this IP address is visited, it will direct you to the pod where the website is executing in dockerized form. (Code from the Petmitra website is pushed to the git repository as input.) This load balancer link to access this website cannot be remembered in the real world. As a result, this link must be mapped to a domain name.

A domain name called "projectpet.tk" was obtained to make this website more accessible.Then build a public hosted zone called "projectpet.tk" in AWS' Route 53 service. By modifying the NS Record acquired in Route 53 to "projectpet.tk," a connection between freenom and AWS is established. Additionally, for the load balancer link, "A Record" is created with some additional credentials. Anyone attempting to visit "projectpet.tk" or the load balancer URL will be redirected to the pet mitra website. User requests for "projectpet.tk" now go to freenom, which redirects to Route 53. It redirects to the load balancer connection from Route 53, and the petmitra website appears.

IV. RESULTS

Fig.3. Code pushed from local to Github Repository after executing git commands

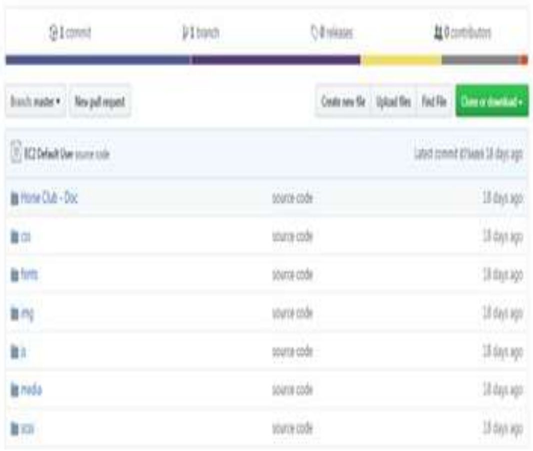

Fig.4.Jenkins creates a job called CI-PetProject, which builds the code and deploys it to S3.

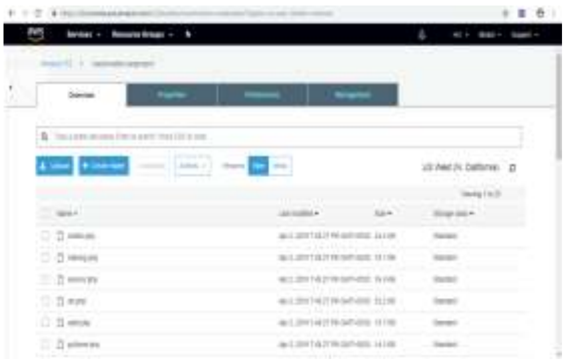

Fig.5.Code deployed to S3 by Jenkins job

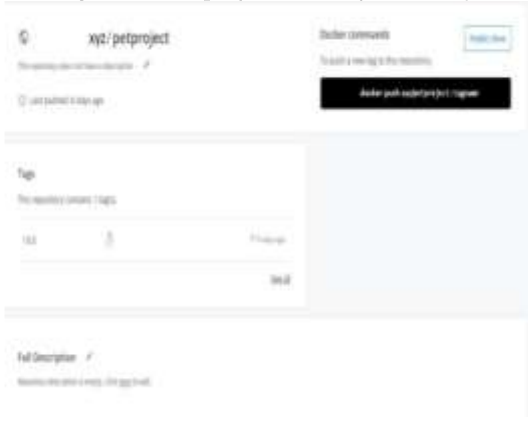

Fig.6.Docker image pushed into dockerhub repository petproject

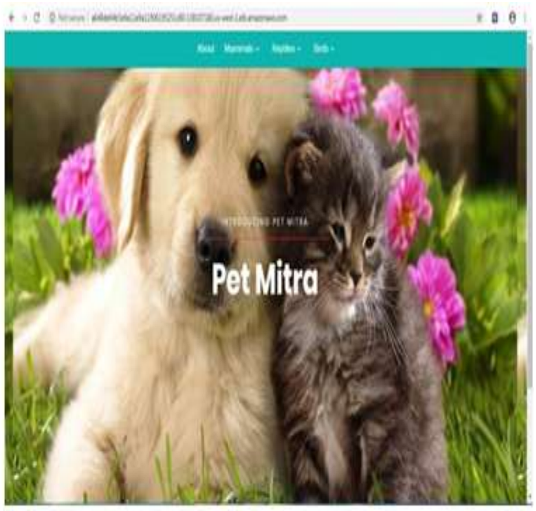

Fig.7. With the help of a load balancer link, a Docker image (which contains the code of the input website) is operating in a Kubernetes cluster (pod).

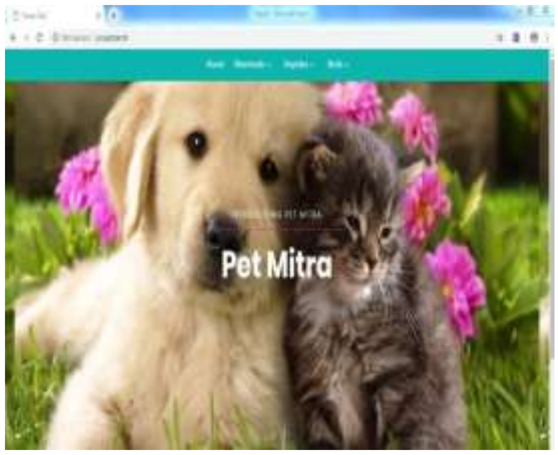

Fig.8.Website accessed via domain name "projectpet.tk"- the user request to "projectpet.tk" goes to freenom, from freenom it redirects to Route 53. From Route 53 it redirects to load balancer link and website appears

V. CONCLUSION

DevOps is a mechanism for improving development and operations teams' collaboration. Enabling DevOps enhances delivery speed based on business and consumer requirements. DevOps automation, in particular, boosts productivity, dependability, and process standardisation, all of which help firms deliver better products. Because the market environment in which we operate is constantly changing, processes must evolve over time. As a result, DevOps adoption in the The modern period facilitates industry operations and product delivery. This project aims to demonstrate the functionality of several tools such as git, maven, Jenkins, docker, terraform, and kubernetes by constructing a single container website.

A single container application is used in this paper to demonstrate how all of the technologies work. Multiple container applications could be developed in the future to demonstrate the operation of these tools as well as the load balancer. Because the goal of a load balancer is to keep the user request, also known as the load on the application, it determines which website access permissions should be granted to the user when numerous websites are running, as seen when employing multiple container applications.

REFERENCES

[1] Constantine Aaron Cois, Joseph Yankel, Anne Connell, "Modern DevOps: Optimizing software development through effective system interactions", IEEE International Professional Communication Conference (IPCC), 2014.

[2] Manish Virmani, "Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery", Fifth international conference on Innovative Computing Technology (INTECH 2015), 2015.

[3] Juhoon Kim, Catalin Meirosu, Ioanna Papafili, Rebecca Steinert, Sachin Sharma, Fritz-Joachim Westphal, Mario Kind, Apoorv Shukla, Felician Nemeth, "Service provider DevOps for large scale modern network services", IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015.

[4] Felicián Németh, Rebecca Steinert, Per Kreuger, Pontus Sköldström, "Roles of DevOps tools in an automated, dynamic service creation architecture", IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015.

[5] Hasan Yasar, Kiriakos Kontostathis, "Secure DevOps Process and Implementation", IEEE Cybersecurity Development (SecDev), 2016.

[6] Christof Ebert, Gorka Gallardo, Josune Hernantes, Nicolas Serrano, "DevOps", IEEE Software, 2016.

[7] Shahin, Muhammad Ali Babar, Liming Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices", IEEE 2016.

[8] Matt Callanan, Alexandra Spillane, "DevOps: Making It Easy to Do the Right Thing", IEEE Software, 2016.

[9] M Rajkumar, Anil Kumar Pole, Vittalraya Shenoy Adige, Prabal Mahanta, "DevOps culture and its impact on cloud delivery and software development", International

Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), 2016

[10] Elisa Diel, Sabrina Marczak, Daniela S. Cruzes, "Communication Challenges and Strategies in Distributed DevOps", IEEE 11th International Conference on Global Software Engineering (ICGSE), 2016.

[11] Hui Kang, Michael Le, Shu Tao, "Container and Microservice Driven Design for Cloud Infrastructure DevOps", IEEE International Conference on Cloud Engineering (IC2E), 2016.

[12] S. Palihawadana, C. H. Wijeweera, M. G. T. N. Sanjitha, V. K. Liyanage, I. Perera, D. A. Meedeniya, "Tool support for traceability management of software artefacts with DevOps practices", Moratuwa Engineering Research Conference (MERCon), 2017.

[13] Wolfgang John, Guido Marchetto, Felician Nemeth, Pontus Skoldstrom, Rebecca Steinert, Catalin Meiros, Ioanna Papafili, Koastas Pentikousis, "Service Provider DevOps", IEEE Communications Magazine, 2017.

[14] Zhenhua Li, Yun Zhang, Yunhao Liu, "Towards a full-stack devops environment (platform-as-a-service) for cloud-hosted applications", Tsinghua Science and Technology, 2017.

[15] Arachchi, Indika Perera, "Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management", Moratuwa Engineering Research Conference (MERCon) 2018.

[16] Aayush Agarwal, Subhash Gupta, Tanupriya Choudhury, "Continuous and Integrated Software Development using DevOps", International Conference on Advances in Computing and Communication Engineering (ICACCE- 2018) Paris, France 22-23 June 2018.

[17] Thomas F. Düllmann, Christina Paule, André van Hoorn, "Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines", IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE), 2018.

[18] Lianping Chen, "Microservices: Architecting for Continuous Delivery and DevOps", IEEE International Conference on Software Architecture (ICSA), 2018.

[19] Aayush Agarwal, Subhash Gupta, Tanupriya Choudhury "Continuous and Integrated Software Development using DevOps", International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018.

[20] Barry Snyder , Bill Curtis, "Using Analytics to Guide Improvement During an Agile/DevOps Transformation", IEEE Software, 2018.

[21] Len Bass, "The Software Architect and DevOps", IEEE SOFTWARE 2018.