Containerization gained prominence with the open source Docker, which developed a method to give containers better portability -- allowing them to be moved among any system that shares the host OS type without requiring code changes. With Docker containers, there are no guest OS environment variables or library dependencies to manage.

Proponents of containerization point to gains in efficiency for memory, CPU and storage as key benefits of this approach, compared with traditional virtualization. Because containers do not have the overhead required by VMs -- separate OS instances -- it is possible to support many more containers on the same infrastructure. As such, containerization improves performance because there is just one OS taking care of hardware calls.

A major factor in the interest in containers is they can be created much faster than hypervisor-based instances. This makes for a much more agile environment and facilitates new approaches, such as microservices and continuous integration and delivery.

Docker containers were created for fast and reliable deployment of self-contained application components on any underlying infrastructure. However, applications are typically multi-tier in their architecture, which means that each container has dependencies which need to be managed properly.

Docker orchestration will take care of the timing of container creation by order of dependency, as well as all of the necessary configurations to allow the containers to communicate and pass the required runtime properties to one another.

At Cybage, we have helped many clients in containerization and Orchestration of their application and infrastructure using below tools ecosystem:

