IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Containerization In Today's World

Anshika Sharma [#1], Dr. Rakesh Kumar Dwivedi [#2], Dr. Ashendra Kumar Saxena [#3]

[#] *College of Computing Sciences and Information Technology,*
Teerthanker Mahaveer University, Moradabad, UP, India.

Abstract: Containers, a light weight Operating System virtualization technology is used to run any program from a small microservice or software process to a larger application. It also facilitates the deployment as well as management of larger applications. In such larger application deployments, multiple containers can also be deployed as one or more container clusters. Such clusters might be managed by container orchestrators such as Kubernetes, Docker swarm, EC2, Amazon EKS, OpenShift, Mesos etc. The main aim of our report is to provide a comparative study of containers, its uses in industry, security, performance, hardware support, orchestration tools, etc.

Keywords: Containerization, Containers, Orchestration Tools, Docker

1. INTRODUCTION

Container technology was bought up earlier in year 1979 with Unix version 7 and the chroot system. This System isolates a process by restricting an application's access to a specific directory, where this directory comprised of a root and child directories.

Even though container technology had remarkable improvements from year 2000-2011, the introduction of Docker Technology by Solomon Hykes revolutionized everything and single-handedly led to a Renaissance in container technology. Docker has built its foundation on two systems they are Linux containers (LXC) and the lib containers.

When docker technology came into the existence the big giant Companies like Target, Walmart, Chase Bank totally relied on Container technology which traditionally uses servers and they added many of them which were used to handle the increasing number of requests from the users.

After a few years the company named as VMware came into existence which allowed to run multiple O.S on the same host that were running isolated on same server, It was too a game-changer for many industries but was expensive too as well as it requires lot if resources like RAM/CPU.

In this modern advancing world container are used a lot as they are light weighted Operating System virtualization technology in which larger applications are run in isolated environment. They too require a less disk space due to the common layers sharing across images and they share a kernel across containers so they use less amount of memory which makes them fast. It provides us a microservice architecture which means in this small application can run independently.

The container technology has bought up an influential change in year 2017 with the introduction of Kubernetes as a technology which was a highly effective orchestration tool. After this it becomes industry standard with the combination of other container tools with Kubernetes like pods which leads to further more improvement in container technology.

In Today's Era containers are used by mostly every company in websites, application deployment etc. and even all the cloud we use most of the service are built over the container as a technology like Amazon EC2, etc. A number of Container orchestration technology tools have been flooded into the market which bought up a huge revolutionary change into every type of industry whether E-commerce, banking, social networking, IT industries, etc. and made their work easier, portable and convenient. As with the increasing of data and user's day by day the containerization was the need of technology to handle the number of requests from the users.

.

Containers are known for its high performance because of its:

- **Efficiency-** Containers always allow any application/program to be more rapidly deployed, patched, or scaled. Container `platforms like Docker, pods are able to drive real business impact thanks to their application density—up to 4x greater than that of virtual machines which uses virtualization, without sacrificing app performance.
- **Flexibility**- Containers are very much flexible to use as they have the liberation of applications from their environment, and the freedom to host them wherever suits your business we can set any environment according to our need and requirements.
- **Security-** Containers are more secure as they isolate apps/OS from your infrastructure, as well as from other applications, which reduces the exposed surface area and keeps us more secure.

2. SERVICES OFFERED BY CONTAINERIZATION

Containers Architecture and Management

Containerization allows lightweight virtualization techniques through the bespoke construction of containers as application packages from individual images that utilizes less resources and time. Containers have huge ability to deploy, test, and run applications to a larger number of severs in a split of time.

1) **Container Cluster Architecture** – Tools like Docker are built around the engine where these containers act as portable means to package applications. A PaaS cloud are built and executes through the workflows from the plan through agents like a container engine. Mostly the containers are based on the Linux LXC technology. Recent LXC provides the kernel mechanisms like namespaces and cgroups to isolate process on shared operating system. The Docker system consists of file system layered over each other using the LXC mechanisms as shown in figure below. In this each of the system consists of several hosts which are termed as nodes and also each node is grouped into interconnected container. In these services also allow scaling the application across different host nodes.

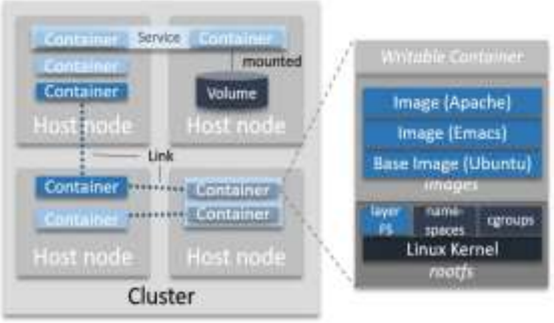


Fig. 1.1 Container Cluster Architectures

1) **Cloud-based Container Architectures –** The Container orchestrations deal not only to start or stop the containers but they are also defined as constructing and managing the distributed clusters of container-based software applications. The Orchestration tools also take care for the availability, networking, and even scaling of the containers. Similarly, Cloud-based containers is also the form of orchestration with the distributed cloud Environment like the Amazon EC2. The Cloud has a tiered and distributed architecture with core infrastructure platform and software application distributed across multi-cloud environment. Popular microservices architecture can be realized in this cloud framework through containers.

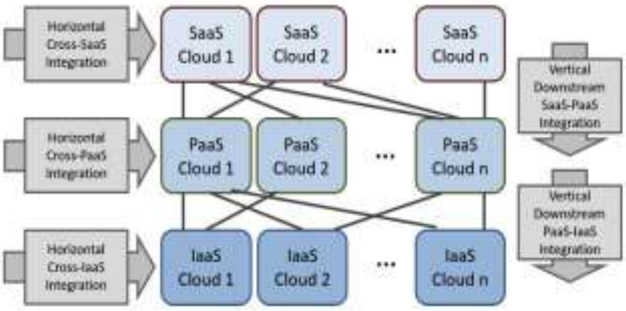

Horizontal Cross-SaaS Integration
SaaS Cloud 1
SaaS Cloud 2
...
SaaS Cloud n
Horizontal Cross-PaaS Integration
PaaS Cloud 1
PaaS Cloud 2
...
PaaS Cloud n
Horizontal Cross-IaaS Integration
IaaS Cloud 1
IaaS Cloud 2
...
IaaS Cloud n
Vertical Downstream SaaS-PaaS Integration
Vertical Downstream PaaS-IaaS Integration

Fig.1. 2: Cloud Reference Architecture Model

Container Runtimes

Docker was the first and foremost open-source container offering which came in year 2013 introduced by Solomon Hykes has led a great change in Industry. Now Kubernetes is evolving as the new standards for the deployment and also the management of the cluster.

Kubernetes initially supported the docker containers and rkt i.e., “rocket” through the custom codes. But recently with the creation of CRI-O i.e., Container Runtime Interface we have many ways to store virtual machines and at the same time they communicate through the interface.

1) Docker

Docker is an open-source platform that runs the application and even makes the process easier to develop and distribute. The container in this keeps running isolated on the top of operating system's kernel. Docker is among the best innovations as it accompanies new capacities that prior technology didn't have. In this the applications can be run easily and can be packed into the container by developer. Docker containers can easily be deployed into the cloud Environment and provides a huge advantage to automate the applications. In this there are four main internal components of docker i.e., Docker Client and Server, Docker Images, Docker Registries and Docker Containers.

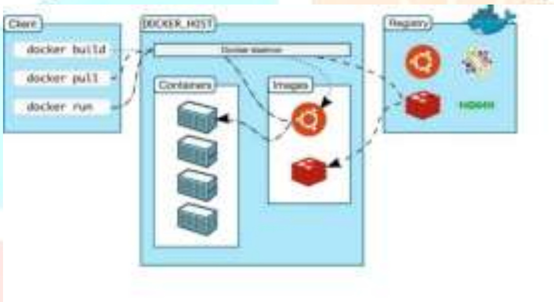


Fig. 1.3: Docker Architecture

2) Docker Enterprise

This set of extensions adds features to the Docker and make it possible for the Company to add the Commercial support from the industry. It helps in supporting the matrix and to know which version we are using and of what software are supported. It also provides us with the contact number if the things go wrong you may contact to their executives. This version of docker is mainly used by the industry officials and this is the paid version of the docker. The best benefit of this Edition is Certification from docker.

3) CRI-O

The first implementation of this CRI-O (Container Runtime Interface) is as an immensely lightweight, open-source reference implementation of the Kubernetes CRI (Container Runtime Interface) to enable using OCI (Open Container Initiative) Compatible runtimes. It is a light weight alternative for the Docker as the runtime for Kubernetes. At present it supports the runc and Kata Containers as the container runtimes but any OCI-conformant runtime can be plugged in principle.

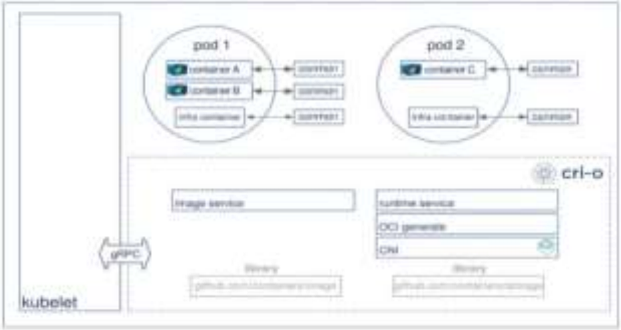


Fig. 1.4: CRI-O Architecture

4) Rktlet

The rkt was redesigned and retooled to use the CRI (Container Runtime Interface) as rktlet. It has a set of supported tools and community too to rival the Docker Community. Rkt is an ongoing CNCF efforts to develop a pod-native container runtime. The main aim for this project was to supplant the rkt package in the main Kubernetes repository.

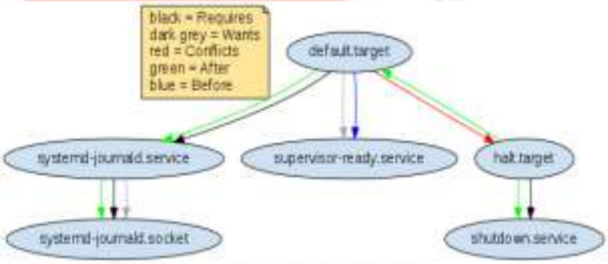


Fig. 1.5: Rktlet Architecture

5) Containerd

Containerd is a project of the Cloud Native Computing Foundation (CNCF) is an early container format. So recently the developers of the containerd built a CRI plugin which lets the Kubernetes to run containerd in the same format as the way it runs the rktlet or the CRI-O. It manages the complete container lifecycle of its nodes, from image transfer and storage to container execution and supervision to low-level storage to network attachment and beyond.

Container Orchestration

Container Orchestration refers to the process of automating the deployment and management of container based microservice application across the multiple clusters. A huge variety of Container Orchestration tools have become essential in deploying microservice based applications. These tools manage workloads, including moving instances from one node to another based-on load, and allocate resources such as CPU and memory.

1) Kubernetes

Google Kubernetes is an open-source container cluster manager that is used to manage containers, including the way they can scale and become more resilient. It can also schedule containers, allocate them, deploy, scale and manage disk space and storage. Kubernetes sets the standard for what a container cluster manager should be doing. As such, it has the largest market share, based on what I'm seeing out there. But some surveys show Google that coming in second to Docker, with Google establishing a larger presence in larger number of enterprises. Most companies are looking for the levels of adoption, as well as the relevance of a given standards to Kubernetes. Now Kubernetes is supported by the MNC's like Amazons AWS, Google Cloud Engine and Microsoft's Azure Container service, Kubernetes is more portable which prevents the vendor lock-in.

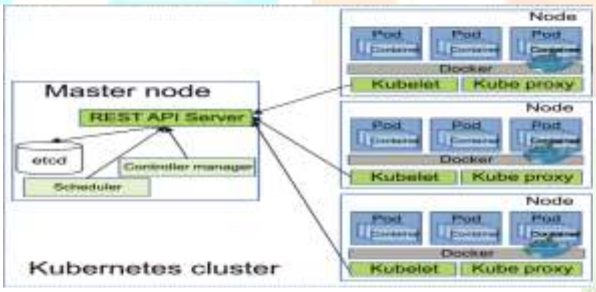


Fig. 1.6: Kubernetes Cluster Architecture

2) Docker Swarm

The Docker Swarm is a free product for cluster manages similarly like the Kubernetes offers like clustering, scheduling, and integration capabilities which makes build and ship multi-container or multi-host distributed applications. It also performs all of the necessary management and scaling for the container-based system. It doesn't support the concept of autoscaling or load balancing. The activities of the cluster are controlled by the swarm manager, and machines which have joined the cluster are referred to as nodes. It mainly has two types of services i.e., replicated and global. The docker swarm has 3 different types of nodes, each with a different role within the docker swam ecosystem, they are Manager Nodes, Leader Nodes, Worker Nodes. It helps guarantee the high service availability, automated load balancing and also leverages the Power of Containers.

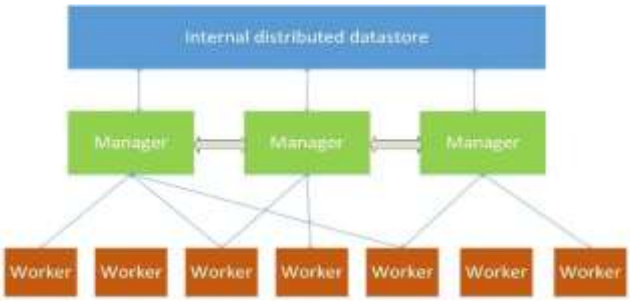


Fig. 1.7: Docker Swarm Architecture

3) Amazon Elastic Kubernetes Service (EKS)

It is also a well-managed service like the Kubernetes. It features the managing, securing, balancing and scaling the containerized applications. This Cluster runs in AWS fargate in different zones which computes the containers without ant server. It is a Amazon Services used in the AWS which is somewhere similar to the Kubernetes so, as a result Kubernetes based Applications can also be easily migrated to the Amazon EKS without any code refactoring. They come from community as well as AWS tools like AWS Application Load Balancers, Auto Scaling and Route 53.

EKS integrates with many services like the Amazon Virtual Private Cloud (VPC), Amazon CloudWatch, AWS Identity, Access Management (IAM) and Auto Scaling. They support distributed infrastructure management in multiple availability zones and make the services scalable and highly available. The EKS Architecture for the Control Plane and worker node Communication is shown in the figure below.

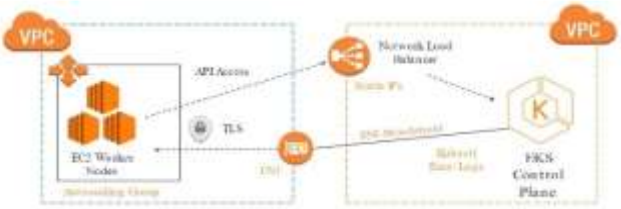


Fig. 1.8: EKS Architecture

3. Applications of Containerization

- **Containerization allows "Lift and shift" of existing applications into modern cloud architectures:**
  Lift and shift migration also known as rehosting is done by some organizations to use containers to migrate existing applications into more modern environments. In simple words, it helps to lift an application from old environment and helps to shift it to cloud without any change in code. It therefore requires minimal effort to move an application and helps in faster migration and deployment.
- **Containerization refactors existing applications for containers:**
  Refactoring can be termed as when the architecture of the application is reconstructed to meet the needs of the business case. A new perspective is taken on the application's architecture and development basis and if there is an urgent need to add features, performance increase, or scalability, we try to refactor the application knowing that these are difficult to achieve in the existing application's environment which helps to provide high level of optimization and performance. In simple terms, refactoring is the method of customizing any application to run on a cloud infrastructure platform. It therefore helps to achieve the full benefits of a container environment.
- **It helps to develop new container-native applications:**
  Application containers, such as Docker, allows the files, dependencies and libraries of an application to run on an operating system. It enables the user to create and run a separate container for multiple independent applications, or multiple services that constitute a single application. Just like refactoring, this approach is used to utilize the full benefits of containers.
- **It provides better support for microservices architectures:**
  Containers offer a lightweight runtime environment for an application, providing a consistent software environment that allows the application to run in a smoothy way from the developer's desktop to testing to final production deployment. Therefore, distributed applications and microservices can be more easily isolated, deployed, and scaled using individual container building blocks.
- **It provides DevOps support for continuous integration and deployment (CI/CD):**
  Containerization technology supports efficient build, test, and deployment from the same container images in just split number of seconds which helps to deliver better quality applications faster and with better compliance.

4. ADVANTAGES OF DISADVANTAGES OF CONTAINERIZATION

Advantages of Containerization

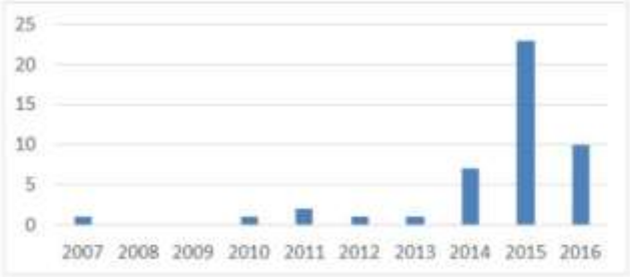
- Containers Are lightweight: As compared to virtual machines containers are light weighted and therefore consumes less on server resource usage. This happens as containers share the host operating system.
- Reduced cost of operations in infrastructure: As many containers usually runs on a single VM, it helps an organization to save money and also helps them to achieve much more on a single system using containerization.
- Solution scalability on the microservice/function level: Containers provide a light weight environment due to which distributed applications and microservices can be more easily isolated, deployed, and scaled using individual container building blocks. That's why there is no need to scale instances/VMs.
- Better security: Containers provides full application isolation which allows to run each application's major process in separate containers.
- Easily portable: Containers provides full portability between clouds and on-premises locations and therefore allows the application to run in a smoothy way from the developer's desktop to testing to final production deployment.
- Faster computation: Containers are always ready to compute as they launch, boot and stop within seconds in comparison to VMs.

Disadvantages of Containerization

- Since containers share the kernel of the host OS, so if the host OS becomes vulnerable, containers become vulnerable too.
- Since containers are isolated, the networking part while working with containers can be quite tricky. Therefore, developers need to maintain a good network connection in the containers.
- The larger monolithic applications can consume gigabytes of storage and memory. Since these are so large, they make containers less ideal for deployment and execution.
- Since containerization is a relatively new technology, organizations may not have IT expertise available to configure containers properly.
- Containerization works well more efficiently for Linux operating systems as compared to windows operating systems.
- Monitoring containers can be quite harder task as containerization is generally used to build multi-layered structure, with one application in another container. Therefore, developers need to monitor more things than they would have to if applications were running in virtual machines.

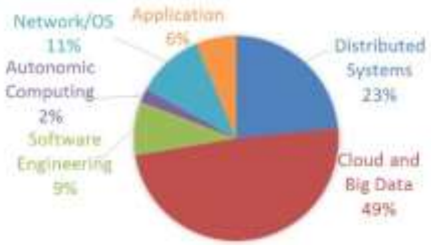
5. TRENDS IN THE GROWTH OF CONTAINERIZATION

In the study of the container orchestration after the LXC introduction there has been influential change after the emergence of this container technology tool Docker. According to the Research papers earlier there was not much open-source technologies in existence that research could consider. The emergence of container technology as shown below according to graph shown below.



Graph 1.1: Year-wise Use of Container Technology

The Container as a technology is also used in various communities and according to the studies and research paper titled as "Cloud Container Technologies: a State-of-the-Art Review" the study distribution by various communities are depicted below.



Graph 1.2: Community wise use of containers

Container technology is the very powerful tools and is now used in all of the platforms in all technologies whether IoT, AI, Machine Learning, Website, etc. Containerization is now used In all of the field in all industry like "Pokémon Go" was deployed on Google cloud platform uses the Kubernetes clusters. Possible Containerization application could be the gateway in the IoT due to its ability of working as an end-point for user data presentation and even light weighted clusters. Containers are also used in the Microservices-Architecture. Microservices mainly consist of multiple scalable, testable and deployable small containers that together combines to constitute a complex software system.

6. CONCLUSIONS

This report gives a review of the Containers which is a light weight Operating System virtualization technology used to run any program or an application from a small microservice or software process to a larger application. Containers are fast, portable and secure and are used in wide range of fields such as in Cloud computing, Edge Computing, for building IaaS platform to build interactive social media platforms. Containerization has bought up a huge revolution by becoming a huge trend in software development as well as an alternative to the virtualization technology. It also facilitates the deployment as well as management of larger applications. In such larger application deployments, multiple containers can also be deployed as one or more container clusters. Such clusters might be managed by container orchestrators such as Kubernetes, Docker swarm, EC2, Amazon EKS, OpenShift, Mesos etc. This technology allows the user/developer to create and deploy applications faster with more security. The present study talks about several promising aspects of containerizations with detailed technicalities involved in their successful industrial implementation, security & isolation, smooth management and orchestration. The report also provides a comparative study of containers, its uses in industry, security, performance, hardware support, orchestration tools, etc.

REFERENCES:

[1] Junzo Watada, Arunava Roy, Ruturaj Kadikar, Hoang Pham, Bing Xu. "Emerging Trends, Techniques and Open Issues of Containerization: A Review", IEEE Access, 2019.
[2] https://techbeacon.com/devops
[3] "Advances in Computing and Network Communications", Springer Science and Business Media LLC, 2021
[4] Baasanjargal Erdenebat, Tamas Kozsik."Adoption of cloud and containerization technologies in Mongolia", 2019 IEEE 15thInternational Scientific Conference on Informatics, 2019
[5] Dimple Patel, Manoj Kumar Patra, Bibhudatta Sahoo. "Energy Efficient Genetic Algorithm for Container Consolidation in Cloud System",2020 7th International Conference on Signal Processing and Integrated Networks (SPIN),2020.
[6] Narisu Cha, Celimuge Wu, Tsutomu Yoshinaga, Yusheng Ji, Kok-Lim Alvin Yau."Virtual Edge: Exploring Computation Offloading in Collaborative Vehicular Edge Computing", IEEE Access, 2021
[7] Tamanna Siddiqui, Shadab Alam Siddiqui, Najeeb Ahmad Khan. "Comprehensive Analysis of Container Technology", 2019 4thInternational Conference on Information Systems and Computer Networks (ISCON),2019

[8] www.searchitoperations.techtarget.com
[9] Bellishree, Dr. Deepamala. N research paper tiltled "A Survey on Docker Container and its Use Cases"
[10] www.netapp.com
[11] bia.unibz.it
[12] www.mafiadoc.com
[13] www.cnblogs.com
[14] www.wikipedia.com
[15] www.kscst.iisc.ernet.in