CONTAINER ORCHESTRATION IN MODERN AGE

In this new era of microservices, cloud-based infrastructure, continuous integration, and continuous deployment; It becomes crucial to know the importance of **containerized architecture** when compared with the native and legacy infrastructure running on Virtual Machines.

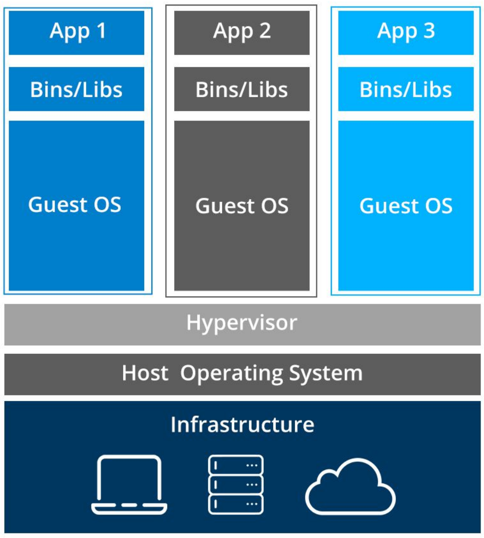
In a non-containerized world, one needs to spin-up virtual machines(VMs) and do a lot of interaction with cloud providers over their APIs, which results in time consumption and complexity. However, on the contrary, containerized environments need a consistent and automated method of defining resources, monitoring, alerting, building, testing, adding a new microservice, and the testing environment in an automated fashion to make the job simple and smooth. With containerization, one need not worry about the recurrent issues related to server memory, computing load, provisioning, and other resource restrictions and performance faced on legacy infrastructure, rather it gives time to focus more on real development.

WHY CONTAINERISATION?

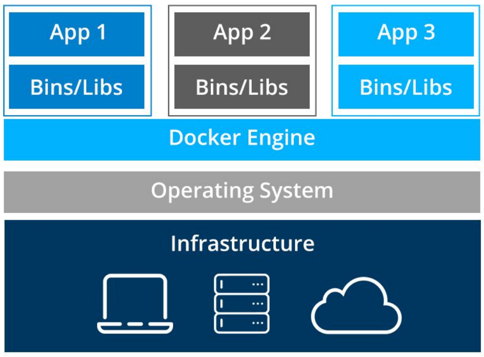
Containers are a light-weight, efficient, and standard way for applications to maneuver between environments and run independently. Everything needed to run the application (like code, runtime system tools, libraries, and dependencies) is packaged inside the container object except the operating system.

It just takes a second to spin up a container where it takes several minutes to provision a virtual machine. It also gives the advantage of handling website traffic load seamlessly still. The same attribute allows containers to be fast and also permits the technology to scale instantly. One can wish to not reboot or restart the system so as to add or remove resources on the go, which is extremely rare in virtual machines.

Each container gets its own exclusive isolated userspace to permit multiple containers to run on one host machine. We are able to see that all the software system-level architecture is being shared across containers. The sole parts that are created from scratch are the binaries and libraries, which make the containers lightweight because it beautifully abstracts the service away from additional libraries it needs to run.



Virtual Machines



Containners

CONTAINER ORCHESTRATION AND ITS BENEFITS

On the basis of necessary information stored in the configuration files, one can automate the container orchestration tool to network between containers and applications, and decide, where to store the logs. The tool helps in automating the process which includes provisioning and deployment of containers, redundancy, and availability of containers, scaling up or down the containers to spread the application load evenly across hosts. It also helps in migration of containers across hosts, in case of shortage of resources in one of the hosts, or if a host dies, allocating the resources between the containers, balancing the requests of service discovery between containers, health monitoring of containers and hosts, and configuring an application in relation to the containers running it and automatically deploys the container onto the best host. It manages the lifecycle of containers/pods especially in large, dynamic environments depending on the machine metrics which is a predetermined specification.

The other benefits of adapting container orchestration are:

Portability:
Modifying and scaling up applications by changing only certain areas using a single command, without affecting the complete application

Enhanced productivity:
Simplified installation process and decreased dependency errors, also increasing and perhaps improving productivity

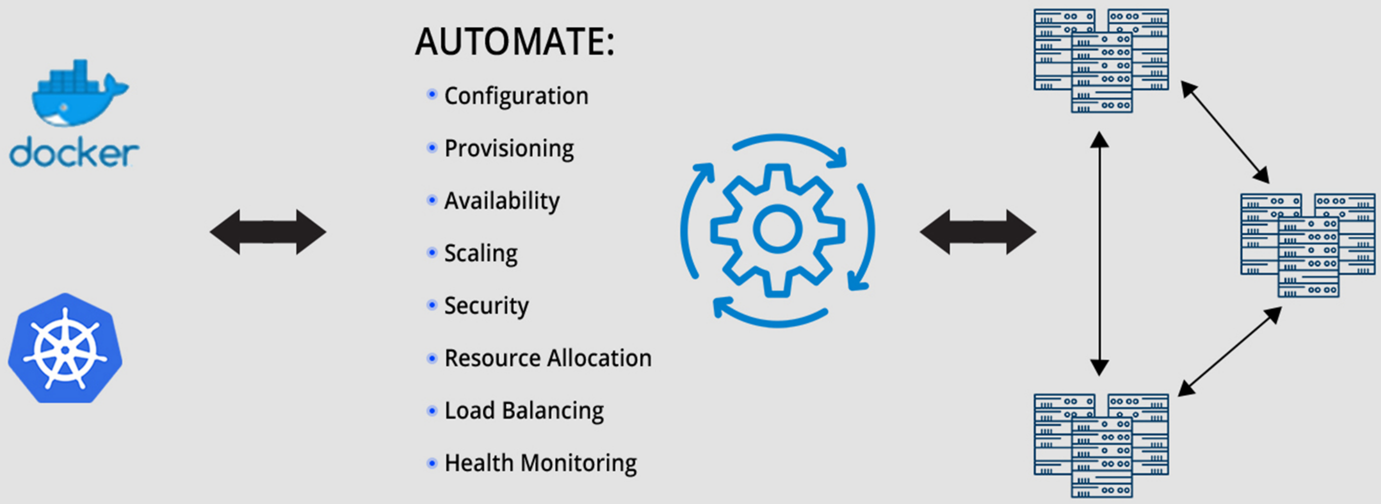
Decreasing Infrastructure Cost:
Reduces the budget across resources that can be used for new projects or innovation

Simple and fast deployment:
Quickly create new containerized applications to address growing traffic, Standardizing processes and products across the spectrum so that they are more consistent and reliable

Improved security:
Share specific resources without risking internal or external654 security.

Decreasing dependencies:
Decreasesthe dependencies cross-team (Development and Operations teams)

Container Orchestration Software (Docker,Openshift & Kubernetes)



EXPERTISE AND DEPLOYMENT STRATEGY: VVDN WAY

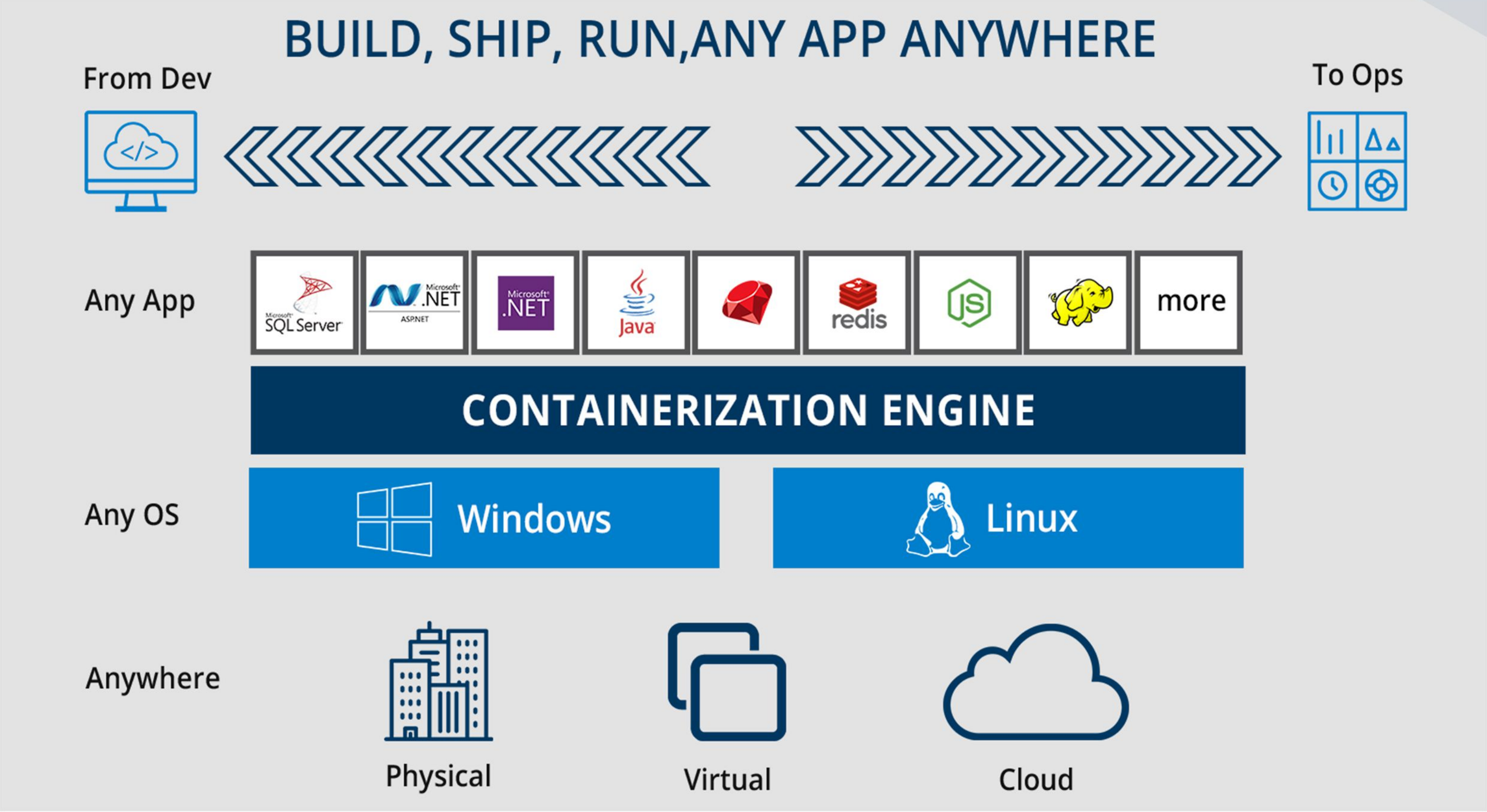
For the longest time, deploying an application into production was tedious and would require many man-hours. However, containers made it easy for standardized testing and deployment, both. Similarly, container orchestration is doing the same job for the data center, it gives us the liberty of choosing where a container will be hosted, managing the lifecycle of containers, how they will be started, monitored, and killed.

While the container format itself is largely settled, for now, though container orchestration is the big fight of the moment, and the real challenge is to deploy and manage those containers. The beauty of container orchestration tools is that it can be used in any environment where we choose to run containers. When using any container orchestration tool, there is a need to define application configuration in a YAML or JSON file, which should have details of how to establish networking between containers, mount storage volumes, and where to store logs for that container.

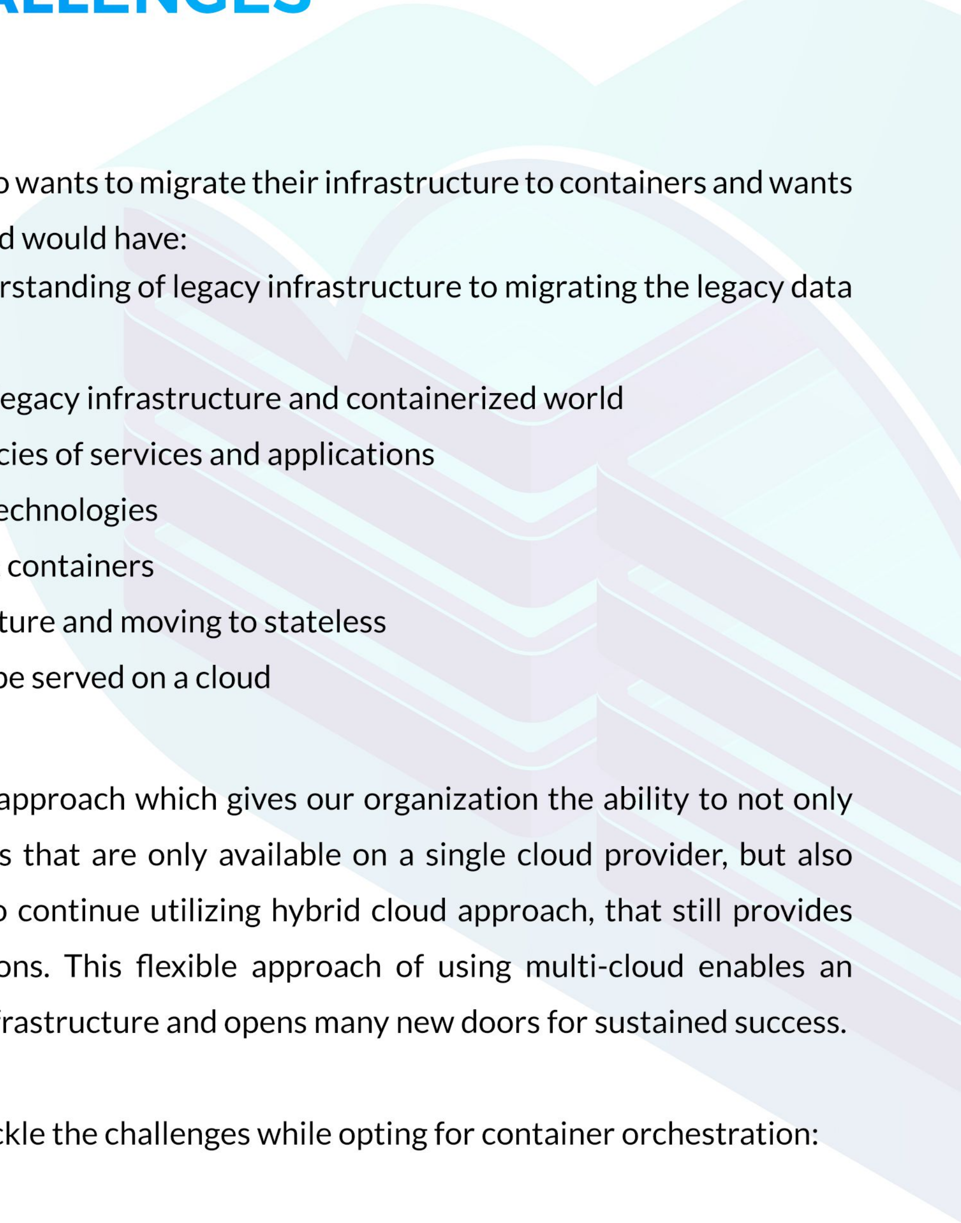
In VVDN, we follow orchestration and containerization practices, which results in a 90% reduction in deployment time for software products, and a 40% reduction in production incidents that could lead to disruptions, faster resolution of application issues and higher customer loyalty.

Development teams branch and version control these configuration files so that the same applications get deployed across different environments before deploying them to production clusters. Once proven to be stable, DevOps promotes the containers from the pre-production environment to production and deploys them on hosts. At the time of deploying a new container into the cluster, the orchestration tool schedules the deployment and looks for the host to place the container, based on machine metrics (like Memory availability, Disk Usage, and CPU).

We at VVDN, place containers according to labels or metadata, or according to their proximity in relation to other hosts—all kinds of constraints can be used. Once the container is running on the host, the orchestration tool manages its lifecycle according to the specifications laid out in the container's definition file. We also take care of rollback as an automatic process, so if there is any need of rollback, or if issues occur with the new deployment at any time, it could be handled on its own.



RESOLVING CHALLENGES
The VVDN Way



The biggest challenges for anyone who wants to migrate their infrastructure to containers and wants to use container orchestration method would have:

- The requirement of deep understanding of legacy infrastructure to migrating the legacy data center infrastructure on cloud
- Bridging the gap between the legacy infrastructure and containerized world
- Decoupling of inter-dependencies of services and applications
- Choosing the right container technologies
- Adapting processes to support containers
- Getting rid of stateful architecture and moving to stateless
- Rewriting application code to be served on a cloud

We at VVDN follow the multi-cloud approach which gives our organization the ability to not only leverage cutting-edge cloud solutions that are only available on a single cloud provider, but also provides a real way for companies to continue utilizing hybrid cloud approach, that still provides meaningful value for their applications. This flexible approach of using multi-cloud enables an entirely new approach to software infrastructure and opens many new doors for sustained success.

We work on the following areas to tackle the challenges while opting for container orchestration:

- Skillful resources
- Adapting to fast-evolving technology
- Resource allocation and utilization
- Implementation strategy
- Complex lifecycle management
- Choosing the right container technology
- Container monitoring
- Container security
- Resource allocation and utilization

We at VVDN,
work on following orchestration services:

Amazon ECS:

The Amazon EC2 Container Service (ECS) supports Docker containers and lets us run applications on a managed cluster of Amazon EC2 instances.

Docker Swarm:

Docker Swarm provides native clustering functionality for Docker containers that enables to turn a group of Docker engines into a single, virtual Docker Engine.

Kubernetes:

Kubernetes is an orchestration system for Docker containers that handles scheduling and manages workloads based on user-defined parameters.

Azure Container Service (ACS):

ACS lets us create a cluster of virtual machines that act as container hosts along with master machines that are used to manage our application containers.

Google Container Engine (GKE):

Google Container Engine, which is built on Kubernetes, lets us run Docker containers on the Google Cloud Platform and helps schedule containers into the cluster & manages them based on user-defined requirements.

From major infrastructure companies to new-age startups providing cloud services, everyone is clamoring to stake out their place in the ecosystem. It is essential that contributors perform on container orchestration tools, as it is required for deploying real-world applications, thus driving the adoption of Docker and containers. Orchestration brings more agility through microservices-based design methodology through containers.

Author

Name: Asmita Singh

Profile: DevOps Lead

About VVDN

VVDN is a leading Product Engineering, Cloud and Manufacturing Company that is uniquely positioned to deliver versatile, innovative, world-class quality products

VVDN Global Footprint : US Canada Europe S.Korea Japan India