IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 09, Issue 6 (June. 2019), ||S (III) || PP 06-09

Application deployment in Cloud: Docker & Kubernetes

Mona Deshmukh[1], Abhijeet Salunkhe[2]
[1]*Associate Professor, MCA, VESIT*
[2]*Student Final Year, MCA, VESIT*

Abstract: Microservice depicts the new Architectural style where small and loosely coupled modules can be developed and deployed instead of developing the composite Application. This has its own benefits such as maintainability, Flexibility in scaling Decrease in downtime for failure and upgrades. Containerization is new alternative for full machine virtualization, it provides many benefits as the application can be run on any suitable physical machine without any worries about dependencies. Containerization is known these days by the open-source Docker. Docker is used for build, ship and run application anywhere. Kubernetes is an Open source platform available for automation, scaling, and Deployment of Container management.Ansible is an It automation tool designed for cloud provisioning, application development and intra service orchestration.

Keywords: Kubernetes, Container, Docker, Ansible, Microservices, Cloud, Virtualization, Orchestration.

Date of Submission: 11-06-2019 Date of acceptance: 28-06-2019

I. INTRODUCTION

Microservice is an Architectural style that structures the application as collection of services that are organised, Loosely coupled, Deployable & maintainable and testable.[4] This kind of architectural design is most widely used these days and has its own benefits, But it is difficult to handle the orchestration between more than one microservices when you have many microservices running parallely.[3] These microservices are running in container as an individual entity and there is no communication done between them, These container can be any Container or Docker container. Kubernetes is nothing but an platform to manage container. Ansible is an Automation tool which can be leveraged to implement the multi node Kubernetes cluster for development purpose, This production is similar to the production like cluster that can be setup on the local machine.[1]

II. MICROSERVICE

Microservice architecture, or just microservices, is a distinctive technique of developing software systems that tries to specialise on building single-function modules with well-defined interfaces and operations. The trend has grown popular in recent years as Enterprises look to become more Agile and move towards a DevOps and continuous testing. MicroSefice has many benefits for Agile and Devops team. Unlike microservices, monolithic application is build as a single application unit, which makes it difficult to handle and make changes, Even smallest of change in the code may cause the Developer to build and deploy the whole code again. Scaling a specific function means scaling the entire application for the same. Microservice solve the problem of monolithic application by being modular as possible. Each microservice is independent of each other can run independently without any interface, In the simplest form they help create as a suite of small services, These service may be written in different programming language and may use different programming language.[4][5]

❖ **Some of the key feature of the Microservices:**

1) **Small and focused** : Microsevice needs to be small as they need to focus on the unit of work. There are no rules on how small they should be. Microsevice should be considered as application or product.
2) **Loosely coupled** : Loose coupling is an absolutely essential characteristic of microservices. You need to be able to deploy a single microservice on its own. There must be zero coordination necessary for the deployment with other microservices. This loose coupling enables frequent and rapid deployments, therefore getting much-needed features and capabilities to the consumers.
3) **Language-neutral** : There is no barrier for language when it comes to Microservice, Any language which is suitable for the give application should be used.Microservices are composed together to form a complex application, and they do not need to be written using the same programming language.

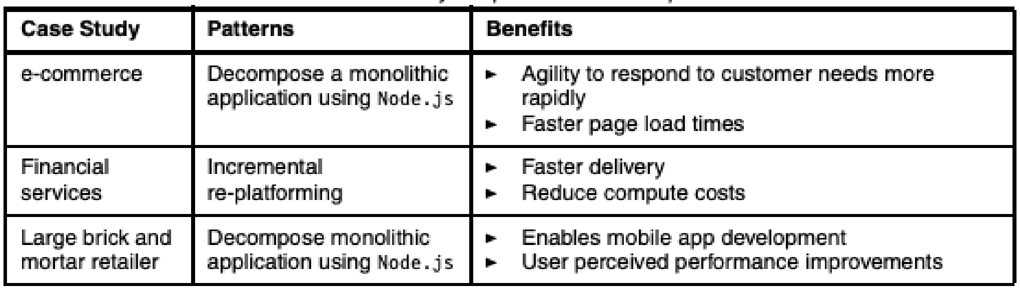
4) **Bounded context** : Bounded context is that a particular microservice does not "know" anything about underlying implementation of other microservices surrounding it. If for whatever reason a microservice needs to know anything about another microservice, you do not have a bounded context.

❖ **Benefits from microservices**

1) **Developer perspective** : It enables you to avoid the large code base making it easier to maintain or add features. Improves deployment time. Making debugging easier. Simplifies code tracking.
2) **Tester perspective** : Decrease the testing time. No need to restart the the container.
3) **Business owner perspective** : Allows frequent delivery and faster delivery.

❖ **Case Study**

Table 1-2 Case Studies and most commonly adopted architectural patterns



Case Study	Patterns	Benefits
e-commerce	Decompose a monolithic application using `Node.js`	► Agility to respond to customer needs more rapidly ► Faster page load times
Financial services	Incremental re-platforming	► Faster delivery ► Reduce compute costs
Large brick and mortar retailer	Decompose monolithic application using `Node.js`	► Enables mobile app development ► User perceived performance improvements

III. CONTAINERIZATION & DOCKER

Application Containerization refers to OS level virtualization method which is used to run and deploy the application without launching the entire virtual machine for each application. Application container includes the necessary components required to run the application such as environment variables, files, libraries,etc. The complete set of information to execute in a container is image file.[9]

The most common container application is Docker. Docker is a platform which packages an application and all its dependencies together in the form of containers. This containerization aspect of Docker ensures that the application works in any environment. Application containerization works with microservice and distributed applications because each container operates individually and others uses minimal resources. Each microservice communicate with other microservice with application programming interface.It is possible to handle more than one container instead of microservice running on the same platform. It also helps during scalability and Reusability of the function. Portability is another benefit as long as the application is running on the same OS.[10]

❖ **Basic Concept of Docker**

1) **Docker File :** A Dockerfile is a text document which contains all the commands that a user can call on the command line to assemble an image. So, Docker can build images automatically by reading the instructions from a Dockerfile.
2) **Docker Image :** Docker Image can be compared to a template which is used to create Docker Containers. So, these read-only templates are the building blocks of a Docker Container.
3) **Docker Container :** Docker Container is a running instance of a Docker Image as they hold the entire package needed to run the application. So, these are basically the ready applications created from Docker Images which is the ultimate utility of Docker.

IV. KUBERNETES

Kubernetes is all about orchestration of containers.Kubernetes is also written as K8s which means a pilot in Greek. So, it is the captain of the ship with all the containers running inside. Its container management responsibilities include container deployment, scaling & descaling of containers & container load balancing. Kubernetes is an open- source platform for managing containerized workloads, services and for automating deployment, scaling and orchestration of containerized applications. Kubernetes is all about running multiple connected containers all organized in pods. Just like docker is the de-facto standard for containers, kubernetes is the de-facto standard for orchestration of containers. Kubernetes does not limit the supported application types. Kubernetes has the power to support a variety of workloads like stateful, stateless and data processing workloads. If the container is able to run the application, kubernetes can definitely run the application. The main application of kubernetes is for orchestration of containers. We can deploy containers using kubectl as well deploying web app. We can create ingress routing and run stateful services on kubernetes.[6]

❖ **Features of Kubernetes**

1) **Automatic Bin-Packing** : Kubernetes automatically packages the application and schedules the containers based on the requirements. To ensure complete utilization and save unused resources, Kubernetes balances between critical and best-effort workloads .
2) **Load Balancing and Service Discovery** : With Kubernetes, there is no need to worry about networking and communication because Kubernetes will automatically assign IP addresses to containers and a single DNS name for a set of containers, that can load-balance traffic inside the cluster.
3) **Storage Orchestration** : With Kubernetes, you can mount the storage system of your choice. You can either opt for local storage, or choose a public cloud provider such as GCP or AWS, or perhaps use a shared network storage system such as NFS, iSCSI, etc.
4) **Self-Healing** : Kubernetes can automatically restart containers that fail during execution and kills those containers that don't respond to user-defined health checks. But if nodes itself die, then it replaces and reschedules those failed containers on other available nodes.
5) **Secret and configuration Management** : Kubernetes can help you deploy and update secrets and application configuration without rebuilding image and without exposing secrets in your stack configuration.
6) **Automatic Rollbacks and Rollouts** : Kubernetes progressively rolls out changes and updates to your application or its configuration, by ensuring that not all instances are worked at the same instance. Even if something goes wrong, Kubernetes will roll back the change for you.[12]

❖ **Ansible**

Ansible is an Automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.The main reason using Ansible as configuration management tool is the fact that it is designed for multi-tier deployment. Ansible represents the information technology infrastructure by narrating how the servers interrelate rather than just one single server at a time. A key concept of Ansible is the use of inventory files which is considered to be a single point of truth. Moreover, cloud infrastructure consists of virtual machines which come and go very frequently, so maintaining a single inventory file in such a dynamically provisioned environment is cumbersome. But Ansible has all the features to handle such situations right from dynamic inventory files to full automation modules. Since Ansible is written in python, it makes it extremely portable and flexible. Ansible provides a production like structure which is similar to the production like cluster that can be set on the local machine. Minikube do not provide the opportunity to work with the multi node cluster, The multi node cluster has its own benefits such as the ops can reproduce the issue in the multi node cluster environment. tester can deploy multiple versions of the same application to test cases and verify the changes , this makes the ansible more agile.[2]

V. CONCLUSION

Docker and Kubernetes works at different levels. Docker is used in It software that helps you to generate different types of Linux Containers. The Docker technology makes the use of the Linux kernel and various features of Linux, Because the entire linux operating system is been separated from the other running applications which makes the process or working of the Docker faster and accurate Kubernetes is the most popular container orchestration system and it was designed specifically with Google Cloud Platform integration in mind. The major advantage of using Kubernetes is orchestrating between many applications. It is also very useful for scaling many applications at very less time. By using this we can save our cost and time. So, it gives us management capabilities, saving cost, proper used of all resources. Kubernetes allows us to deploy and manage the application running multiple host using the docker. Kubernetes enables healing through its failure recovery actions and they are often evaluated through internal operations. Ansible can be used to create a multi node node cluster which is similar to production like cluster, For the instance the Ops can reproduce an issue in multi node cluster. Testers can deploy multiple versions of an application for executing test cases and verifying changes. These benefits enable teams to resolve issues faster which make the more agile.

REFERENCES

[1]. https://kubernetes.io/
[2]. https://www.ansible.com/overview/how-ansible-works
[3]. https://www.webopedia.com/TERM/C/containerization.html
[4]. https://microservices.io/
[5]. https://microservices.io/patterns/microservices.html
[6]. https://kubernetes.io/blog/2019/03/15/kubernetes-setup-using-ansible-and-vagrant/
[7]. https://smartbear.com/solutions/microservices/
[8]. https://www.edureka.co/blog/what-is-microservices/

[9]. https://searchitoperations.techtarget.com/definition/application-containerization-app-containerization
[10]. https://www.docker.com/products/container-runtime
[11]. http://www.redbooks.ibm.com/redbooks/pdfs/sg248275.pdf
[12]. https://www.edureka.co/blog/what-is-kubernetes-container-orchestration?utm_source=medium&utm_medium=content-link&utm_campaign=kubernetes-architecture#KubernetesFeatures
[13]. https://www.edureka.co/blog/docker-explained/

Mona Deshmukh. "Application deployment in Cloud: Docker & Kubernetes." IOSR Journal of Engineering (IOSRJEN), vol. 09, no. 06, 2019, pp. 06-09.